



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

**Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων**



Πτυχιακή εργασία

**Online Σύστημα Διαχείρισης Ζητημάτων
(Online Issue Management System)**



**Ειρήνη Ορφανουδάκη (ΑΜ: 1372)
E-mail: rena-or@hotmail.com**

Ηράκλειο – Δεκέμβριος 2010

Επόπτης Καθηγητής: Παπαδουράκης Γεώργιος

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά, όλους τους ανθρώπους που ήταν κοντά μου κατά τη διάρκεια εκπόνησης της πτυχιακής αυτής, ιδιαίτερα όσους με βοήθησαν και με καθοδήγησαν σε στιγμές που ίσως απόκλινα του στόχου μου.

Κυρίως θέλω να πω ένα μεγάλο ευχαριστώ στην οικογένεια μου που πάντα βρίσκεται δίπλα μου και με στηρίζει.

Abstract

Issue Tracking Systems assist service provider organizations to manage and maintain lists of issues about support interventions on behalf of clients who make requests. They are considered very useful systems that allow information flow between customers and organizations, increasing the efficiency and effectiveness of operational management and leading to the ultimate goal that is the customer satisfaction.

This thesis proposes a web based solution, namely CIMaS, that extends the Joomla! Content Management System in order to support Issue Tracking functionality. Combining these two technology domains, Issue and Content Management Systems, will create a unified bidirectional communication platform that will improve the operations of service provider organizations and at the same time it will enhance customer experience. More specifically, the proposed system will allow customers to receive all relevant information within their interests as well as they will be given the ability to report issues and monitor the progress of their resolution within the same environment. The approach followed in the current work includes initially an extensive review of the relevant technologies. Afterwards, the proposed system requirements are clearly set and analyzed. The design of the proposed work is done with Use Case Scenarios and various diagrams such as workflow, dataflow and class. Finally, the implementation of the system is based on the integration of existing components as well as custom development following the Model View Controller pattern.

Σύνοψη

Τα Issue Tracking Systems βοηθούν οργανισμούς παροχής υπηρεσιών να διαχειρίζονται και να συντηρούν καταλόγους ζητημάτων, σχετικά με παρεμβάσεις υποστήριξης για λογαριασμό πελατών που κάνουν σχετικά αιτήματα. Θεωρούνται πολύ χρήσιμα συστήματα που επιτρέπουν τη ροή της πληροφορίας μεταξύ πελατών (τελικών χρηστών) και οργανισμών, αυξάνοντας έτσι την αποτελεσματικότητα και την αποδοτικότητα της λειτουργικής διαχείρισης και οδηγούν στον τελικό στόχο που είναι η ικανοποίηση του πελάτη.

Αυτή η πτυχιακή, προτείνει μια web based λύση (το CIMaS), που επεκτείνει το σύστημα διαχείρισης περιεχομένου Joomla! προκειμένου να υποστηρίξει λειτουργικότητα Issue Tracking. Ο συνδυασμός αυτών των τεχνολογικών τομέων, Issue και Content Management Systems, θα δημιουργήσει μια ενιαία πλατφόρμα αμφίδρομης επικοινωνίας η οποία αφενός βελτιώνει τις διεργασίες οργανισμών παροχής υπηρεσιών και αφετέρου ενισχύει την εμπειρία των πελατών. Πιο συγκεκριμένα, το προτεινόμενο σύστημα, θα επιτρέπει στους πελάτες να λαμβάνουν όλες τις σχετικές με τα ενδιαφέροντά τους πληροφορίες, ενώ παράλληλα στο ίδιο περιβάλλον, τους δίνεται η δυνατότητα να καταχωρούν ζητήματα και να παρακολουθούν την πρόοδο για την επίλυσή τους. Η προσέγγιση που ακολουθείται σε αυτή την εργασία, περιλαμβάνει αρχικά μια εκτενή επισκόπηση των σχετικών τεχνολογιών. Στη συνέχεια, καθορίζονται και αναλύονται οι προτεινόμενες απαιτήσεις του συστήματος. Ο σχεδιασμός του προτεινόμενου συστήματος, γίνεται με χρήση Σεναρίων Χρήσης και διαφόρων διαγραμμάτων όπως τα διαγράμματα ροής εργασιών, ροής δεδομένων και κλάσεων. Τέλος, για την υλοποίηση του συστήματος, ακολουθείται το πρότυπο του Model View Controller.

Περιεχόμενα

Ευχαριστίες	ii
Abstract	iii
Σύνοψη	iv
Περιεχόμενα	v
Πίνακας Εικόνων	viii
Λίστα Πινάκων	x
Πίνακας Διαγραμμάτων.....	xi

1. Εισαγωγή

1.1	Περίληψη.....	1
1.2	Κίνητρο για την Διεξαγωγή της Εργασίας.....	1
1.3	Σκοπός και Στόχοι Εργασίας.....	1
1.4	Δομή Εργασίας.....	2

2. CMS: Ανασκόπηση Τεχνολογιών

2.1	Εισαγωγή.....	3
2.2	Content Management Systems (CMS).....	3
	2.2.1 Joomla!.....	4
	2.2.2 Drupal.....	12
	2.2.3 WordPress	15
	2.2.4 Σύγκριση των CMSs	19
2.3	Περίληψη Κεφαλαίου.....	20

3. Issue Tracking Systems: Ανασκόπηση Τεχνολογιών

3.1	Εισαγωγή.....	21
3.2	Issue Tracking Systems.....	21
3.3	Ροή Εργασίας Issue Tracking Systems	21
3.4	Αρχιτεκτονική Issue Tracking System.....	22
3.5	Παραδείγματα Issue Tracking Systems.....	22
	3.5.1 Joomla! Extentions.....	22
	3.5.2 Διαδικτυακές εφαρμογές (Web Based εφαρμογές).....	27
3.6	Περίληψη Κεφαλαίου.....	30

4. CIMaS: Καταγραφή και Ανάλυση Απαιτήσεων

4.1	Εισαγωγή.....	31
4.2	Ορολογία	31
4.3	Προσδιορισμός Απαιτήσεων	32
4.4	Καταγραφή Εμπλεκόμενων χρηστών.....	32
4.5	Καταγραφή απαιτήσεων.....	32
	4.5.1 Σκοπιά Πελάτη	32
	4.5.2 Σκοπιά Διευθυντή.....	33
	4.5.3 Σκοπιά Υπαλλήλου	33
4.6	Ανάλυση Απαιτήσεων	33
	4.6.1 Σκοπιά Πελάτη	33
	4.6.2 Σκοπιά Διευθυντή.....	34
	4.6.3 Σκοπιά Υπαλλήλου	35
4.7	Περίληψη Κεφαλαίου.....	36

5. Σχεδιασμός CIMaS

5.1	Εισαγωγή.....	37
5.2	Αρχιτεκτονική Συστήματος.....	37
5.3	Καταγραφή Σεναρίων χρήσης.....	37
5.3.1	Πελάτης.....	37
5.3.2	Υπάλληλος.....	38
5.3.3	Διευθυντής.....	38
5.4	Ανάλυση Σεναρίων Χρήσης.....	38
5.4.1	Σενάρια Χρήσης για τον Πελάτη.....	39
5.4.2	Σενάρια Χρήσης για τον Υπάλληλο.....	47
5.4.3	Σενάρια Χρήσης για τον Διευθυντή.....	54
5.5	Διάγραμμα Ροής Εργασιών.....	64
5.5.1	Για τον Πελάτη.....	65
5.5.2	Για τον Υπάλληλο.....	67
5.5.3	Για τον Διευθυντή.....	68
5.6	Class Diagram.....	70
5.7	Data Flow Diagrams.....	72
5.7.1	Data Flow Diagram για τον Πελάτη.....	72
5.7.2	Data Flow Diagram για τον Υπάλληλο.....	74
5.7.3	Data Flow Diagrams για τον Διευθυντή.....	75
5.8	Περίληψη Κεφαλαίου.....	79

6. Υλοποίηση Συστήματος

6.1	Εισαγωγή.....	80
6.2	Επιλογή και Εγκατάσταση CMS.....	80
6.2.1	Εγκατάσταση WampServer.....	81
6.2.2	Εγκατάσταση Joomla!.....	82
6.3	Δημιουργία και Παρακολούθηση Ζητημάτων.....	85
6.3.1	Model View Controller Patern.....	85
6.3.2	Απαραίτητες ενέργειες για την υλοποίηση του component.....	86
6.3.3	Τρόπος γραφής των Model – View - Controller.....	91
6.4	CIMaS Component.....	92
6.4.1	Front End.....	92
6.4.2	Back end.....	99
6.5	Επεκτάσεις που προσφέρουν επιπλέον λειτουργικότητα.....	102
6.5.1	Δημιουργία Λογαριασμού Χρήστη.....	102
6.5.2	Απόδοση δικαιωμάτων ανάλογα με το χρήστη.....	103
6.5.3	Αποστολή και λήψη αρχείων ενδιαφέροντος.....	103
6.6	Περίληψη Κεφαλαίου.....	104

7. Αποτελέσματα

7.1	Συμπεράσματα.....	105
7.2	Μελλοντική Εργασία και Επεκτάσεις.....	106

Βιβλιογραφία	107
Πηγές	108
Παράρτημα Α: Πηγαίος Κώδικας CIMaS component.....	109
Παράρτημα Β: Εγκατάσταση CIMaS component	193
Παράρτημα Γ: Παρουσίαση Πτυχιακής Εργασίας (Διαφάνειες).....	194

Πίνακας Εικόνων

Εικόνα 1: Δομή ενός Joomla! site	5
Εικόνα 2: Front End Joomla! Site (joomla.org)	6
Εικόνα 3: Joomla! site - Back End	7
Εικόνα 4: Joomla! users Διαχείριση Χρηστών - Απόδοση Δικαιωμάτων	8
Εικόνα 5: Joomla! Διαχείριση Περιεχομένου	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 6: Joomla! Extentions Management.....	11
Εικόνα 7: Drupal site (drupal.org).....	12
Εικόνα 8: Drupal core – Πυρίνας του Drupal	13
Εικόνα 9: Drupal Διαχείριση Χρηστών.....	14
Εικόνα 10: Drupal Διαχείριση Επεκτασιμότητας.....	15
Εικόνα 11: Front End WordPress Site (wordpress.org)	16
Εικόνα 12: WordPress Dashboard.....	17
Εικόνα 13: WordPress Διαχείριση Χρηστών	18
Εικόνα 14: Freestyle Support - Περιοχή Διαχείρισης Εισητηρίων	22
Εικόνα 15: Freestyle Support - Φόρμα Απάντησης σε Ticket	23
Εικόνα 16: Issue Manager - Φόρμα Εισαγωγής Νέου Ticket	24
Εικόνα 17: Issue Manager - Λίστα Tickets	25
Εικόνα 18: Διαχείριση Huru Helpdesk για το Joomla! back end	26
Εικόνα 19: Πίνακας Διαχείρισης Ζητημάτων απο το Προσωπικό Υποστήριξης.....	27
Εικόνα 20: SysAid Διεπαφή Διαχείρισης Ζητημάτων	28
Εικόνα 21: SysAid Portal Τελικού Χρήστη – Πελάτη.....	29
Εικόνα 22: Wizard Παράθυρο Διαλόγου για Εγκατάσταση WampServer 2	81
Εικόνα 23: Δημιουργία Βάσης Δεδομένων στην MySQL	82
Εικόνα 24: MySQL Απόδοση Δικαιωμάτων Χρήστη.....	83
Εικόνα 25: Εγκατάσταση Joomla! Προληπτικός Έλεγχος.....	84
Εικόνα 26: Σχήμα Λειτουργία Model View Controller patern	86
Εικόνα 27: Φάκελος com_cimas\administrator.....	87
Εικόνα 28: Περιεχόμενα Φακέλου com_cimas\administrator\controllers	87
Εικόνα 29: Περιεχόμενα Φακέλου com_cimas\administrator\models	87
Εικόνα 30: Περιεχόμενα Φακέλου com_cimas\administrator\views	88
Εικόνα 31: Περιεχόμενα Φακέλου com_cimas\administrator\tables	89
Εικόνα 32: Παράδειγμα toolbar	89
Εικόνα 33: Φάκελος com_cimas\components.....	89
Εικόνα 34: Περιεχόμενα Φακέλου com_cimas\components\controllers	89
Εικόνα 35: Περιεχόμενα Φακέλου com_cimas\components\models.....	90
Εικόνα 36: Περιεχόμενα Φακέλου com_cimas\components\views	90
Εικόνα 37: Είσοδος Χρήστη στο Σύστημα	92
Εικόνα 38: CIMaS Μενού.....	93
Εικόνα 39: Δημιουργία Issue από Πελάτη	93
Εικόνα 40: Αποθήκευση στη Βάση Δεδομένων.....	93
Εικόνα 41: Δημιουργία νέου Issue από Υπάλληλο – Διευθυντή	94
Εικόνα 42: Αποθήκευση στη Βάση Δεδομένων.....	94
Εικόνα 43: Δημιουργία Tasks	95
Εικόνα 44: Φόρμα Δημιουργίας Task	95
Εικόνα 45: Εμφάνιση Task μετά τη Δημιουργία του	96
Εικόνα 46: Αποθήκευση στη Βάση Δεδομένων.....	96
Εικόνα 47: Δημιουργία Σχόλιου Πελάτη	97
Εικόνα 48: Αποτελέσματα στη Βάση Δεδομένων.....	97
Εικόνα 49: Δημιουργία Comment για Υπάλληλο – Διευθυντή	98
Εικόνα 50: Προβολή Ιστορικού Πελάτη	98
Εικόνα 51: Προβολή Ιστορικού για Υπάλληλο – Διευθυντή.....	99
Εικόνα 52: Διαχείριση Στοιχείων Χρήστη απο τον Διευθυντή.....	99

Εικόνα 53: Εμφάνιση Μη Ολοκληρωμένων Issues στο Διευθυντή	100
Εικόνα 54: Επεξεργασία υπάρχοντος Issue.....	100
Εικόνα 55: Δημιουργία Issue απο το back end	101
Εικόνα 56: Δημιουργία Task απο το back end.....	101
Εικόνα 57: Δημιουργία Comment από το back end.....	102
Εικόνα 58: Λογαριασμός Χρήστη.....	102

Λίστα Πινάκων

Πίνακας 1: Ομάδες Χρηστών και Δικαιώματα χρήσης Joomla!	9
Πίνακας 2: Ερμηνεία Χρησιμοποιημένων Σχημάτων των Work Flows	65
Πίνακας 3: Ερμηνεία Χρησιμοποιηθέντων Σχημάτων των Data Flow Diagrams	72

Πίνακας Διαγραμμάτων

Διάγραμμα 1: Ροή Εργασιών Πελάτη	66
Διάγραμμα 2: Ροή Εργασιών Υπαλλήλου.....	67
Διάγραμμα 3: Ροή Εργασιών Διευθυντή.....	69
Διάγραμμα 4: Σχήμα Βάσης Δεδομένων.....	71
Διάγραμμα 5: Ροή Δεδομένων Πελάτη	73
Διάγραμμα 6: Ροή Δεδομένων για Δημιουργία Λογαριασμού Πελάτη	74
Διάγραμμα 7: Ροή Δεδομένων για Υπάλληλο.....	75
Διάγραμμα 8: Αρχικό Διάγραμμα Ροής Δεδομένων Διευθυντή.....	76
Διάγραμμα 9: Ροή Δεδομένων Διευθυντή - Δημιουργία Issue	76
Διάγραμμα 10: Ροή Δεδομένων Διευθυντή για Υπάρχον Issue	77
Διάγραμμα 11: Ροή Δεδομένων Διευθυντή για Tasks.....	78
Διάγραμμα 12: Ροή Δεδομένων Διευθυντή για Χρήστες.....	79

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Η εποχή που διανύουμε, χαρακτηρίζεται από τη ραγδαία αύξηση της χρήσης της ηλεκτρονικής ενημέρωσης, της ηλεκτρονικής παραγγελίας αγαθών, της ηλεκτρονικής επικοινωνίας των ανθρώπων ακόμα και της χρήσης της ηλεκτρονικής δοσοληψίας με τράπεζες και δημόσιους οργανισμούς.

Η μεταστροφή λοιπόν αυτή των τελευταίων ετών καθιστά αναγκαίο για κάθε επιχείρηση – οργανισμό που παρέχει υπηρεσίες, υλικά αγαθά ή και δραστηριοποιείται στο internet να παρέχει ηλεκτρονική δημιουργία και παρακολούθηση ζητημάτων (είτε για αναφορά κάποιου προβλήματος – δυσλειτουργίας είτε για ενημέρωση πορείας εργασιών σε πρωτύτερη δήλωση δυσλειτουργίας κτλ.) τόσο για ενημέρωση του πελάτη, όσο και για την παρακολούθηση αυτών από την πλευρά του προσωπικό της εταιρείας - οργανισμού.

Η παρούσα πτυχιακή ασχολείται με την υλοποίηση ενός συστήματος online διαχείρισης ζητημάτων, που στόχο έχει, τη διευκόλυνση της ροής εργασιών, για ζητήματα που δημιουργούνται, έως την αποκατάστασή τους.

1.1 Περίληψη

Στην πτυχιακή αυτή, γίνεται ανασκόπηση των τεχνολογιών που υπάρχουν για συστήματα διαχείρισης περιεχομένου καθώς και σύγκριση αυτών για την επιλογή ενός, με τη βοήθεια του οποίου θα δημιουργηθεί ένα online σύστημα διαχείρισης ζητημάτων.

Καθώς υπάρχουν ήδη έτοιμα συστήματα που διαχειρίζονται ζητήματα, γίνεται μια ανασκόπηση και μελέτη γύρω από αυτά, για τη λήψη ιδεών και πληροφοριών, ώστε στη συνέχεια, να αρχίσει η διαδικασία μελέτης και υλοποίησης ενός τέτοιου συστήματος από την αρχή.

Επίσης στην παρούσα πτυχιακή, γίνονται ανάλυση απαιτήσεων, μελέτη καθώς και η υλοποίησή για ένα σύστημα διαχείρισης ζητημάτων, το οποίο έχει ως κύριο στόχο τη βέλτιστη διαχείριση ζητημάτων, για να επιτευχθεί η μέγιστη ικανοποίηση των χρηστών (τόσο των πελατών όσο και των διαχειριστών), καθώς οι οντότητες αυτές είναι που θα διατηρήσουν ένα σύστημα ζωντανό ή θα το απαξιώσουν περνώντας το στην αχρηστία.

1.2 Κίνητρο για την Διεξαγωγή της Εργασίας

Βασικό κίνητρο για τη διεξαγωγή της εργασίας αυτής, αποτέλεσε η μεταστροφή όλο και περισσότερων χρηστών σε διαδικτυακές τεχνολογίες. Σημαντικό ρόλο έπαιξε επίσης η ανάγκη δημιουργίας εξειδικευμένων γνώσεων για υλοποίηση συστημάτων – εφαρμογών από τη σύλληψη της ιδέας έως και την υλοποίησή τους. Τέλος κίνητρο παρέχει η ενασχόληση με τη γλώσσα υλοποίησης διαδικτυακών εφαρμογών, PHP με τη χρήση της οποίας θα δημιουργηθεί μια επέκταση που εγκαθίσταται στο CMS Joomla! και θα παρέχει τη λειτουργικότητα που αναφέρεται στην περίληψη.

1.3 Σκοπός και Στόχοι Εργασίας

Σκοπός της εργασίας αυτής, είναι η τριβή και η ενασχόληση με τη γλώσσα υλοποίησης διαδικτυακών εφαρμογών PHP καθώς και η βέλτιστη εκμάθησή της.

Στόχος της εκπόνησης αυτής της εργασίας, είναι η απόκτηση εμπειρίας σε θέματα υλοποίησης εφαρμογών και συστημάτων. Πιο συγκεκριμένα ο στόχος που επιδιώκεται είναι μετά το πέρας της εργασίας αυτής, να μπορούν να δημιουργηθούν συστήματα από την αρχή (τη σύλληψη της ιδέας) μέχρι και την υλοποίησή τους.

1.4 Δομή Εργασίας

Η δομή της εργασίας αυτής, έχει ως εξής:

Στο 1^ο κεφάλαιο, παραθέτονται εισαγωγικά στοιχεία, σχετικά με τους λόγους επιλογής υλοποίησης της πτυχιακής αυτής και θέτονται στόχοι ώστε μετά την ολοκλήρωσή της, να μπορούν να διεξαχθούν συμπεράσματα.

Στο 2^ο Κεφάλαιο γίνεται ανασκόπηση των CMS, παρουσίαση τρόπου χρήσης τους καθώς και σύγκριση μεταξύ τους, για την επιλογή ενός, στο οποίο θα βασιστεί η υλοποίηση της εφαρμογής που θα μελετηθεί και θα δημιουργηθεί σε αυτή την πτυχιακή.

Στο 3^ο Κεφάλαιο, γίνεται ανασκόπηση των Issue Tracking Systems, για την άντληση ιδεών και για την ενημέρωση των προϊόντων που είναι διαθέσιμα και των δυνατοτήτων που παρέχουν.

Στο 4^ο Κεφάλαιο, γίνεται η συλλογή και ανάλυση των απαιτήσεων που δημιουργούνται από τους χρήστες για το σύστημα διαχείρισης ζητημάτων που θα υλοποιηθεί.

Στο 5^ο Κεφάλαιο, γίνεται η μελέτη του συστήματος. Στη μελέτη αυτή, περιλαμβάνονται διαγράμματα ροής εργασιών, διαγράμματα ροής δεδομένων, διάγραμμα βάσης δεδομένων, καθώς και η καταγραφή των σεναρίων χρήσης του συστήματος ανά εμπλεκόμενο χρήστη.

Στο 6^ο Κεφάλαιο, παρουσιάζεται η υλοποίηση της εφαρμογής, το μοντέλο – αρχιτεκτονική που ακολουθήθηκε για το σκοπό αυτό, καθώς και μερικές επιπλέον επεκτάσεις, που χρησιμοποιούνται για την παροχή επιπλέον δυνατοτήτων σε όλους τους χρήστες του συστήματος.

Τέλος στο 7^ο Κεφάλαιο, εξάγονται τα συμπεράσματα από την υλοποίηση της πτυχιακής αυτής. Επίσης αναφέρονται πιθανές μελλοντικές αναβαθμίσεις για το υλοποιηθέν σύστημα.

ΚΕΦΑΛΑΙΟ 2

CMS: Ανασκόπηση Τεχνολογιών

2.1 Εισαγωγή

Στο κεφάλαιο αυτό, θα γίνει μια γνωριμία με τα Συστήματα Διαχείρισης Περιεχομένου (CMSs) και θα παρουσιαστούν τα βασικά δομικά χαρακτηριστικά των πλέον γνωστότερων από αυτά: Joomla!, Drupal και WordPress. Επίσης για να μπορέσουν να εξαχθούν συμπεράσματα για το ποιο αρμόζει καλύτερα ανά περίπτωση χρήσης, θα γίνει μια σύγκριση μεταξύ τους.

2.2 Content Management Systems (CMS)

CMS: Είναι το ακρώνυμο του Content Management System (Σύστημα Διαχείρισης Περιεχομένου). Είναι ένα σύστημα που επιτρέπει στους χρήστες να δημιουργούν, να επεξεργάζονται και να διαχειρίζονται αρχεία διαφορετικών τύπων, με σκοπό να ενταχθούν στο περιεχόμενο μιας ιστοσελίδας. Αυτά τα αρχεία, μπορούν να περιέχουν αρχεία δεδομένων, audio / video, εικόνες κ.α.

Το CMS αφενός βοηθάει στη διαχείριση του περιεχομένου ενός ιστοτόπου χωρίς καμία τεχνική γνώση της γλώσσας σήμανσης HTML (ειδικός τρόπος γραφής κειμένου για εμφάνιση σε διαδικτυακούς τόπους¹), αφετέρου κάνει διαχωρισμό σε γκρουπ χρηστών, σε καθένα από τα οποία υπάρχουν διαφορετικοί ρόλοι, αρμοδιότητες και δικαιώματα. Η ιδέα είναι ότι παραπάνω από ένα άτομα μπορούν να δημιουργούν, να επεξεργάζονται και να διαχειρίζονται το περιεχόμενο το οποίο θα δουν οι απλοί χρήστες, που έχουν περιορισμένα δικαιώματα (συνήθως μόνο να δουν το περιεχόμενο). Εν ολίγοις, μπορούμε να κάνουμε μια εύκολα διαχειρίσιμη ιστοσελίδα με τη βοήθεια ενός CMS, καθώς η δημιουργία, η επεξεργασία και η διαχείριση του περιεχομένου γίνεται απλή υπόθεση.

Συχνά θα ακουστεί η αναφορά σε «δυναμική» ή «στατική» ιστοσελίδα. Η κύρια διαφορά μεταξύ αυτών των όρων είναι ο τρόπος αποθήκευσης των δεδομένων τους. Το περιεχόμενο ενός δυναμικού site, αποθηκεύεται και διαχειρίζεται από κάποιο σύστημα διαχείρισης βάσεων δεδομένων (DBMS – DataBase Management System), ενώ ενός στατικού, αποθηκεύεται και διαχειρίζεται μέσα από συγκεκριμένο φάκελο αρχείων γραμμένων σε HTML γλώσσα.

Η διαφορά που δημιουργείται από τον τρόπο αποθήκευσης και διαχείρισης των δεδομένων, έχει αντίκτυπο στον τρόπο και την ταχύτητα που ανανεώνεται το περιεχόμενο των διαδικτυακών τόπων. Έτσι με τη χρήση της βάσης δεδομένων, οι αλλαγές φαίνονται σε πραγματικό χρόνο δηλαδή αμέσως μόλις αποθηκευτεί το αρχείο, ενώ οι αντίστοιχες αλλαγές σε ένα στατικό site για να υλοποιηθούν θα έπρεπε να ακολουθηθούν τα εξής βήματα: Συγγραφή κώδικά → Τοποθέτηση ιστοσελίδας εκτός λειτουργίας → Αποθήκευση του αρχείου στο επιθυμητό σημείο → Επεξεργασία δεδομένων σε ήδη υπάρχοντα αρχεία για την εμφάνιση του νέου αντικειμένου → Έλεγχος ορθής λειτουργίας → Στην περίπτωση ορθής λειτουργίας, τοποθετείται η ιστοσελίδα εντός λειτουργίας (online). Σε περίπτωση όμως μη ορθής λειτουργίας σε ποιο από όλα τα αρχεία θα εντοπιστεί το λάθος αν η ιστοσελίδα αποτελείται από πολλές σελίδες;

¹ el/Wikipedia.org/wiki/html

Με ένα CMS, όλος ο δικτυακός τόπος αποθηκεύεται σε μια βάση δεδομένων. Όλοι οι σύνδεσμοι, τα άρθρα, οι πληροφορίες χρηστών, οι εικόνες και άλλα κομμάτια της ιστοσελίδας επιμελούνται από το διαχειριστή, χρησιμοποιώντας τη βάση δεδομένων. Μπορεί να φαίνεται τρομακτική όλη αυτή η δομή στη βάση δεδομένων, αλλά όλη η διαχείριση και η επιμέλεια του website γίνεται μέσω του Administration Interface, ενός φιλικού προς το χρήστη μενού, που καθιστά το έργο της επικαιροποίησης και της διαχείρισης του περιεχομένου πολύ εύκολο².

2.2.1 Joomla!

Το Joomla! είναι ένα από τα πιο ισχυρά και δημοφιλή αρθρωτά ανοιχτού κώδικα CMS και διανέμεται δωρεάν υπό την GNU General Public Licence. Σχεδιάστηκε για τη γρήγορη δημιουργία διαδραστικών πολύγλωσσων ιστοσελίδων, για online κοινότητες, blogs, εμπορικές εφαρμογές κ.α. Είναι server - based και διατηρεί όλα τα περιεχόμενά του δικτυακού τόπου σε Σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων (<http://en.wikipedia.org/wiki/MySQL>).

Το CMS αυτό, παρέχει ενσωματωμένες διάφορες λειτουργίες σε modules και components, για την προσθήκη χαρακτηριστικών στις ιστοσελίδες (π.χ. δημοσκοπήσεις, αναζήτηση κτλ.). Επίσης, υπάρχουν διαθέσιμες στο διαδίκτυο πολλές έτοιμες επεκτάσεις (components – modules - plugins), που ικανοποιούν σχεδόν όλες τις απαιτήσεις των σχεδιαστών ιστοσελίδων.

Το Joomla! είναι πολύ επιτυχημένο και νικητής του Source Content Management System Award. Έχει πολλές δυνατότητες, πολλά χαρακτηριστικά και ακόμα περισσότερα μέλη στην κοινότητα υποστήριξής του².

Ο επίσημος ιστότοπος για το Joomla! είναι το www.joomla.org

2.2.1.1 Ιστορική Αναδρομή

Το 2000, η εταιρεία Miro με έδρα στην Αυστραλία ανέπτυξε ένα ιδιόκτητο CMS που το ονόμασε Mambo. Ένα χρόνο αργότερα, το κυκλοφόρησε δωρεάν για το κοινό υπο την GNU General Public Licence (GPL). Το Mambo, συγκέντρωσε σε μικρό χρονικό διάστημα μια μεγάλη κοινότητα υποστήριξης η οποία διέπονταν από ενθουσιασμό για την πλατφόρμα αυτή.

Το 2005, μια διαφορά για τα πνευματικά δικαιώματα ανάμεσα στη Διευθύνουσα Επιτροπή και στους τότε σχεδιαστές του Mambo, προκάλεσε την παραίτηση των περισσότερων μελών της εταιρείας που είχαν ασχοληθεί με το project αυτό.

Το αποτέλεσμα ήταν μια νέα οντότητα ονόματι Open Source Matters, η οποία μετονόμασε την πρωτότερη πλατφόρμα από Mambo σε Joomla!. Το όνομα αυτό της καινούριας πλατφόρμας, είναι ένας ακουστικός συλλαβισμός της λέξης «joomla» που προέρχεται από τη φυλή των Σουαχίλοι και σημαίνει «όλοι μαζί».

Το Joomla! έκανε επίσημα την εμφάνισή του με την έκδοση 1.0 στις 16 Σεπτεμβρίου του 2005. Η πρώτη έκδοση ήταν ουσιαστικά μια επανέκδοση του Mambo, για να γίνει η γνωστοποίηση αλλαγής της ονομασίας, με μερικές μόνο διορθώσεις σε bugs (“bugs = έντομα”, τεχνική ορολογία για την αναφορά δυσλειτουργιών – “ανάρμοστων συμπεριφορών”). Στα επόμενα όμως 2 χρόνια, ακολούθησαν 14 αναβαθμισμένες εκδόσεις (updates) και πολλά βραβεία ανοιχτού κώδικα.

Στις 21 Ιανουαρίου 2008, ανακοινώθηκε το Joomla! 1.5 το οποίο αποτελούσε την πρώτη μεγάλη αναθεώρηση. Η αναβάθμιση αυτή, ενδυνάμωσε τα CMS ανοιχτού κώδικα. Σε αυτή την έκδοσή του, το Joomla! απέκτησε καινούριο API (αρχικά από το Application Programming Interface που είναι η διεπαφή προγραμματισμού εφαρμογών) και έγινε διεθνώς αναγνωρίσιμο καθώς πρόσθεσε πολλά σετ χαρακτήρων και πλέον υποστήριξε τις γλώσσες που διαβάζονται από τα δεξιά στα αριστερά. Επίσης ενισχύθηκε στους τομείς της ευχρηστίας, επεκτασιμότητας και της διαχείρισης του template (μορφή παρουσίασης και αισθητικής του site τόσο για το διαχειριστικό μέρος όσο και για το front end).

Μετά από 22 αναβαθμίσεις (updates) της έκδοσης 1.5.x, αυτή την περίοδο κάνει την εμφάνισή της μια καινούρια έκδοση, η 1.6, η οποία είναι σχεδιασμένη με το πρότυπο Model View Controller (αναλύεται στο κεφάλαιο 6) και υπόσχεται τη διόρθωση των περισσότερων bugs που εξακολουθούσαν να υπάρχουν μέχρι και την έκδοση 1.5.22³.

² Apress, Foundation Joomla! 2009

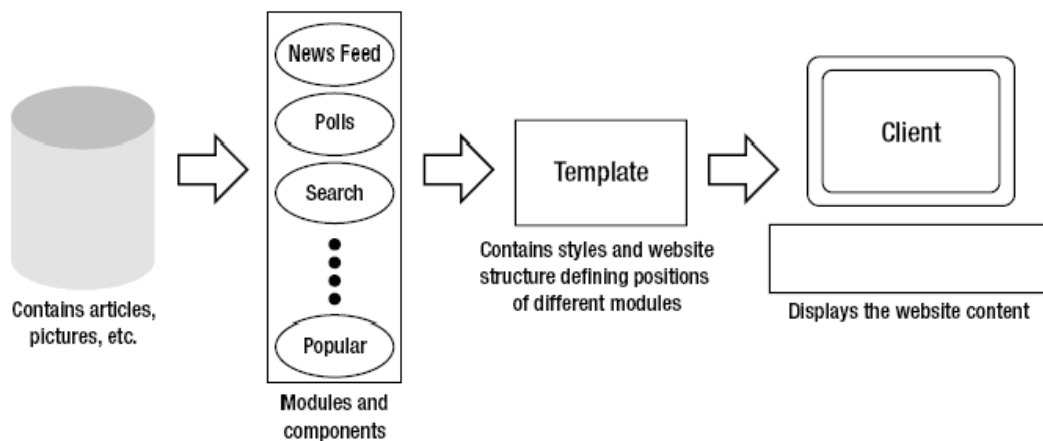
³ Prentice Hall, Joomla! Guide 1.5 Joomla!book, 2008

2.2.1.2 Δομή ενός Joomla! Website

Στο παρακάτω σχήμα (Εικόνα 1) φαίνεται ότι τα περιεχόμενα (φωτογραφίες, άρθρα κτλ.) της ιστοσελίδας, αποθηκεύονται σε μια βάση δεδομένων.

Όταν ο επισκέπτης επιλέξει κάποιο σύνδεσμο από το μενού της ιστοσελίδας, τα επιθυμητά δεδομένα ανακτώνται από τη βάση αυτή και του εμφανίζονται (μέσω του front end). Πριν όμως από την εμφάνιση τους στο χρήστη, εφαρμόζονται διάφορα components και modules για να φιλτράρουν και να προβάλλουν την πληροφορία που ζητήθηκε.

Μόλις ολοκληρωθεί η παραπάνω διεργασία και πριν ακόμα φτάσει η πληροφορία στο χρήστη, εφαρμόζεται το επιλεγμένο πρότυπο (template) το οποίο αποτελείται από στυλ, που δίνουν μια ελκυστική εμφάνιση στο περιεχόμενο. Το template επίσης καθορίζει το σχεδιασμό και τη δομή της ιστοσελίδας, καθώς περιέχει τη θέση στην οθόνη όλων των modules που πρέπει να εμφανιστούν στο διαδικτυακό τόπο, όπως polls, αναζήτηση κ.α.



Εικόνα 1: Δομή ενός Joomla! site

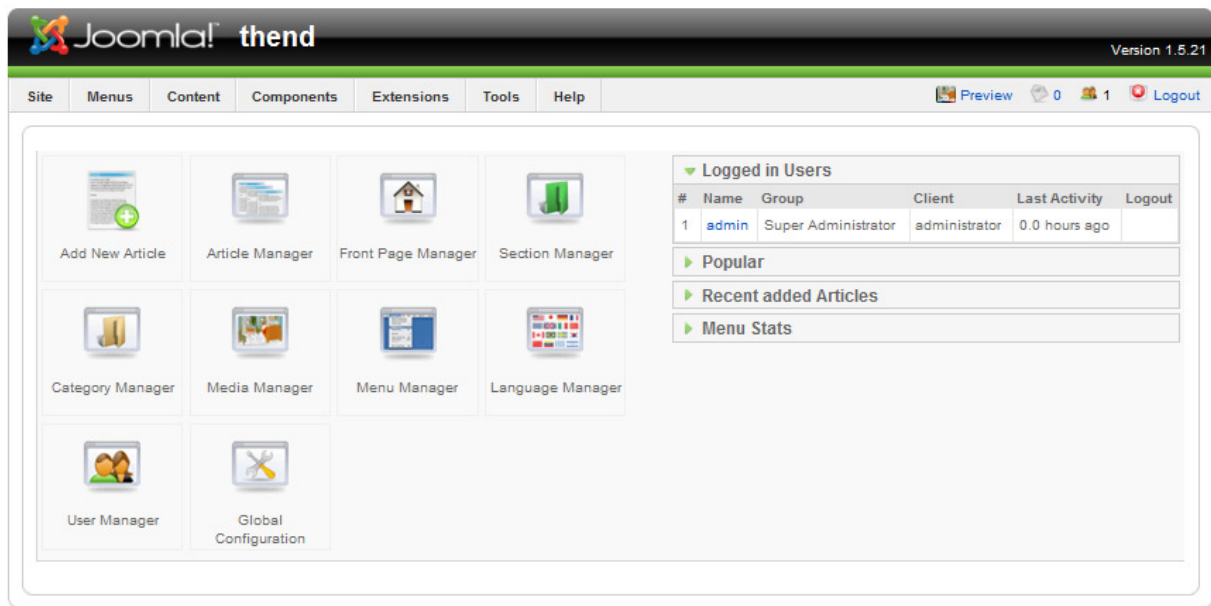
2.2.1.3 Δομή Joomla! (Front End – Back End)

Το **Front End** (παράδειγμα του φαίνεται στην Εικόνα 2) αναφέρεται στην ίδια την ιστοσελίδα, το μέρος δηλαδή στο οποίο έχουν πρόσβαση οι χρήστες / επισκέπτες, όπου εμφανίζεται και το περιεχόμενο της ιστοσελίδας. Το Front End καλείται ανοίγοντας ένα φυλλομετρητή (browser) και εισάγοντας τη διεύθυνση που υπάρχει η ιστοσελίδα (για παράδειγμα <http://localhost/Joomla!site> για τοπική εγκατάσταση Joomla!).



Εικόνα 2: Front End Joomla! Site (joomla.org)

Το **Back End** είναι το interface του administrator (Εικόνα 3), το οποίο χρησιμοποιείται για τη διαχείριση της ιστοσελίδας. Παρέχει ένα πίνακα ελέγχου καθώς και άλλα εργαλεία, απαραίτητα για τη διαχείριση της ιστοσελίδας. Το Back End καλείται ανοίγοντας ένα φυλλομετρητή (browser) και εισάγοντας της διεύθυνση που βρίσκεται το site, προσθέτοντας μια κάθετη και τη λέξη administrator (π.χ. για τοπική εγκατάσταση και πάλι, το administration panel βρίσκεται στη διεύθυνση <http://localhost/Joomla!site/administrator>)³.



Joomla! is Free Software released under the GNU/GPL License.

Εικόνα 3: Joomla! site - Back End

2.2.1.4 Χρήστες & Δικαιώματα

Το Joomla! διαθέτει έτοιμη λειτουργικότητα για τη διαχείριση των χρηστών όπως φαίνεται και στην Εικόνα 4. Πιο συγκεκριμένα παρέχει εργαλεία για τη δημιουργία, την προβολή, επεξεργασία και διαγραφή χρηστών για την ιστοσελίδα.

Registered User: Επισκέπτες που έχουν απλά εγγραφεί στο διαδικτυακό τόπο. Μπορούν να δουν κάποιες επιλογές που ως απλοί επισκέπτες (guests) δεν θα μπορούσαν, αλλά δεν μπορούν να υποβάλουν ή να επεξεργαστούν άρθρα.

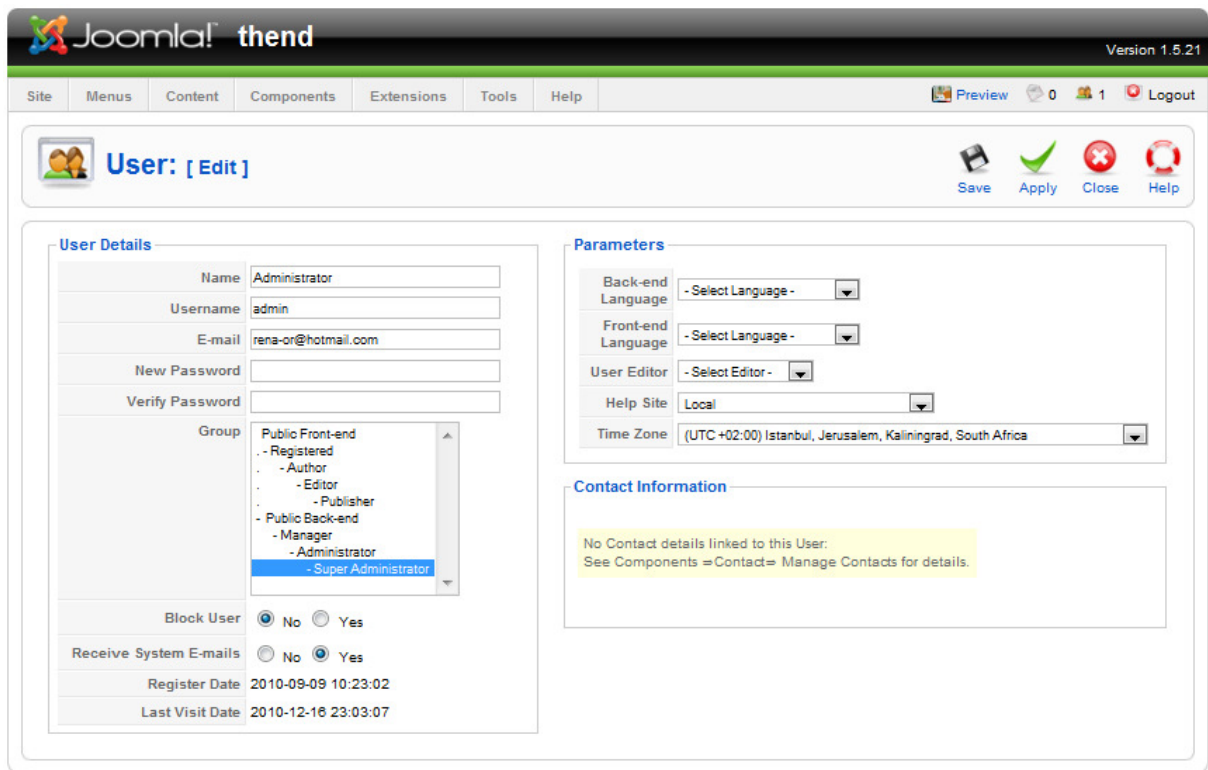
Author: Χρήστες που μπορούν να υποβάλουν άρθρα προς έγκριση. Έγκριση για δημοσίευση γίνεται μόνο από μέλη που ανήκουν στην ομάδα Publisher ή σε υψηλότερη. Επίσης οι χρήστες της ομάδας Author δεν μπορούν να επεξεργαστούν ήδη υπάρχοντα άρθρα (αυτές οι διαδικασίες για τους συγκεκριμένους χρήστες γίνονται μόνο από το front end).

Editor: Χρήστες που μπορούν να υποβάλουν άρθρο ή να επεξεργαστούν κάποιο ήδη υπάρχον. Τα άρθρα αυτά για να δημοσιευθούν, θα πρέπει να πάρουν έγκριση από μέλη που ανήκουν στην ομάδα Publisher ή σε κάποια άλλη με ακόμη περισσότερα δικαιώματα (μόνο front end).

Publisher: Χρήστες που μπορούν να υποβάλουν, να επεξεργαστούν και να δημοσιεύσουν άρθρα (μόνο από το front end).

Manager: Είναι χρήστες, που μπορούν να κάνουν ότι και αυτοί της ομάδας Publisher, αλλά επιπλέον έχουν πρόσβαση στο back end, απ' όπου μπορούν να διαχειριστούν τα αντικείμενα – άρθρα που υπάρχουν στα menu.

Administrator: Επίσης χρήστες που μπορούν να έχουν πρόσβαση τόσο στο front end όσο και στο back end, όπου μπορούν να κάνουν ότι και όσοι ανήκουν στην παραπάνω κατηγορία. Επιπλέον διαχειρίζονται άλλους χρήστες και επεκτάσεις (components, moduls, plugins).



Joomla! is Free Software released under the GNU/GPL License.

Εικόνα 4: Joomla! users Διαχείριση Χρηστών - Απόδοση Δικαιωμάτων

Super Administrator: Τελευταίοι αλλά όχι χειρότεροι. Τουναντίον, είναι χρήστες που εκτός του ότι μπορούν να κάνουν όλα τα παραπάνω, επιπλέον από την είσοδό τους στο back end, έχουν πρόσβαση για την παραμετροποίηση ολόκληρου του site (μπορούν να επεξεργαστούν το περιεχόμενο, επεκτάσεις, templates, γλώσσες χρήστες και έχουν πρόσβαση στο Global Configuration Manager).

Στον παρακάτω πίνακα (Πίνακας 1: Ομάδες Χρηστών και Δικαιώματα χρήσης Joomla! Πίνακας 1), φαίνονται αναλυτικά όλες οι ομάδες – groups και τα δικαιώματα της καθεμιάς³.

	Front-end Groups				Back- end Groups		
	Registered	Author	Editor	Publisher	Manager	Administrator	Super Administrator
View Public Content	✓	✓	✓	✓	✓	✓	✓
View Registered Content	✓	✓	✓	✓	✓	✓	✓
View Special Content	✗	✓	✓	✓	✓	✓	✓
Create new Articles	✗	✓	✓	✓	✓	✓	✓
Edit Own Articles	✗	✓	✓	✓	✓	✓	✓
Edit all Articles	✗	✗	✓	✓	✓	✓	✓
Publish Articles	✗	✗	✗	✓	✓	✓	✓
Access Admin System	✗	✗	✗	✗	✓	✓	✓
Manage Menu Items	✗	✗	✗	✗	✓	✓	✓
Manage Menus	✗	✗	✗	✗	✗	✓	✓
Manage Users	✗	✗	✗	✗	✗	✓	✓
Manage Components, Modules, Plugins	✗	✗	✗	✗	✗	✓	✓
Manage Templates	✗	✗	✗	✗	✗	✗	✓
Manage Language Packs	✗	✗	✗	✗	✗	✗	✓
Access Global Configuration Manager	✗	✗	✗	✗	✗	✗	✓

Πίνακας 1: Ομάδες Χρηστών και Δικαιώματα χρήσης Joomla!⁴

2.2.1.5 Διαχείριση & Δομή Περιεχομένου

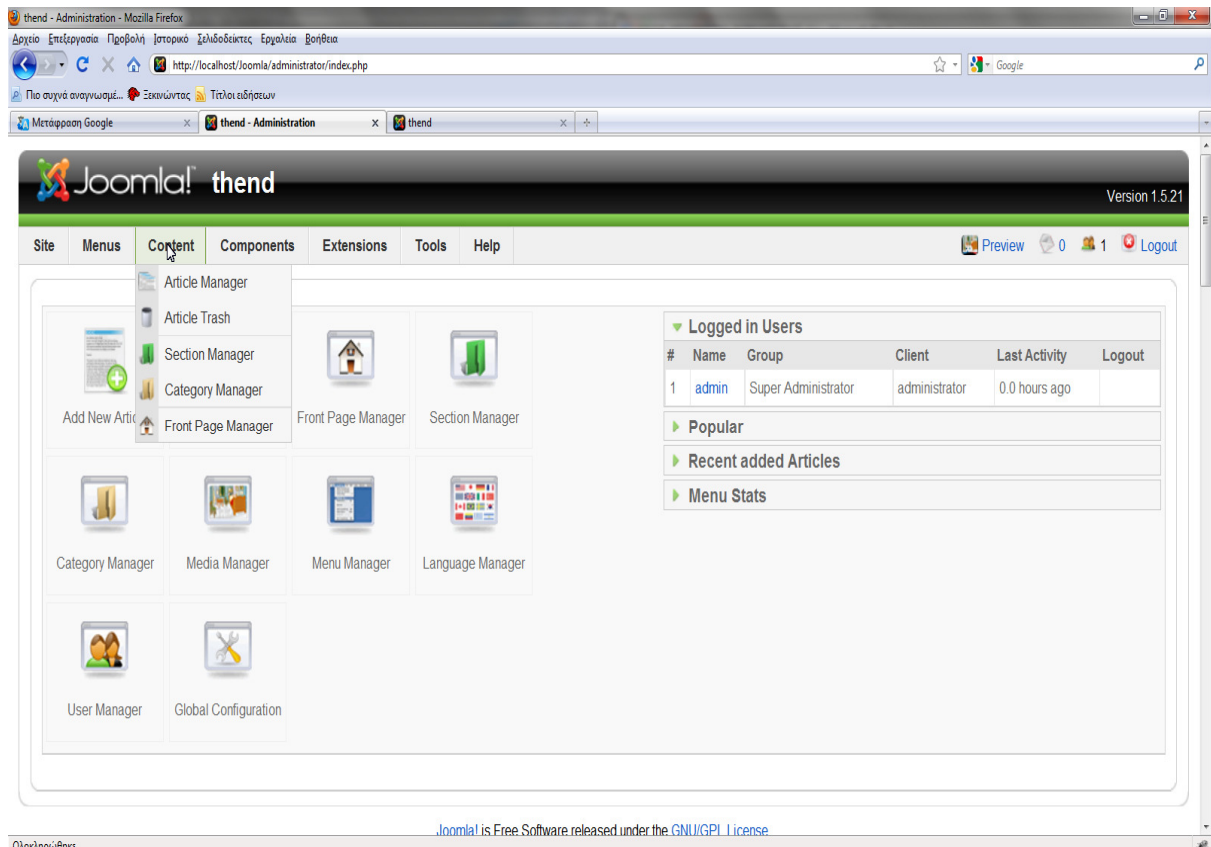
Στα sites, είναι αναγκαίο να χωρίζουμε το περιεχόμενο σε κατηγορίες και υποκατηγορίες για την ευκολότερη και ασφαλέστερη διαχείρισή. Το Joomla! παρέχει αυτή τη δυνατότητα για την ιεραρχική δομή μιας ιστοσελίδας (Εικόνα 5). Συγκεκριμένα έχει τρία επίπεδα ιεράρχησης του περιεχομένου.

Sections: Είναι το ανώτερο επίπεδο στην ιεραρχία. Χωρίζει το περιεχόμενο σε τμήματα. Αν η ιστοσελίδα έχει μικρό όγκο περιεχομένου, μπορεί ένα section να είναι αρκετό. Τα περισσότερα όμως sites, έχουν κάποιο αριθμό από τμήματα, για να ελέγχονται καλύτερα και να υπάρχει μεγαλύτερη ευκολία σχετικά με τον τρόπο εμφάνισης των άρθρων. Πριν δημιουργηθεί περιεχόμενο, θα πρέπει να υπάρχει κάποιο section.

⁴ Wiley, Joomla! Bible, 2010

Categories: Είναι το μεσαίο επίπεδο στην ιεραρχία. Τα categories, υπάγονται σε κάποιο section και το χωρίζουν σε κατηγορίες. Τα categories επίσης χωρίζουν τα περιεχόμενα για τη βελτιστοποίηση του ελέγχου και του τρόπου εμφάνισης του περιεχομένου. Πριν δημιουργηθεί περιεχόμενο, θα πρέπει επίσης να υπάρχει κάποιο category.

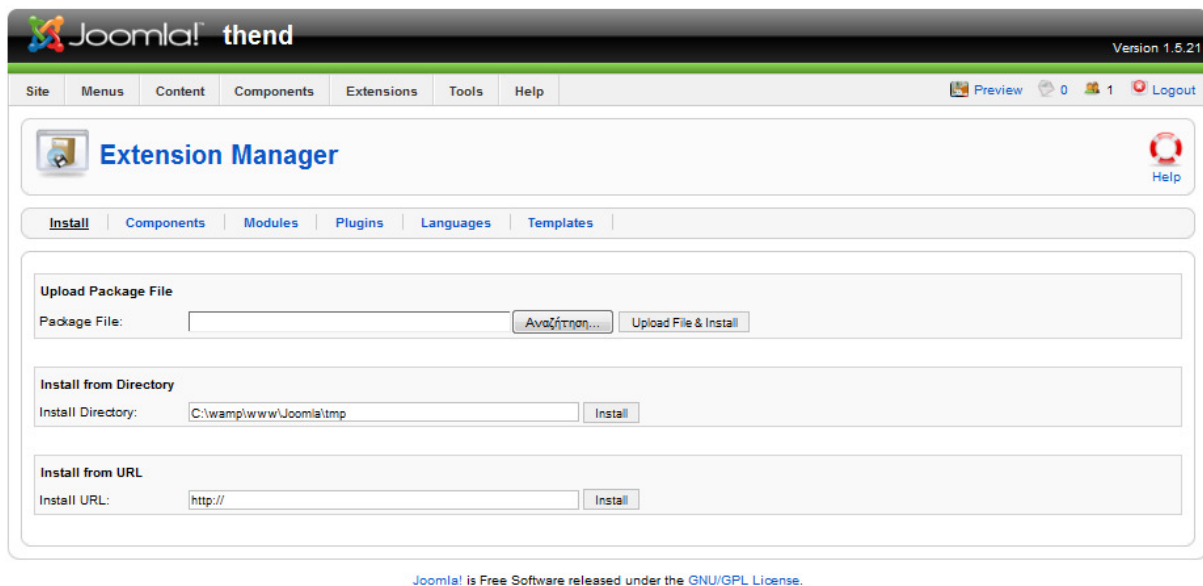
Articles: Τα άρθρα είναι το τελευταίο επίπεδο της ιεραρχίας αυτής. Υπάγονται σε κάποιο category. Στα articles γράφεται το περιεχόμενο που θα δει ο χρήστης από το front end. Στο Joomla! υπάρχει εγκατεστημένος ένας κειμενογράφος τύπου WYSIWYG (What You See Is What You Get), στον οποίο γίνεται η δημιουργία και η επεξεργασία του κειμένου, χωρίς την υποχρέωση γνώσης HTML³.



Εικόνα 5: Joomla! Διαχείριση Περιεχομένου

2.2.1.6 Επεκτάσεις Joomla! (Extensions)

Επεκτάσεις στο Joomla!, είναι πακέτα (packages) που μπορούν να εγκατασταθούν και να επεκτείνουν τη βασική λειτουργικότητα του Joomla!. Υπάρχουν πέντε είδη πακέτων (όπως φαίνεται και παρακάτω στην Εικόνα 6: Joomla! Extensions Management), αυτά είναι:



Εικόνα 6: Joomla! Extentions Management

Components

Τα components είναι ο πιο περίπλοκος τύπος επέκτασης. Είναι ένα είδος μίνι – εφαρμογής. Εμφανίζονται συνήθως στα πλαϊνά του κυρίως σώματος της ιστοσελίδας. Ένα παράδειγμα αυτού, είναι το registration component, το οποίο διαχειρίζεται τις καταχωρήσεις (registrations) των χρηστών.

Modules

Τα modules είναι συνήθως πολύ μικρότερα και λιγότερο πολύπλοκα από τα components. Επίσης εμφανίζονται στα πλαϊνά του κυρίως σώματος της ιστοσελίδας, στις επικεφαλίδες κ.α. Είναι μικρές επεκτάσεις που κάνουν μικρές εργασίες. Ένα παράδειγμα module είναι η φόρμα login, που επιτρέπει στους επισκέπτες της ιστοσελίδας να συνδεθούν σε αυτή.

Plug-ins

Τα Plug-ins είναι ένα ειδικό κομμάτι κώδικα που μπορεί να χρησιμοποιηθεί οπουδήποτε στην ιστοσελίδα, και «τρέχει» όταν φορτώνεται μια σελίδα.

Templates

Τα templates, είναι υπεύθυνα για το τι θα δει ο επισκέπτης σε ένα site, καθώς και τη θέση που θα είναι τοποθετημένα τα στοιχεία προς εμφάνιση. Για παράδειγμα στα templates παρέχονται πληροφορίες για το πώς εμφανίζονται η σελίδες, πως και που εμφανίζονται τα components, τι γραφικά παρουσιάζονται στην ιστοσελίδα κ.α.

Languages

Με την εγκατάσταση πακέτων για διάφορες γλώσσες, παρέχεται η δυνατότητα στο site που έχει φτιαχτεί με Joomla!, όταν θα εμφανίζεται σε χώρες που μιλάνε διαφορετική γλώσσα από την προεγκατεστημένη, να γίνεται μετάφραση, αναλόγως των προτιμήσεων του χρήστη. Όλες οι λέξεις που είναι μέρος του Joomla! (π.χ. το read more) θα εμφανίζεται αυτόματα μεταφρασμένο στην επιλεγμένη γλώσσα³.

2.2.1.7 Παραδείγματα Sites σε Joomla!

Παρακάτω δίδεται η διεύθυνση μερικών ιστοσελίδων που έχει γίνει η ανάπτυξη καθώς και η συντήρησή τους με τη χρήση του Joomla!. Όπως φαίνεται, το CMS αυτό χρησιμοποιείται για blogs, δημόσιους οργανισμούς, e-shops, μεγάλους οργανισμούς κ.α.

- www.gazi.gov.gr
- www.linux.com
- www.moiros.gov.gr
- www.Joomla.org
- www.adespoto.gr
- www.geniki.gr

2.2.2 Drupal

Το Drupal είναι ένα αρθρωτό, δωρεάν και ανοιχτού κώδικα σύστημα διαχείρισης περιεχομένου, γραμμένο στη γλώσσα προγραμματισμού PHP και διανέμεται υπό την GNU General Public Licence. Χρησιμοποιείται ως back-end σύστημα σε τουλάχιστον του 1% όλων των διαδικτυακών τόπων σε όλο τον κόσμο (από προσωπικά blogs σε εταιρικά, πολιτικά, κυβερνητικά κ.α.).

Η τυπική έκδοση του Drupal που είναι γνωστή και ως Drupal core, περιέχει τα βασικά χαρακτηριστικά που είναι κοινά σε όλα τα CMSs, δηλαδή δημιουργία λογαριασμού χρήστη και διατήρηση αυτού, διαχείριση των μενού, RSS-feeds, παραμετροποίηση προβαλλόμενης σελίδας και διαχείριση του συστήματος.



The screenshot shows the Drupal.org homepage. At the top, there's a navigation bar with links like 'Get Started', 'Community & Support', 'Documentation', 'Download & Extend', 'Marketplace', and 'About'. The main header features the Drupal logo and the tagline 'Come for the software, stay for the community'. Below this, there's a search bar and a 'Refine your search' section with radio buttons for 'All', 'Modules', 'Themes', 'Documentation', 'Forums & Issues', and 'Groups'. The main content area is divided into several sections: 'Why Choose Drupal?' with a 'Get Started with Drupal' button, 'Sites Made with Drupal' with a case study link, 'Develop with Drupal' with statistics on modules, themes, and active developers, and a world map showing global usage statistics. A news section on the right highlights the release of Drupal 6.20.

Εικόνα 7: Drupal site (drupal.org)

Εξαιτίας της επεκτασιμότητας του και του αρθρωτού σχεδιασμού του, το Drupal περιγράφεται και ως framework διαδικτυακή εφαρμογή καθώς ανταποκρίνεται στις γενικές υπάρχουσες απαιτήσεις.

Δεν χρειάζονται ιδιαίτερες γνώσεις προγραμματισμού για την αρχική εγκατάσταση και τη βασική διαχείριση του ιστοτόπου.

Το Drupal λειτουργεί σε οποιαδήποτε πλατφόρμα υπολογιστή που υποστηρίζει web server που να χρησιμοποιεί PHP 4.4.0 και μια βάση δεδομένων για την αποθήκευση του περιεχομένου και των ρυθμίσεών του. Η έκδοση Drupal 7, απαιτεί έκδοση PHP 5.2 ή πιο πρόσφατη.⁵

Ο επίσημος διαδικτυακός του τόπος βρίσκεται στη διεύθυνση drupal.org (Εικόνα 7: Drupal site (drupal.org)).

⁵ <http://en.wikipedia.org/wiki/Drupal>

2.2.2.1 Ιστορική Αναδρομή

Το Drupal αρχικά υλοποιήθηκε από τον Dries Buytaert και διανέμεται ως CMS ανοιχτού κώδικα από το 2001.

Drupal είναι μια αγγλική απόδοση της Ολλανδικής λέξης «druppel», που σημαίνει σταγόνα (μια σταγόνα νερού). Το όνομα πάρθηκε από τον πλέον νεκρό οργανισμό Drog.org, του οποίου ο κώδικας σταδιακά εξελίχθηκε σε Drupal. Ο Dries Buytaert, αρχικά ήθελε να δώσει το όνομα «dorp» που στα Ολλανδικά σημαίνει χωριό, αλλά ένα λάθος στη διαδικασία υποβολής του ονόματος (domain name), οδήγησε στην υπάρχουσα ονομασία του Drupal.

Τώρα πλέον, για την ανάπτυξη του Drupal βοηθάει μια ολόκληρη κοινότητα και η δημοτικότητα του αυξάνεται διαρκώς.

Από το Μάιο του 2007 έως και τον Απρίλιο του 2008, το Drupal έχει μεταφορτωθεί από τον επίσημο ιστότοπο του (drupal.org) περισσότερο από 1,4 εκατομμύρια φορές.

Το Drupal χρησιμοποιείται από αρκετά γνωστούς οργανισμούς, συμπεριλαμβανομένων εταιρικών οργανισμών, κυβερνητικών, σχολείων, κ.α.

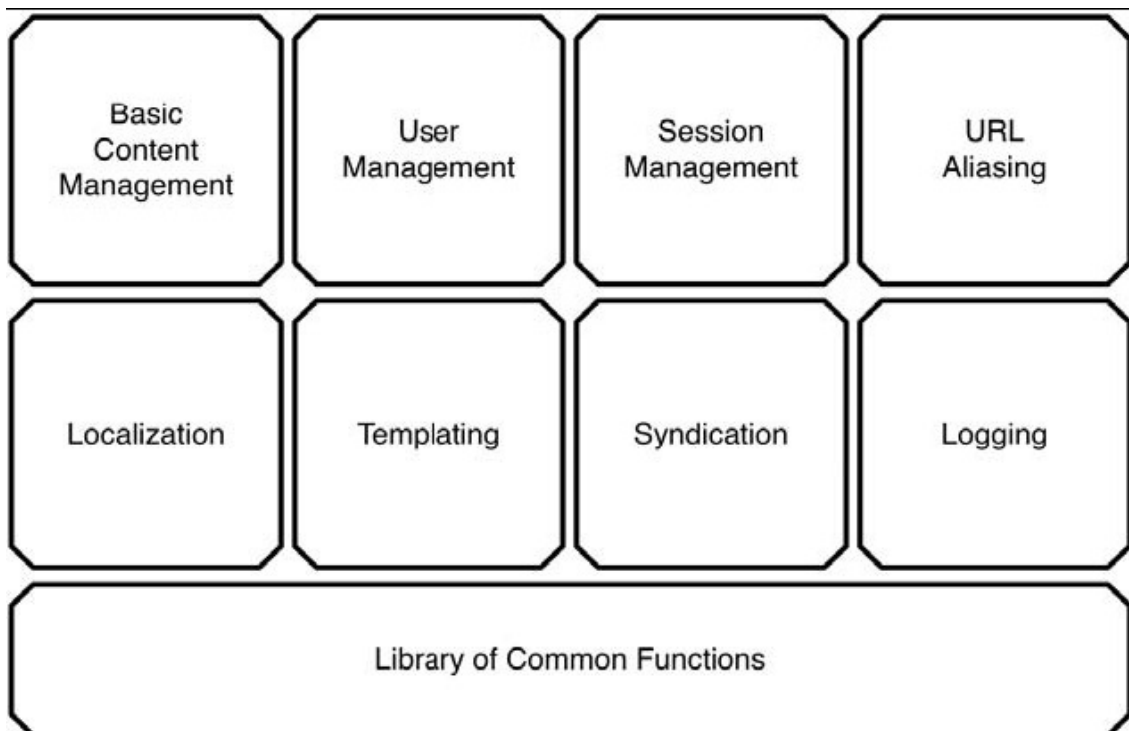
Επίσης έχει κερδίσει αρκετές φορές βραβεία Packt Open Source CMS και τρεις φορές στη σειρά το Webware 100.

Το Drupal 6.19 που κυκλοφόρησε τον Αύγουστο του 2010 αποτελεί την πλέον πρόσφατη έκδοση, ενώ δεν υπάρχει ακόμα ακριβή αναγγελία ημερομηνίας για την διάθεση της τελικής έκδοσης του Drupal 7⁵.

2.2.2.2 Πυρήνας του Drupal

Ο πυρήνας του Drupal, αποτελείται από ένα ελαφρύ πλαίσιο. Αυτό είναι που μεταφορτώνεται από το drupal.org που περιέχει τα αρχεία εγκατάστασης του.

Ο πυρήνας αυτός είναι υπεύθυνος για την παροχή των βασικών λειτουργιών που θα χρησιμοποιηθούν για τη στήριξη άλλων τμημάτων του συστήματος. Εμπεριέχεται επίσης μια βιβλιοθήκη με λειτουργίες που εκτελούνται συχνά καθώς και modules που παρέχουν τη βασική λειτουργικότητα (διαχείριση των χρηστών, ταξινόμηση αρχείων και προτυποποίηση εμφάνισης κ.α.) όπως φαίνεται παρακάτω στην Εικόνα 8.



Εικόνα 8: Drupal core – Πυρήνας του Drupal⁶

⁶ Pact, Pro Drupal Development

2.2.2.3 Χρήστες & Δικαιώματα

Το Drupal έχει ένα σύστημα αδειών, βασισμένο στους ρόλους μέσω του οποίου αποδίδονται τα δικαιώματα στους χρήστες και επιτρέπει να δημιουργούνται διαφορετικές ομάδες χρηστών με διαφορετικές άδειες (Εικόνα 9:Drupal Διαχείριση Χρηστών). Η ενότητα αυτή ερευνά πως δημιουργούνται οι ρόλοι και η εκχώρηση δικαιωμάτων σε αυτούς. Επίσης εξετάζονται σε αυτό το τμήμα οι κανόνες πρόσβασης που μπορούν να ρυθμιστούν, ώστε να προσδιορίζονται ποιοι μπορούν να δημιουργήσουν λογαριασμούς, βάση username, e-mail, ή του διαδικτυακού σημείου που επισκέπτονται τη σελίδα.

Το Drupal προσπαθεί να προσφέρει λεπτομερή έλεγχο στην πρόσβαση του περιεχομένου και την εκτέλεση των δράσεων. Είναι σημαντικό, ότι ο διαχειριστής του site μπορεί να αποφασίσει ακριβώς τι μπορεί να κάνει ο κάθε χρήστης και τι μπορεί να δει. Για να μπορεί να υποστηριχτεί αυτή η λειτουργία, έχουν εκχωρηθεί σε όλους τους χρήστες ρόλοι και δικαιώματα.

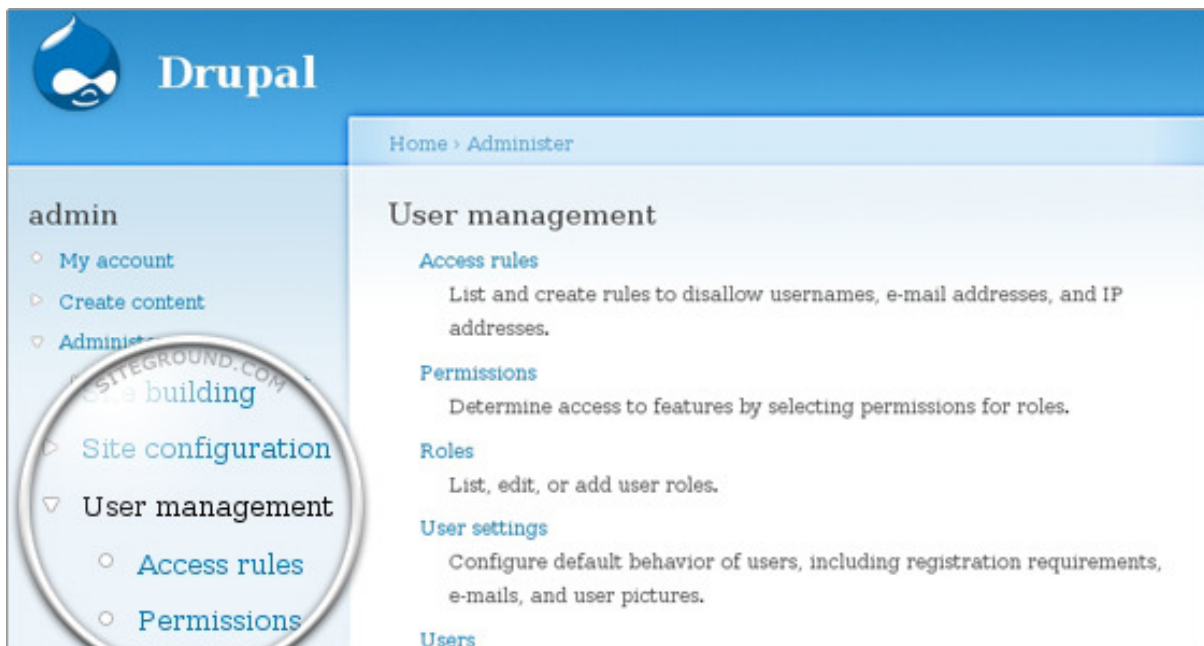
Ένας ρόλος περιγράφει το προφίλ ενός χρήστη ή το σενάριο χρήσης από ένα χρήστη ή από μια ομάδα χρηστών. Οι ρόλοι που δημιουργούνται με την εγκατάσταση του Drupal είναι οι εξής:

Anonymous User: Είναι ρόλος χρηστών που ανατίθεται σε κάθε επισκέπτη που δεν έχει λογαριασμό ή δεν έχει εισέλθει ως επισκέπτης.

Authenticated User: Ρόλος που καταχωρούνται οι συνδεδεμένοι χρήστες. Με την δημιουργία νέου λογαριασμού, ο χρήστης μετατίθεται αυτόματα σε αυτό το ρόλο.

Επίσης μπορούν να δημιουργηθούν από το διαχειριστή νέοι ρόλοι και εκεί να τοποθετούνται χρήστες ή ομάδες χρηστών.

Η λίστα των ρόλων βρίσκεται στο μονοπάτι `administer` → `access control` → `roles`⁷.



Εικόνα 9:Drupal Διαχείριση Χρηστών⁸

2.2.2.4 Επεκτάσεις Drupal core(Extensions)

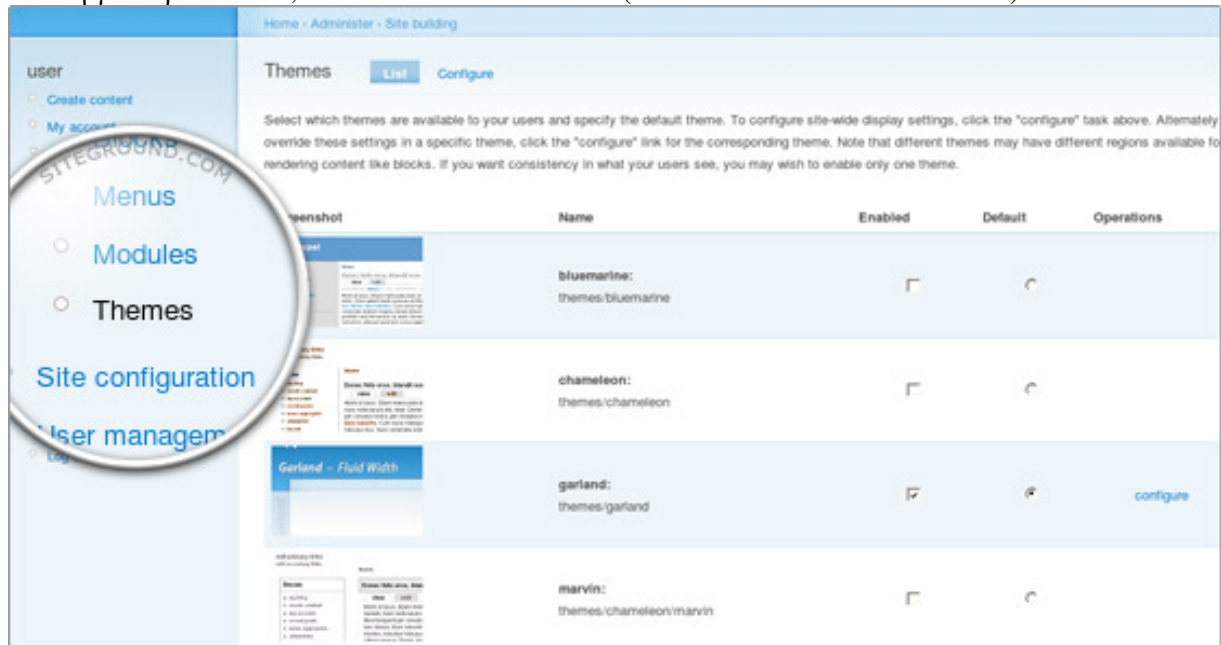
Καθώς ο πυρήνας του Drupal αποτελείται από υπομονάδες, ορίζει ένα σύστημα που μπορεί να είναι προσβάσιμο στο εσωτερικό του μέσω των API. Αυτός ο σχεδιασμός επιτρέπει τη συμβολή από έμμεσα εμπλεκόμενους με modules και themes. Μπορούν να επεκτείνουν ή να πανωγράψουν και να

⁷ Apress, Building Online Communities with Drupal, phpBB and Wordpress, 2006

⁸ <http://www.siteground.com/tutorials/drupal-tutorial>

μετατρέψουν την προκαθορισμένη συμπεριφορά του Drupal, χωρίς να είναι ανάγκη να επεξεργαστούν τον εσωτερικό του κώδικα (Εικόνα 10: Drupal Διαχείριση Επεκτασιμότητας).

Modules: Είναι αρχεία που επεκτείνουν τη λειτουργικότητα του Drupal. Παράδειγμα τέτοιων αρχείων είναι η γκαλερί εικόνων, editors τύπου WYSIWYG (What You See Is What You Get).



Εικόνα 10: Drupal Διαχείριση Επεκτασιμότητας⁸

Themes: Η συνεισφορά των θεμάτων (themes) εντοπίζεται στην αντικατάσταση και την προσαρμογή του προεπιλεγμένου τρόπου εμφάνισης και αισθητικής ενός site⁵.

2.2.2.5 Παραδείγματα Sites σε Drupal

whitehouse.gov
data.gov.uk
ubuntu.com
economist.com

2.2.3 WordPress

Το WordPress είναι ένα ανοιχτού κώδικα CMS, το οποίο χρησιμοποιείται συχνά για τη δημιουργία blogs. Είναι σχεδιασμένο με τις γλώσσες προγραμματισμού PHP και MySQL. Διαθέτει αρκετά χαρακτηριστικά, μεταξύ αυτών αρχιτεκτονική plug-ins και ένα σύστημα προτύπων (templates). Χρησιμοποιείται περίπου από το 12% ανάμεσα σε 1.000.000 από τις μεγαλύτερες ιστοσελίδες. Αυτά τα ποσοστά, κατατάσσουν το WordPress σε ένα από τα πιο γνωστά CMSs⁹.

Ο επίσημος ιστότοπος του WordPress βρίσκεται στη διεύθυνση wordpress.org

2.2.3.1 Ιστορική Αναδρομή

Η πρώτη έκδοση του WordPress, κυκλοφόρησε στις 27 Μαΐου 2003 αποτέλεσμα της κοινής προσπάθειας του Matt Mullenweg και του Mike Little.

Το 2004 οι όροι χορήγησης αδειών για τη συμμετοχή στο πακέτο Movable Type άλλαξαν από τη Six Apart (εταιρεία λογισμικού, δημιουργός του Movable Type), και πολλοί από τους χρήστες μεταφέρθηκαν στο WordPress δημιουργώντας έτσι μια αξιοσημείωτη και διαρκή αύξηση της δημοτικότητας του WordPress.

Το 2007 κερδίζει το βραβείο Packt Open Source CMS

Τον Οκτώβριο του 2009, το 2009 Open Source CMS Market Share Report, καταλήγει στο συμπέρασμα ότι το WordPress αποτελεί τη μεγαλύτερη εμπορική δύναμη από οποιοδήποτε άλλο

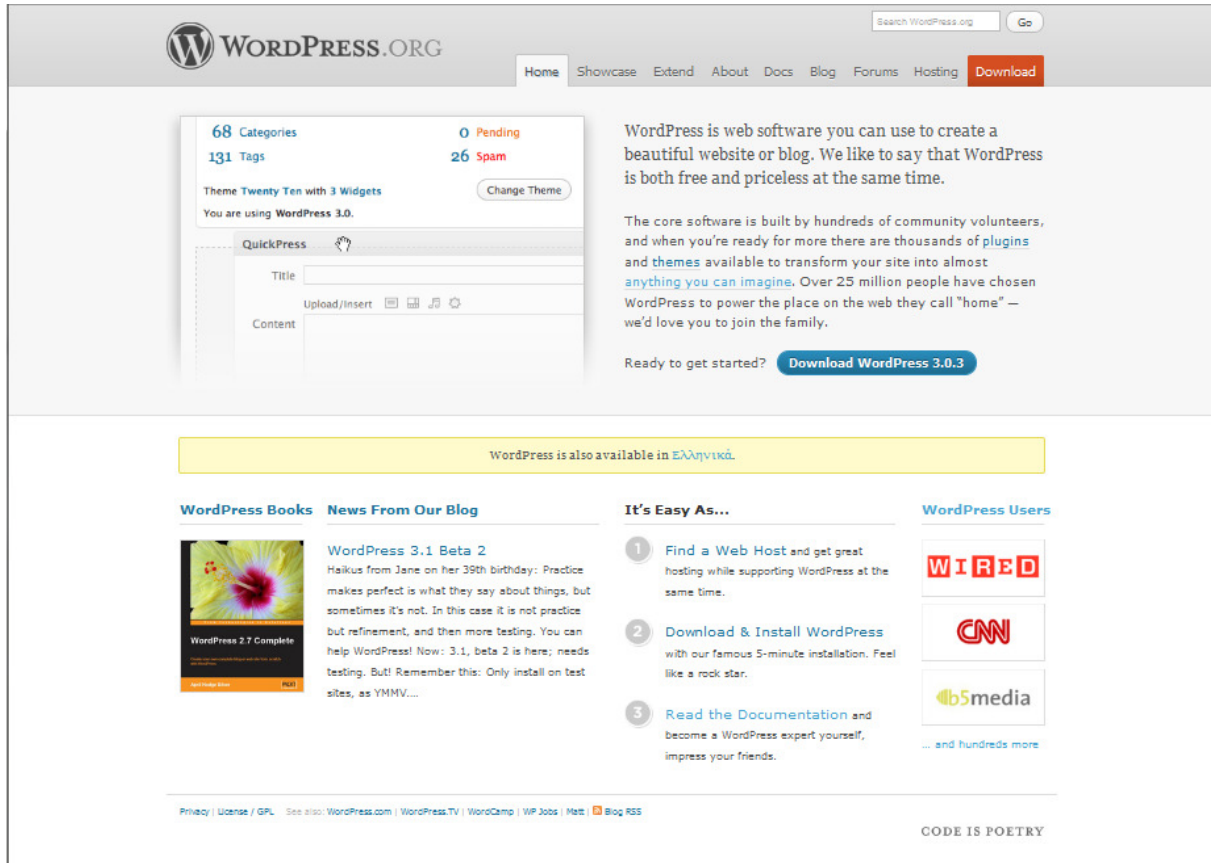
⁹ <http://en.wikipedia.org/wiki/WordPress>

Σύστημα Διαχείρισης Περιεχομένου. Το συμπέρασμα αυτό, βασίζεται σε μια εκτενή ανάλυση του ποσοστού της υιοθέτησης του προτύπου και της εμπορικής δύναμης κατόπιν έρευνας των χρηστών.

Το 2009 επίσης, κατακτά το βραβείο του καλύτερου Open Source CMS.

Αξιοσημείωτο είναι ότι από τον Αύγουστο του 2010, η έκδοση 3.0 έχει μεταφορτωθεί περίπου 12,5 εκατομμύρια φορές⁹.

2.2.3.2 Δομή WordPress (Front Page – Dashboard)

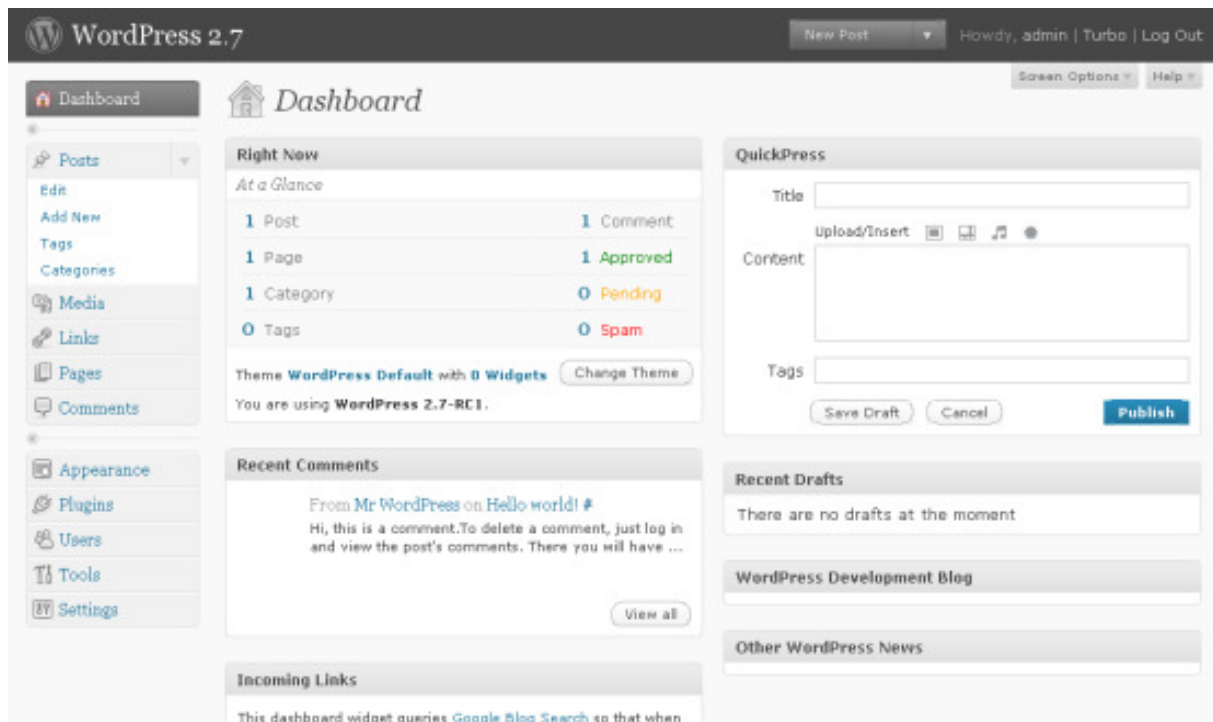


Εικόνα 11: Front End WordPress Site (wordpress.org)

Το **Front Page**: Είναι η περιοχή στην οποία εμφανίζεται ο ιστότοπος ή το ιστολόγιο (αναλόγως των σκοπών που ικανοποιεί) και είναι προσβάσιμο από όλους τους χρήστες του διαδικτύου σε όλο τον κόσμο (εικόνα 11). Για τις ρυθμίσεις του, τι εμφανίζεται, σε ποιο σημείο του site και σε ποιους χρήστες χρησιμοποιείται το Dashboard.

Το **Dashboard**: Είναι η περιοχή διαχείρισης του WordPress, απ' όπου γίνονται οι αλλαγές που χρειάζονται να εμφανιστούν στην ιστοσελίδα (Εικόνα 12: WordPress Dashboard). Η σελίδα αυτή εμφανίζεται ανοίγοντας ένα φυλλομετρητή (browser), γράφοντας τη διεύθυνση που είναι αποθηκευμένο το site και πληκτρολογώντας στο τέλος /wp-admin. Για παράδειγμα σε μία τοπική εγκατάσταση, το Dashboard βρίσκεται στη διεύθυνση `http://localhost/WordPress's site name/wp-admin`¹⁰.

¹⁰ Wrox. WordPress, 24 Hour Trainer, Nov. 2009



Εικόνα 12: WordPress Dashboard¹¹

2.2.3.3 Χρήστες & Δικαιώματα

Το WordPress υποστηρίζει τη διαχείριση των χρηστών, καθώς και των δικαιωμάτων που έχουν στην ιστοσελίδα, δίνοντας διαφορετικό επίπεδο προσβασιμότητας σε κάθε ένα από αυτούς, αναλόγως το role που ανήκουν.

Τύποι User Roles

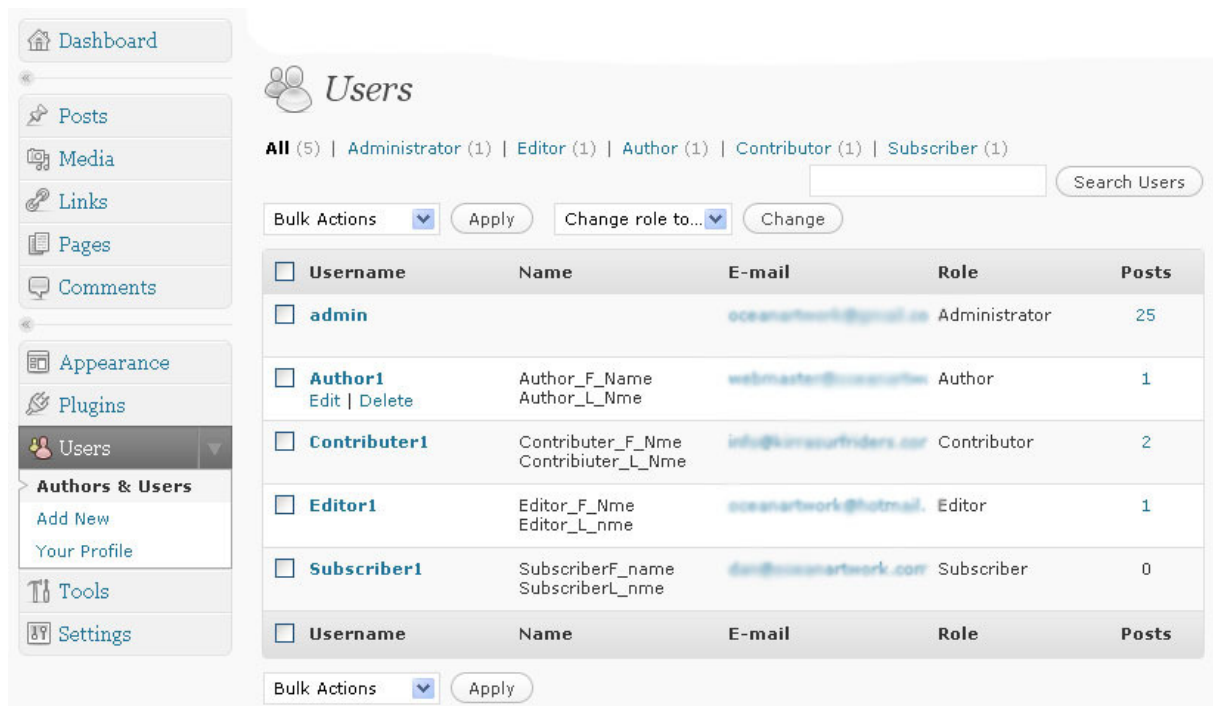
Οι ρόλοι που είναι χωρισμένοι οι χρήστες, είναι οι παρακάτω:

Administrator: Όπως παραπέμπει και το όνομα του ρόλου, ο Administrator έχει πρόσβαση σε οποιαδήποτε διαχειριστική λειτουργία στο WordPress, συμπεριλαμβανομένου και τις επεξεργασίας αρχείων του θέματος, αλλαγής θέματος σελίδας, στην επεξεργασία των Plugins, στις λεπτομέρειες των χρηστών κτλ.

Editor: Ένας χρήστης που ανήκει στην κατηγορία του Editor, μπορεί να διαχειριστεί σχεδόν όλα τα περιεχόμενα του site, εκτός από τις ρυθμίσεις ελέγχου του (όπως για παράδειγμα το theme ή τα plugins). Ο Editor μπορεί να προσθέσει ή να διαγράψει οποιοδήποτε περιεχόμενο σχετίζεται με το WordPress συμπεριλαμβανομένων των κατηγοριών, των posts των σελίδων και των συνδέσμων. Επίσης έχει πλήρη πρόσβαση στον τόπο αποθήκευσης των media και πλήρη έλεγχο στην δημοσίευση περιεχομένου. Τέλος οι editors μπορούν να προσθέτουν και να διαγράφουν users, αλλά δεν μπορούν να επεξεργάζονται τις πληροφορίες που είναι αποθηκευμένες για τον κάθε χρήστη.

Author: Οι χρήστες που ανήκουν στην κατηγορία author, είναι ότι ένας αρθρογράφος για μια εφημερίδα ή ένα περιοδικό. Έχει πλήρη έλεγχο στα δικά του περιεχόμενα (πρόσθεση – επεξεργασία – δημοσίευση – διαγραφή), αλλά δεν έχει πρόσβαση στα περιεχόμενα κανενός άλλου χρήστη, επίσης δεν μπορεί να συνδέσει άλλα categories και τέλος δεν μπορεί να χρησιμοποιήσει μη φιλτραρισμένη HTML στα posts του (κώδικας από Javascript ή παρόμοιων HTML tags ή χαρακτηριστικά, πρέπει να περαστούν πρώτα από κάποιο πρόγραμμα όπως το Dreamweaver).

¹¹ <http://www.blogsdna.com/1601/wordpress-27-release-candidate-1-rc-1-released-with-new-icon-set-download-now.htm>



Εικόνα 13: WordPress Διαχείριση Χρηστών¹²

Contributor: Οι contributors, μπορούν να δημιουργήσουν, να επεξεργαστούν και να διαγράψουν τα δικά τους posts, αλλά δεν μπορούν να τα δημοσιεύσουν (αποθηκεύονται ως πρόχειρα αρχεία). Επίσης, μετά τη δημοσίευση των posts από κάποιο editor ή κάποιο administrator, δεν μπορούν να επεξεργαστούν ούτε τα δικά τους περιεχόμενα.

Subscriber: Στην περίπτωση αυτή, η λέξη Subscriber χρησιμοποιείται με την έννοια του εγγεγραμμένου χρήστη. Τα μόνα δικαιώματα που έχει, είναι να μπορεί να δει περιεχόμενα, ή να κάνει κάποια πράγματα που δεν μπορεί κάποιος μη εγγεγραμμένος χρήστης και να επεξεργαστεί της αποθηκευμένες προσωπικές του πληροφορίες.

Ο ρόλος του κάθε χρήστη αποδίδεται από το Dashboard του WordPress όπως φαίνεται στην παραπάνω (Εικόνα 13: WordPress Διαχείριση Χρηστών) και μόνο από κάποιο χρήστη που έχει το ρόλο του administrator¹⁰.

2.2.3.4 Επεκτάσεις WordPress (Extensions)

Plugins: Είναι scripts (κομμάτια κώδικα) που παρέχουν επιπρόσθετες λειτουργίες στο WordPress, χρησιμοποιώντας ένα συνδυασμό από έτοιμα εγκατεστημένα κομμάτια κώδικα και καινούριο κώδικα. Η πολυπλοκότητά τους ποικίλει. Μπορεί να είναι από μερικές γραμμές κώδικα έως και φάκελοι με δεκάδες αρχεία scripts και πίνακες βάσεων δεδομένων.

Τα περισσότερα από αυτά, είναι ανοιχτού κώδικα και διαθέσιμα χωρίς κάποιο αντίτιμο (υλοποιούνται συνήθως για να ικανοποιήσουν κάποια εξειδικευμένη ανάγκη κάποιου και στη συνέχεια κοινοποιούνται και διαμοιράζονται μέσω της κοινότητας του WordPress).

Ένα από τα πιο σημαντικά πλεονεκτήματα των plugins, είναι ότι μπορούν να αποσυνδεθούν. Αυτό είναι πολύ σημαντικό για την προστασία της βάσης δεδομένων και του site, διατηρώντας τα plugins ξεχωριστά από τα υπόλοιπα αρχεία του WordPress. Αν κάτι πάει στραβά λόγω σύγκρουσης με κάποιο άλλο plugin, είτε λόγω αναβάθμισης του WordPress, το πρόβλημα λύνεται απλά με την απενεργοποίηση του συγκεκριμένου plugin.

Themes: Είναι ένα σύνολο αρχείων που ελέγχει την παρουσίαση – εμφάνιση των προκαθορισμένων θέσεων των περιεχομένων που έχει δημιουργήσει το WordPress σε σελίδες HTML.

¹²http://webdesign.oceanartwork.com/wp-content/uploads/2009/07/manage_users.jpg

Κάθε θέμα περιέχει τουλάχιστον ένα αρχείο μορφοποίησης (style sheet) και τουλάχιστον ένα αρχείο Template.

Templates: Είναι ένα αρχείο που περιέχεται στο φάκελο themes του WordPress και παρέχει οδηγίες για το συντονισμό (τη μορφή που θα εμφανίζονται) οι HTML σελίδες. Τα Templates, μπορεί να χρησιμοποιήσουν άλλα Templates στη διαδικασία του συντονισμού. Τα templates περιέχουν widgets¹⁰.

2.2.3.5 Παραδείγματα Sites σε WordPress

www.thefordstory.com
tdi.vw.com
politicalticker.blogs.cnn.com
www.xblog.gr
www.greecetraveladvisor.com

2.2.4 Σύγκριση των CMSs

Παραπάνω έχουμε αναφερθεί σε τρία διαφορετικά Συστήματα Διαχείρισης Περιεχομένου, όπου και τα τρία έχουν κερδίσει βραβεία, αναπτύσσονται διαρκώς, υποστηρίζονται από πολύ μεγάλες κοινότητες και υπόσχονται εν γένει πολλά. Το ερώτημα είναι πιο από αυτά είναι το καλύτερο για την υλοποίηση ενός site; Η σωστή απάντηση σε αυτό το ερώτημα αυτό, δεν υπάρχει. Αναλόγως του είδους του site που χρειάζεται να δημιουργηθεί (πχ. e-shop, φόρουμ κτλ.) αποφασίζεται το προς χρήση CMS. Για το λόγο αυτό, θα πραγματοποιηθεί μια έρευνα ώστε να εντοπιστούν τα καλύτερα μέρη από αυτά σε κάθε κατηγορία (για τη σύγκριση που ακολουθεί, έχει γίνει χρήση πληροφοριών αυτών των τριών CMS από το site www.cmsmatrix.org και οι εκδόσεις που λαμβάνουν μέρος είναι το Drupal 6.10, το Joomla! 1.5.10, το WordPress 2.2.1).

Κοινά: Είναι δωρεάν εφαρμογές ανοιχτού κώδικα και έχουν υλοποιηθεί με γλώσσα προγραμματισμού PHP.

Απαιτήσεις Συστήματος: Τα τρία αυτά CMSs, έχουν κοινές απαιτήσεις για τη λειτουργία τους σε κάποιο σύστημα. Χρειάζονται όλα ένα Apache server και MySQL βάση δεδομένων, ενώ δεν έχουν καμία απαίτηση για το λειτουργικό σύστημα που χρησιμοποιείται, καθώς συνεργάζονται με όλα.

Δυνατότητες ασφαλείας: Κοινά σε αυτό το επίπεδο έχουν την έγκριση πριν τη δημοσίευση του περιεχομένου, τη συμβατότητα με το πρωτόκολλο SSL (πρωτόκολλο για ασφαλή διακίνηση δεδομένων στο δίκτυο) και τη διασφάλιση πιστοποίησης χρήστη. Παρόλο που είναι συμβατά με το πρωτόκολλο SSL, το Drupal δεν διασφαλίζει τη σύνδεση ενός χρήστη χρησιμοποιώντας το, το WordPress, για να υποστηρίξει αυτή τη λειτουργικότητα πρέπει να προστεθεί ένα plugin ενώ για το Joomla! προσφέρεται με την αρχική εγκατάσταση του. Επίσης το Drupal και το Joomla! διαχειρίζονται πληροφορίες του χρήστη μετά την είσοδο του (προσφέρεται από την αρχική εγκατάστασή τους), ενώ για το WordPress αυτό είναι μια πρόσθετη λειτουργία. Τέλος το Joomla! και το Drupal κρατάνε ιστορικό για τα logins του χρήστη, ενώ το Wordpress χρειάζεται να επεκταθεί για να το υποστηρίξει αυτό.

Ευκολία Χρήσης: Το WordPress, στο συγκεκριμένο τομέα υπερέχει ελαφρά κυρίως γιατί έχει ενσωματωμένη τη λειτουργία του Drag N- Drop. Το Drupal μπορεί να προσφέρει τη δυνατότητα αυτή εγκαθιστώντας ένα module, ενώ το Joomla! δεν το υποστηρίζει. Προσφέρουν και τα τρία φιλικά URLs, παρέχουν τη δυνατότητα χρήσης WYSIWYG editors (για το Drupal είναι πρόσθετο), ενώ μια από τις πλέον χρησιμοποιούμενες ενέργειες, η αναιρέση δεν υποστηρίζεται σχεδόν από κανένα τους, εκτός του WordPress μετά την εγκατάσταση του plugin αυτού.

Ενσωματωμένη λειτουργικότητα: Κάθε ένα από αυτά, στην αρχική του εγκατάσταση, τοποθετεί κάποιες πρόσθετες λειτουργίες για τη διευκόλυνση της χρήσης του.

Ποιο συγκεκριμένα με την εγκατάσταση του Drupal εγκαθίστανται και οι εξής λειτουργίες:

- ❖ Χρήση blog
- ❖ Φόρουμ – τόπο συζητήσεων
- ❖ FAQ διαχείριση
- ❖ Δημοσκοπήσεις
- ❖ Μηχανή αναζήτησης

- ❖ RSS

Το Joomla! στην αρχική του εγκατάσταση περιέχει

- ❖ Χρήση blog
- ❖ Διαχείριση επικοινωνίας
- ❖ FAQ διαχείριση
- ❖ Διαχείριση συνδέσμων
- ❖ Φόρμα αποστολής email
- ❖ Δημοσκοπήσεις
- ❖ Διαχείριση προϊόντων
- ❖ Μηχανή αναζήτησης
- ❖ RSS
- ❖ Διαδικτυακές υπηρεσίες για το front end

Ενώ το WordPress περιέχει:

- ❖ Χρήση blog
- ❖ Διαχείριση συνδέσμων
- ❖ Σελίδα Dashboard
- ❖ Μηχανή αναζήτησης
- ❖ RSS

Ακόμα και αν δεν προσφέρεται κάποια έτοιμη λειτουργικότητα με την αρχική εγκατάσταση των CMS, μπορούν να προστεθούν μετέπειτα. Ενδεικτικά κάποιες εύχρηστες εφαρμογές που δεν προσφέρονται σε κανένα από τα παραπάνω στην αρχική εγκατάσταση, είναι τα εξής: εμφάνιση καιρού, χάρτης ιστοσελίδας, φωτογραφικό άλμπουμ τεστ κτλ.

Τέλος, βασικό μειονέκτημα έχει το WordPress σε σύγκριση με τα άλλα δύο, καθώς δεν υποστηρίζει επεκτάσεις εμπορίου (commerce). Σε αντίθεση, το Drupal και το Joomla! έχουν ανεπτυγμένες ικανοποιητικές εφαρμογές για δημιουργία e-shops, που χρειάζονται απλά εγκατάσταση και παραμετροποίηση (βάση καταστήματος).

2.3 Περίληψη Κεφαλαίου

Στο κεφάλαιο αυτό, μελετήθηκαν και συγκρίθηκαν τα τρία πλέον δημοφιλή και χρησιμοποιούμενα Συστήματα Διαχείρισης Περιεχομένου ανοιχτού κώδικα. Από αυτά, στα επόμενα κεφάλαια θα χρησιμοποιήσουμε το Joomla! καθώς τα τελευταία χρόνια έχει μια διαρκή και σταθερή ανοδική πορεία και στο άμεσο μέλλον αναμένεται να κυκλοφορήσει μια έκδοση (την 1.6 που βρίσκεται σε στάδιο ελέγχου).

ΚΕΦΑΛΑΙΟ 3

Issue Tracking Systems: Ανασκόπηση Τεχνολογιών

3.1 Εισαγωγή

Στο κεφάλαιο αυτό, γίνεται έρευνα σχετικά με τα υπάρχοντα συστήματα παρακολούθησης ζητημάτων. Η ενέργεια αυτή είναι πολύ χρήσιμη, καθώς παρέχει πληροφόρηση και ενημέρωση για τις υπάρχουσες εφαρμογές.

3.2 Issue Tracking Systems

Με τον όρο Issue Tracking System εννοούμε ένα σύστημα διαχείρισης ζητημάτων, τη διαδικασία δηλαδή και τις ενέργειες που γίνονται ακολουθώντας συγκεκριμένη ροή εργασιών, μετά τη δημιουργία ενός ζητήματος έως ότου επιτευχθεί ο τελικός στόχος που δεν είναι άλλος από την ορθή και πλήρη επίλυση αυτού.

Κυκλοφορούν ήδη κάποιες έτοιμες εφαρμογές. Κάποιες από αυτές είναι επεκτάσεις του Joomla! ενώ άλλες είναι αυτόνομες. Μπορούν να κατηγοριοποιηθούν σε Web Based (δηλαδή η διαχείρισή τους και η ενημέρωσή τους βασίζεται στο διαδίκτυο) και σε αρχεία που απαιτούν εγκατάσταση για τη λειτουργία τους (.exe αρχεία). Στην ενότητα αυτή θα γίνει μια έρευνα για κάποιες που επεκτείνουν το Joomla! και παρουσίαση για κάποιες εφαρμογές άμεσου ενδιαφέροντος, δηλαδή είτε επεκτείνουν το Joomla! ή είναι Web Based.

3.3 Ροή Εργασίας Issue Tracking Systems

Τα Issue Tracking Systems υλοποιούν μια ροή εργασιών (workflow) η οποία υποστηρίζει τη βασική διαδοχή εργασιών ώστε να επιτευχθεί ο τελικός στόχος, ο οποίος δεν είναι άλλος από την ικανοποίηση του πελάτη. Παρακάτω παρουσιάζονται τα βασικά βήματα μιας ροής εργασιών την οποία ακολουθούν τα περισσότερα συστήματα. Αν και τα βήματα αυτά συνήθως ακολουθούνται, ενδέχεται να υπάρξουν κάποιες διαφοροποιήσεις, ανάλογα του αντικειμένου ενασχόλησης κάθε εταιρείας καθώς και των διεργασιών που έχουν θεσπιστεί.

Βήμα 1: Ο υπάλληλος (τεχνικός, υπάλληλος υποδοχής κλήσεων ή κάποιος άλλος εργαζόμενος της εταιρείας) λαμβάνει ένα τηλεφώνημα, e-mail ή κάποιας άλλης μορφής ενημέρωση από τον πελάτη σχετικά με κάποιο ζήτημα (τεχνικό, θέμα λογιστηρίου, θέμα πωλήσεων κ.α.).

Βήμα 2: Στη συνέχεια, ο υπάλληλος ελέγχει την ορθότητα του ζητήματος, καθώς και τις απαιτήσεις που υπάρχουν για αυτό.

Βήμα 3: Έπειτα ο υπάλληλος δημιουργεί το ζήτημα στο σύστημα, εισάγοντας όλα τα σχετικά στοιχεία όπως παρουσιάστηκαν από τον πελάτη.

Βήμα 4: Το βήμα αυτό είναι επαναλαμβανόμενο. Μετά τη δημιουργία του ζητήματος στο σύστημα, θα πρέπει να γίνεται συνεχής ενημέρωση για αυτό. Δηλαδή κάθε εργασία που γίνεται για την επίλυση του ζητήματος, θα πρέπει να καταχωρείται στο Issue Tracking System.

Βήμα 5: Μετά την ολοκλήρωση του ζητήματος, αυτό σημειώνεται ως «επιλυμένο» ή «ολοκληρωμένο» ή όπως έχει οριοθετηθεί από την εκάστοτε επιχείρηση και εξακολουθεί να υπάρχει στο Issue Tracking System για ιστορικούς και στατιστικούς λόγους. Αν το ζήτημα δεν έχει επιλυθεί, θα πρέπει να σημειωθεί ως μη ολοκληρωμένο.

Βήμα 6: Στην περίπτωση ολοκλήρωσης του ζητήματος, θα πρέπει να γίνει ενημέρωση του πελάτη για αυτό, καθώς επίσης για την ορθή λειτουργία, καλό θα είναι να ενημερωθεί το issue tracking system για το περιεχόμενο της συζήτησης (μεταξύ υπαλλήλου – πελάτη).

Τέλος ολοκληρωμένο θεωρείται ένα issue στο οποίο έχουν ολοκληρωθεί όλες οι απαιτούμενες εργασίες και έχει γίνει η παράδοση στον τελικό χρήστη (πελάτη)¹³.

3.4 Αρχιτεκτονική Issue Tracking System

Μια βάση δεδομένων, είναι απαραίτητη καθώς είναι ο χώρος στον οποίο γίνεται η αποθήκευση όλων των δεδομένων. Τα δεδομένα αυτά, διαχειρίζονται από το επίπεδο της επιχειρηματικής λογικής της εφαρμογής. Επιχειρηματική λογική (business logic) είναι ένας μη τεχνικός όρος που χρησιμοποιείται γενικά για την περιγραφή των λειτουργικών αλγορίθμων, που χειρίζονται την ανταλλαγή πληροφοριών μεταξύ της βάσης δεδομένων και της διεπαφής του χρήστη. Το επίπεδο αυτό (business logic layer) διαμορφώνει τα πρωτογενή δεδομένα βάση της χρησιμοποιούμενης δομής για να είναι πλήρη και κατανοητά για τη διεκπεραίωσή τους.

Για ασφάλεια, τα Issue Tracking Systems, πιστοποιούν τους χρήστες πριν τους επιτραπεί η είσοδος στο σύστημα¹³.

3.5 Παραδείγματα Issue Tracking Systems

Παρακάτω, παρουσιάζονται κάποια υπάρχοντα συστήματα διαχείρισης ζητημάτων, κάποια από αυτά είναι επεκτάσεις του Joomla! και κάποια είναι εφαρμογές βασισμένες στο web.

3.5.1 Joomla! Extentions

Freestyle Support Portal

Το Freestyle Support Portal, είναι μια επέκταση για Joomla! sites το οποίο διανέμεται έναντι κάποιας αμοιβής. Είναι ένα ticket support module, στο οποίο οι χρήστες μπορούν να δημιουργήσουν νέα tickets για να υποβάλουν το αίτημά τους και να λάβουν απάντηση από τον διαχειριστή.

Current Support Tickets

Open New Ticket	View Open (1)	View Closed (1)	View All (2)
Ticket Ref	Status	Last Update	Assigned
Demo			
JKCF-LMLG-KCAM	Open	Thursday, 11 March 2010 05:53	Unassigned

Εικόνα 14: Freestyle Support - Περιοχή Διαχείρισης Εισητηρίων

Ποιο συγκεκριμένα η διαδικασία που ακολουθείται για το συγκεκριμένο module είναι η εξής:

α) Οι πελάτες συμπληρώνουν το θέμα – περιεχόμενο, ορίζουν σπουδαιότητα, μπορούν αν θέλουν να επισυνάψουν αρχεία και να αποστείλουν τη συμπληρωμένη φόρμα τους.

β) Το αίτημα από τον πελάτη, εισέρχεται στη φόρμα του διαχειριστή πλέον ως ένα ticket. Στο ticket αυτό, ο διαχειριστής μπορεί να προσθέσει σχόλια (που να φαίνονται ή όχι στον πελάτη), να προωθήσει το ticket σε κάποιο άλλο υπάλληλο, να επισυνάψει κάποιο αρχείο κτλ. Για όλες τις ενέργειες που γίνονται, ο πελάτης ενημερώνεται με emails.

γ) Ο πελάτης επίσης μπορεί να ενημερωθεί για την πορεία του ticket του μέσα από την περιοχή διαχείρισης των αιτημάτων του (απ' όπου μπορεί να δει και παλαιότερα tickets του).

δ) Τέλος ο διαχειριστής του ticket, το χαρακτηρίζει ως open, follow – up ή closed.

Άλλα χαρακτηριστικά της εφαρμογής είναι τα εξής:

¹³ http://en.wikipedia.org/wiki/Issue_tracking_system

Post Reply

Message Details

Subject

Upload File (Max 8 MB)

Εικόνα 15: Freestyle Support - Φόρμα Απάντησης σε Ticket

Γενικά:

- Οι χρήστες μπορούν να κάνουν login στην περιοχή διαχείρισης του support ticket.
- Οι χρήστες μπορούν να δουν και παλαιότερα tickets στην περιοχή διαχείρισης.
- Τα support tickets μπορούν να διαχωριστούν ανά κατηγορία, τομέα ή ανά προϊόν.
- Οι χρήστες μπορούν να επισυνάψουν αρχεία στο support ticket.
- Μπορούν να εισαχθούν σχόλια από την πλευρά του διαχειριστή, τα οποία δεν εμφανίζονται στον πελάτη.
- Παρέχεται η δυνατότητα να προωθηθεί το μήνυμα σε άλλο υπάλληλο από το διαχειριστή, ενημερώνοντας παράλληλα και τον πελάτη.
- Μπορεί να υποβληθεί ticket χωρίς να είναι απαραίτητα registered κάποιος χρήστης.
- Παρέχεται η δυνατότητα να εισαχθούν απεριόριστα επισυναπτόμενα αρχεία.
- Προσθήκη / επεξεργασία απεριόριστων πεδίων και κατηγοριών tickets.
- Διάφορες ρυθμίσεις σε θέματα που αφορούν τη διαχείριση χρηστών, την κατοχή του ticket, την καταχώρηση του ticket από το διαχειριστή καθώς και για την περιοχή των support emails.
- Τα αρχεία για το παραπάνω extension βρίσκονται: <http://freestyle-joomla!.com/products/freestyle-support-portal>

Issue Manager (<http://www.fast-forward.es/joomla>)

Το Issue Manager, είναι μια εφαρμογή διαχείρισης περιστατικών, επεκτείνει το Joomla! διανέμεται δωρεάν και αποτελείται από ένα component και ένα module.

Επιτρέπει την κεντρική παρακολούθηση της επικοινωνίας μεταξύ του πελάτη (ή χρήστη) και του προσωπικού υποστήριξης (που έχουν πρόσβαση στη web εφαρμογή). Αντικαθιστά την επικοινωνία μέσω των emails (που μπορεί να προκαλέσουν πρόβλημα σε κάποιες περιπτώσεις), καθώς το Issue Manager διαχειρίζεται όλη τη ροή της επικοινωνίας σε ένα μόνο κανάλι, στο οποίο το προσωπικό που ασχολείται με την υποστήριξη μέσω διαδικτύου, καθιστάτε υπεύθυνο να παρακολουθεί κάθε αίτημα και ανάδραση από τους χρήστες ή τους πελάτες, μέσω των «trouble tickets».

Ticket Subject:

Message:

Εικόνα 16: Issue Manager - Φόρμα Εισαγωγής Νέου Ticket

Το Issue Manager χρησιμοποιεί διαφορετικά εργαλεία που βοηθάνε στη διαχείριση κάθε περιστατικού (ή ticket), όπως είναι για παράδειγμα το «νέο μήνυμα» και το «επίλυση εισιτηρίου», ο χρωματισμός για τον ευκολότερο εντοπισμό ενός κλεισμένου ζητήματος ή ζητημάτων που περιμένουν απάντηση. Υπάρχει επίσης η δυνατότητα να αναρτώνται κοινοποιήσεις, και ειδοποιήσεις των νέων μηνυμάτων που έχουν σταλεί μέσω email. Επιπλέον το Issue Manager, προσφέρει λειτουργικότητα για να ενσωματωθεί με το σύστημα παραγγελιών της λύσης που προσφέρει το Joomla! για ηλεκτρονικό κατάστημα, του VirtueMart, προσφέροντας έτσι ένα καλό τρόπο διαχείρισης περιστατικών σχετιζόμενων με τις παραγγελίες.

Οι χρήστες της συγκεκριμένης υπηρεσίας που προσφέρεται μέσω τις ιστοσελίδας κάποιας εταιρίας – οργανισμού, είναι σε θέση οποιαδήποτε στιγμή να δημιουργήσουν νέα tickets για να μπορούν να υποβάλουν κάποιο παράπονο, κάποιο αίτημα, κάποια πρόταση κτλ.

Κάθε ένα από αυτά τα εισιτήρια (tickets), θα πρέπει να παρακολουθείται και να δίνεται απάντηση από τους διαχειριστές της συγκεκριμένης λειτουργικότητας.

Όλα τα εισιτήρια, συμπεριφέρονται σαν θέματα ενός φόρουμ μηνυμάτων, με τη μόνη διαφορά ότι είναι ιδιωτικά. Είναι δηλαδή ορατά μόνο στο χρήστη / πελάτη που το δημιούργησε και στο εξουσιοδοτημένο προσωπικό του διαδικτύου.

Ποιο συγκεκριμένα τα χαρακτηριστικά που προσφέρει η συγκεκριμένη εφαρμογή είναι:

- Εικονίδια ενημέρωσης «μη αναγνωσμένο νέο μήνυμα» και «επιλυμένο εισιτήριο»
- Χρωματικούς κώδικες για μηνύματα τύπου «περιμένει απάντηση» και «εισιτήριο κλειστό»

Issue Manager

<input type="checkbox"/>	Alerts	Ticket Number	Order Number	Subject	Creation Date	Date Last Modified	View
<input type="checkbox"/>		QIU4EYM272	00000058	Problemas. Esto es una mierda	17-12-2010, 17:35:54	17-12-2010, 22:22:03	
<input type="checkbox"/>		NZGZ8M1OKQ		chm	17-12-2010, 16:57:35	17-12-2010, 17:35:03	
<input type="checkbox"/>		CX3Z6BGNP7		wertztrhrthbrtsb	16-12-2010, 18:35:47	16-12-2010, 22:40:25	
<input type="checkbox"/>		KG0ZYXNCLE		ciao	16-12-2010, 15:52:35	17-12-2010, 05:29:44	
<input type="checkbox"/>		LTN9003SAQ	00000036	sdfg	16-12-2010, 00:04:48	16-12-2010, 15:24:51	
<input type="checkbox"/>		6TT1183BHP		testen wat wanneer getoond wordt	15-12-2010, 17:30:37	15-12-2010, 18:02:49	
<input type="checkbox"/>		LR9UB12GMX		bnnnn	15-12-2010, 15:34:17	15-12-2010, 15:36:22	
<input type="checkbox"/>		A7FX2QLEUN		test 01	15-12-2010, 13:18:45	15-12-2010, 13:22:03	
<input type="checkbox"/>		JTMQGR8BYA	00000057	Houston, we have a problem.	14-12-2010, 20:52:15	15-12-2010, 19:20:10	
<input type="checkbox"/>		9DV54J7E18		this is a test	14-12-2010, 20:49:58	16-12-2010, 10:52:45	

Display #

<< Start < Prev 1 2 3 4 5 6 7 8 9 10 Next > End >>

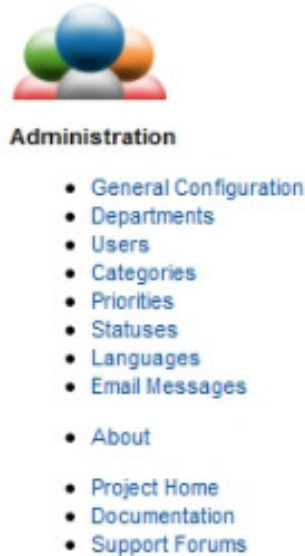
Page 1 of 99

Εικόνα 17: Issue Manager - Λίστα Tickets

- Πλήρη λειτουργικότητα με την επέκταση ηλεκτρονικού καταστήματος VirtueMart σαν επιλογή, που μπορεί να ενεργοποιηθεί από το back end του Joomla!. Αυτό θα επιτρέπει στους πελάτες του ηλεκτρονικού καταστήματος να αναφέρουν οποιοδήποτε περιστατικό σχετικά με οποιαδήποτε παραγγελία είχαν κάνει. Ένα νέο εισιτήριο, θα μπορεί να ανατεθεί σε μια συγκεκριμένη παραγγελία από μια λίστα παραγγελιών που αφορούν στο συγκεκριμένο πελάτη. Φυσικά ο χρήστης θα μπορεί να δημιουργήσει εισιτήριο και εκτός του συστήματος παραγγελιών.
- Με τη βοήθεια ενός AJAX αναδιδόμενου παράθυρου, εμφανίζονται στον πελάτη τα αντικείμενα που έχουν μπει στο καλάθι αγορών, περνώντας απλά το ποντίκι πάνω από τον αριθμό της παραγγελίας.
- Το Issue Manager επιτρέπει την αποστολή ειδοποιήσεων μέσω email (εάν η υπηρεσία αυτή είναι ενεργοποιημένη).
- Επιδέχεται αλλαγή στο σύμβολο της νομισματικής μονάδας που χρησιμοποιείται.
- Υποστηρίζει μετάφραση σε άλλες γλώσσες.

Huru Helpdesk (<http://www.huruhelpdesk.net>)

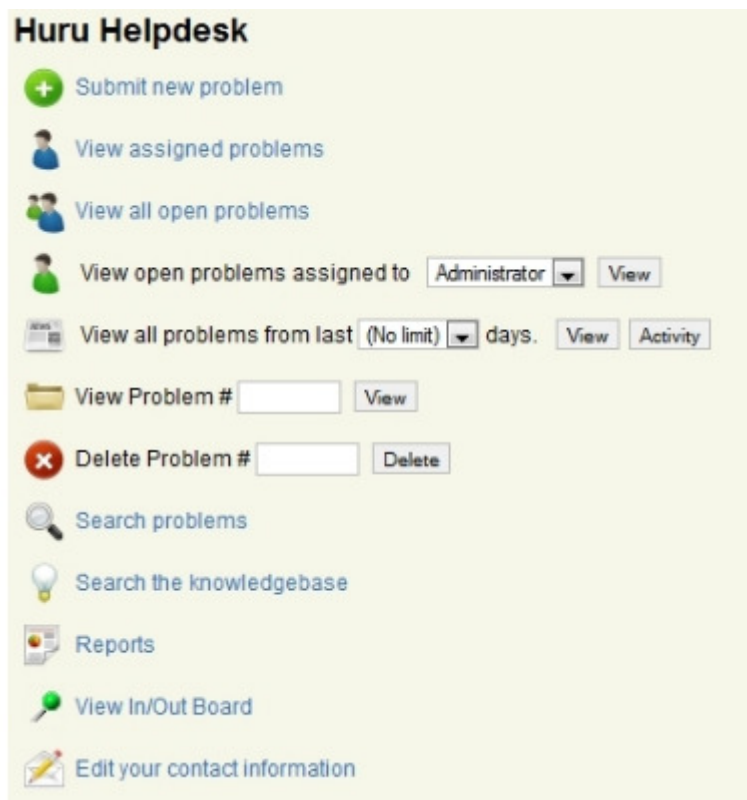
Το Huru Helpdesk είναι μια δωρεάν (δέχεται μόνο δωρεές για το συγκεκριμένο έργο) εφαρμογή συμβατή με τις εκδόσεις 1.5.x του Joomla! Αναφέρεται σε υπηρεσίες helpdesk και έχει ενσωματωμένη λειτουργικότητα της βάσης γνώσεων. Το Huru Helpdesk έχει πολλά χαρακτηριστικά σχεδιασμένα για να βοηθήσουν τους υπαλλήλους και τους διαχειριστές του ιστοτόπου με την παρακολούθηση και το χειρισμό των θεμάτων υποστήριξης.



Εικόνα 18: Διαχείριση Huru Helpdesk για το Joomla! back end

Το Huru Helpdesk έχει παρέχει αρκετά χαρακτηριστικά. Μερικά από αυτά είναι:

- Ολοκληρωμένη βάση γνώσεων (Knowledgebase). Νέες «γνώσεις» μπορούν να αποθηκευτούν και να παρουσιάζονται από τη knowledgebase απλά τοποθετώντας ένα σημαίακι.
- Οι χρήστες μπορούν να δουν την πρόοδο σχετικά με κάποιο ζήτημα και να υποβάλουν συμπληρωματικές πληροφορίες.
- Οι χρήστες μπορούν να φορτώσουν και να επισυνάψουν αρχεία στα ζητήματα που δημιουργούν
- Έχει ενσωματωμένη λειτουργία που εμφανίζει μια αναφορά σχετικά με το χρόνο που δαπανάται ανα κατηγορία και ανά υπηρεσία.
- Η σύνδεση και η έγκριση χρήστη για εργασίες στο σύστημα, είναι συνδεδεμένο με τους πίνακες χρηστών του Joomla!(άρα αρκεί κάποιος χρήστης να κάνει είσοδο μόνο στο site)



Εικόνα 19: Πίνακας Διαχείρισης Ζητημάτων απο το Προσωπικό Υποστήριξης

- Προσφέρεται ένα plug in που συγχρονίζει τους πίνακες χρηστών του Joomla! με αυτούς του Huru Helpdesk
- Μη εγγεγραμμένοι χρήστες, μπορούν να δημιουργήσουν νέο ζήτημα (προορατικό)
- Όλη η διαχείριση των ζητημάτων γίνεται από το front end, που σημαίνει ότι δεν είναι ανάγκη να δοθούν δικαιώματα στους υπαλλήλους των τμημάτων υποστήριξης για το back end.
- Συνδεδεμένη λειτουργία αποστολής email με το Joomla! για την επικοινωνία με τους υπαλλήλους τμημάτων υποστήριξης και τους πελάτες.
- Μπορεί να καταχωρεί κάποιο προεπιλεγμένο εκπρόσωπο υποστήριξης αναλόγως την κατηγορία του προβλήματος.

Το Huru Helpdesk έχει αποσπάσει αρκετά καλές κριτικές και από τα δωρεάν components είναι αυτό που έχει δεχτεί τις περισσότερες ψήφους.

Επίσης άλλα πακέτα έτοιμης λειτουργικότητας, παρόμοια με τα παραπάνω είναι:

- LiveHelpNow
- Billets
- JTag Helpdesk
- EMI Tracking System
- Webamoeba Ticket System

Αυτά και ακόμα περισσότερα, βρίσκονται στη διεύθυνση: extensions.joomla!.org

Επίσης εκτός από επεκτάσεις Joomla!, υπάρχουν και διάφορες άλλες διαδικτυακές εφαρμογές (Web Based Applications) που καλύπτουν την ανάγκη του Help Desk και του Project Management, χωρίς την ανάγκη ύπαρξης κάποιας ιστοσελίδας για να εγκατασταθούν.

3.5.2 Διαδικτυακές εφαρμογές (Web Based εφαρμογές)

Οι Web Based εφαρμογές, έχουν τη δυνατότητα να λειτουργούν αυτόνομα στο internet, δηλαδή δεν επεκτείνουν κάποιο υφιστάμενο Σύστημα Διαχείρισης Περιεχομένου. Στην ενότητα αυτή θα μελετηθεί στο SysAid Help Desk, που είναι αντιπροσωπευτικό παράδειγμα της κατηγορίας αυτής.

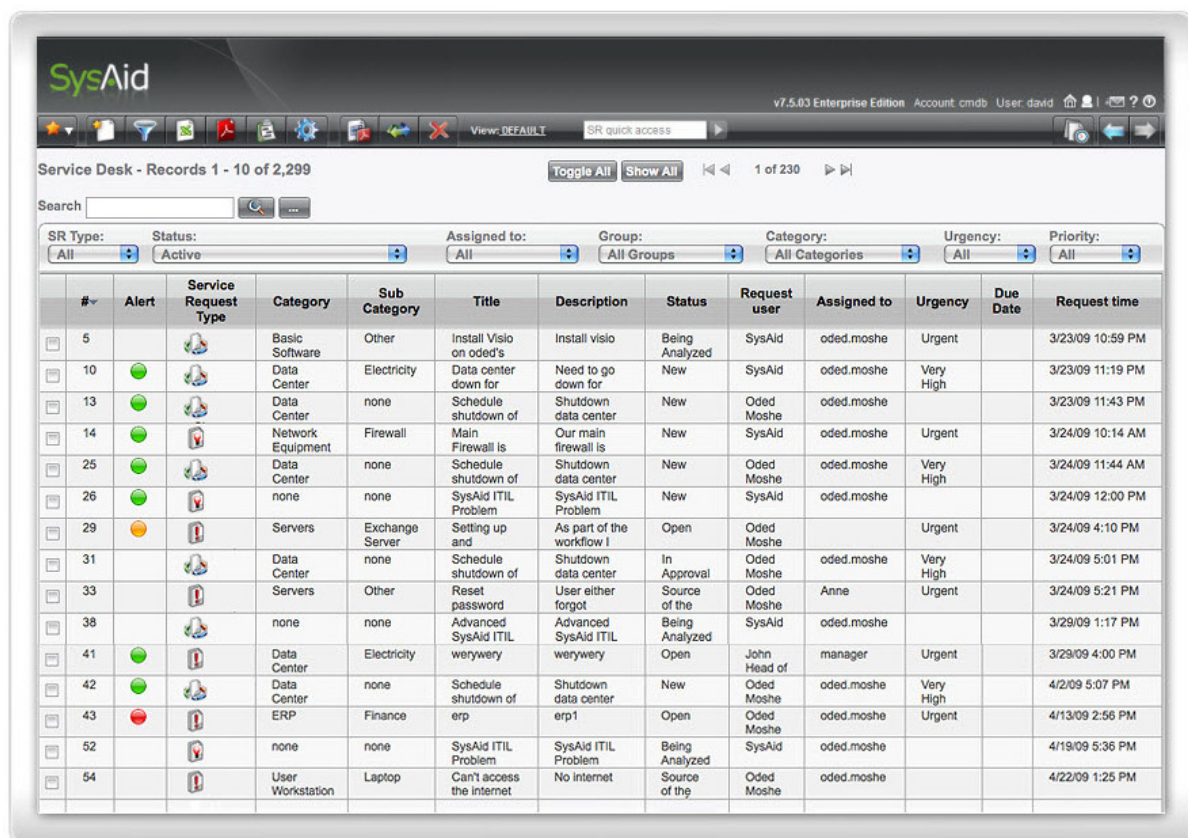
SysAid Help Desk (www.ilient.com)

Το SysAid είναι ένα ολοκληρωμένο web based σύστημα διαχείρισης ζητημάτων. Αυτοματοποιεί τις διαδικασίες για Help desk, διαμόρφωσης υλικού, παρακολούθησης αδειών χρήσης λογισμικού, καθήκοντα κ.α.

Ποιο συγκεκριμένα τα χαρακτηριστικά που προσφέρει είναι τα εξής:

Για το Help Desk

- Εύκολη υποβολή αιτήσεων παροχής υπηρεσιών μέσω του portal του τελικού χρήστη. Παρέχεται η δυνατότητα εγκατάστασης ενός κουμπιού άμεσης πρόσβασης στο portal τελικού πελάτη καθώς και η σύνδεση του portal αυτού την ιστοσελίδα της εταιρείας.
- Πλήρη υπηρεσία emails, παραμετροποιημένη κατάλληλα ώστε να αποθηκεύονται όλα τα μηνύματα μέσα στο πλαίσιο αιτημάτων υπηρεσιών
- Αυτόματη δρομολόγηση των αιτημάτων παροχής, σύμφωνα με προκαθορισμένες ρυθμίσεις.



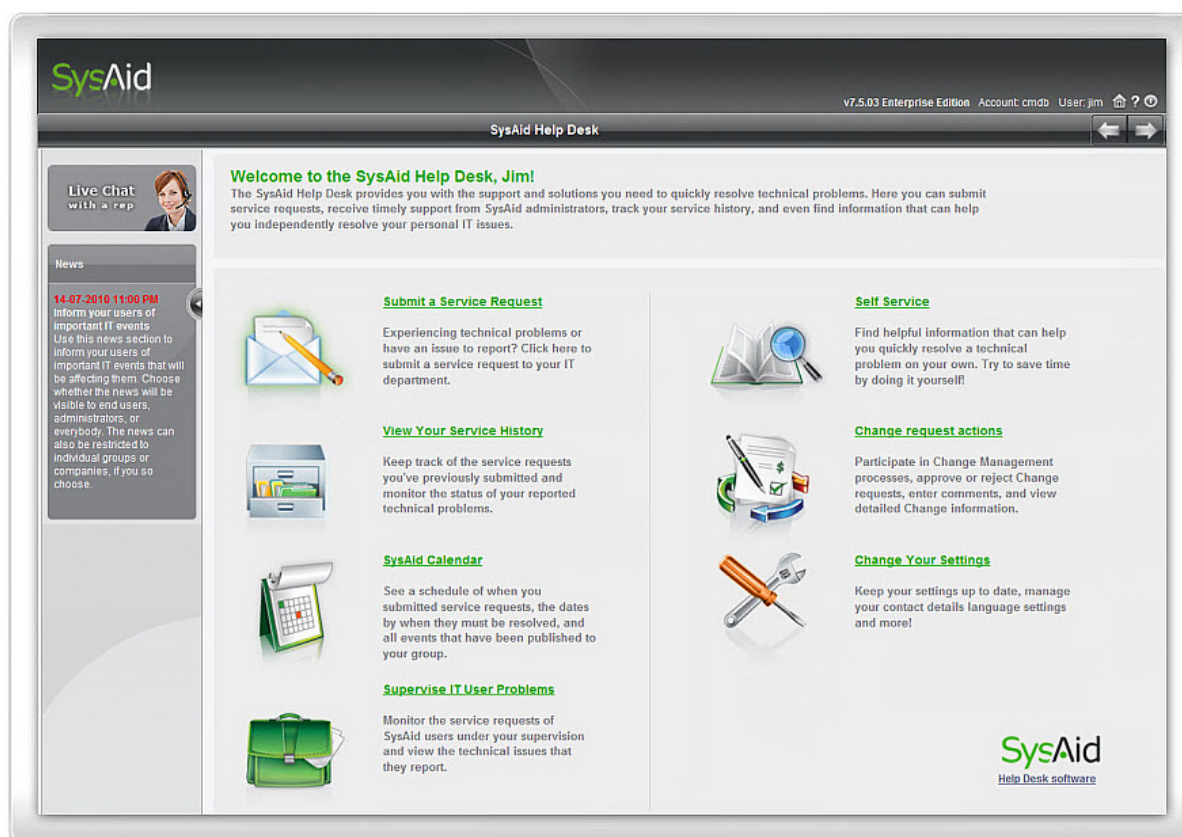
#	Alert	Service Request Type	Category	Sub Category	Title	Description	Status	Request user	Assigned to	Urgency	Due Date	Request time
5			Basic Software	Other	Install Visio on oded's	Install visio	Being Analyzed	SysAid	oded.moshe	Urgent		3/23/09 10:59 PM
10			Data Center	Electricity	Data center down for	Need to go down for	New	SysAid	oded.moshe	Very High		3/23/09 11:19 PM
13			Data Center	none	Schedule shutdown of	Shutdown data center	New	Oded Moshe	oded.moshe			3/23/09 11:43 PM
14			Network Equipment	Firewall	Main Firewall is	Our main firewall is	New	SysAid	oded.moshe	Urgent		3/24/09 10:14 AM
25			Data Center	none	Schedule shutdown of	Shutdown data center	New	Oded Moshe	oded.moshe	Very High		3/24/09 11:44 AM
26			none	none	SysAid ITIL Problem	SysAid ITIL Problem	New	SysAid	oded.moshe			3/24/09 12:00 PM
29			Servers	Exchange Server	Setting up and	As part of the workflow I	Open	Oded Moshe		Urgent		3/24/09 4:10 PM
31			Data Center	none	Schedule shutdown of	Shutdown data center	In Approval	Oded Moshe	oded.moshe	Very High		3/24/09 5:01 PM
33			Servers	Other	Reset password	User either forgot	Source of the	Oded Moshe	Anne	Urgent		3/24/09 5:21 PM
38			none	none	Advanced SysAid ITIL	Advanced SysAid ITIL	Being Analyzed	SysAid	oded.moshe			3/29/09 1:17 PM
41			Data Center	Electricity	werywery	werywery	Open	John Head of	manager	Urgent		3/29/09 4:00 PM
42			Data Center	none	Schedule shutdown of	Shutdown data center	New	Oded Moshe	oded.moshe	Very High		4/2/09 5:07 PM
43			ERP	Finance	erp	erp1	Open	Oded Moshe	oded.moshe	Urgent		4/13/09 2:56 PM
52			none	none	SysAid ITIL Problem	SysAid ITIL Problem	Being Analyzed	SysAid	oded.moshe			4/19/09 5:36 PM
54			User Workstation	Laptop	Can't access the internet	No internet	Source of the	Oded Moshe	oded.moshe			4/22/09 1:25 PM

Εικόνα 20: SysAid Διεπαφή Διαχείρισης Ζητημάτων

- Κλιμάκωση των κανόνων και των προτεραιοτήτων που καθορίζονται αναλόγως του χρήστη, της ημερομηνίας λήξης προθεσμίας, αν είναι επείγουσα ανάγκη και σε άλλους τομείς.
- Αυτόματες ενημερώσεις προς τους διαχειριστές και τους πελάτες, για να είναι διαρκώς ενημερωμένοι για την πορεία επίλυσης του ζητήματος.
- Προσαρμόσιμο περιβάλλον εργασίας για τη διαλογή των αιτήσεων παροχής υπηρεσιών.
- Προσαρμοζόμενο σύστημα έγκαιρης προειδοποίησης για να επισημαίνονται τα επείγοντα ζητήματα, τα άλτα, κτλ.
- Knowledgebase και γρήγορη εμφάνιση σε λίστα στοιχείων για την εύκολη επίλυση των τεχνικών θεμάτων.
- Έλεγχος δικαιωμάτων για την καλύτερη διαχείριση των δραστηριοτήτων των διαχειριστών.

- Προσαρμόσιμο για την έκδοση μια έρευνας για τους τελικούς χρήστες κατά το κλείσιμο ενός αιτήματος υπηρεσίας για την αξιολόγηση της ικανοποίησης του τελικού χρήστη.

Portal Τελικού Χρήστη – Πελάτη



Εικόνα 21: SysAid Portal Τελικού Χρήστη – Πελάτη

- Διαθέσιμο σε 42 διαφορετικές γλώσσες με δυνατότητα μετάφρασης σε οποιαδήποτε επιπλέον γλώσσα.
- Αυτόματη επισύναψη εικόνας οθόνης για να διευκρινιστεί το περιεχόμενο των αιτημάτων.
- Πρόσβαση του τελικού χρήστη στο ιστορικό του, για να παρακολουθεί την κατάσταση των υποβληθέντων αιτημάτων παροχής υπηρεσιών.
- Διάθεση τμημάτων Knowledgebase και FAQs, ώστε οι τελικοί χρήστες – πελάτες να μπορούν από μόνοι τους να επιλύουν το ζήτημά τους.
- Ημερολόγιο, ώστε να βλέπουν οι τελικοί χρήστες – πελάτες τις ημερομηνίες υποβολής αιτήματος, καθώς και την ημερομηνία λήξης του αιτήματός τους. Το ημερολόγιο βοηθάει επίσης στην ενημέρωση γεγονότων που γίνονται και αφορούν την ομάδα που έχει τοποθετηθεί ο χρήστης.
- Αυτόματη προβολή προτάσεων (από Knowledgebase) και φόρμα αντιμετώπισης προβλήματος, καθώς ο χρήστης γράφει τον τίτλο των εργασιών που σκοπεύει να ζητήσει.
- Εύκολη ενσωμάτωση του Portal του τελικού χρήστη σε υπάρχον site.
- Γρήγορη και εύκολη παραμετροποίηση για τις φόρμες που υποβάλλονται αιτήματα.
- Δυνατότητα εγκατάστασης κουμπιού άμεσης σύνδεσης με το portal του τελικού χρήστη (hot key).

Επίσης το SysAid παρέχει αρκετές ακόμα λειτουργίες με την αγορά του. Αυτές είναι knowledgebase, διαχείριση περιουσιακών στοιχείων της εταιρείας (λογισμικό, hardware), απομακρυσμένη διασύνδεση (remote control), αναφορές και στατιστικά, chat, ημερολόγιο.

Τέλος, με κάποια επιπλέον χρέωση, παρέχονται υπομονάδες (modules) όπως προηγμένη παρακολούθηση (περιέχει πειράματα σε πραγματικό χρόνο χρησιμοποίησης σκληρού δίσκου,

αυτόματες ενημερώσεις μόλις εισαχθεί νέο αίτημα υπηρεσιών όπως αποστολή μηνύματος στο κινητό, αποστολή email κτλ.), διαμοιρασμός εργασιών σε projects και tasks και έλεγχο των πινάκων οργάνων.

Επίσης μερικές άλλες από τις πιο γνωστές και χρησιμοποιούμενες παρόμοιες εφαρμογές είναι:

- Zendesk (www.zendesk.com)
- @ttask (www.webhelpdesk.com)
- Web Help Desk (www.webhelpdesk.com)

3.6 Περίληψη Κεφαλαίου

Στο κεφάλαιο αυτό παρουσιάστηκαν η ροή εργασιών, η αρχιτεκτονική καθώς και συγκεκριμένες υλοποιήσεις Issue Tracking Systems σχετικές με το αντικείμενο της παρούσας εργασίας. Αξίζει να σημειωθεί, πως για την επιλογή του τύπου συστήματος, παίζουν ρόλο πολλοί παράγοντες: όπως το οικονομικό μέγεθος που διατίθεται, η κάλυψη εξειδικευμένων απαιτήσεων, η δυνατότητα παραμετροποίησης και επεκτάσεων καθώς και η υφιστάμενη βάση εγκαταστάσεων (ως ένδειξη καλής υποστήριξης και αναβαθμίσεων).

Στην περίπτωση που επιλεγεί κάποιο component του Joomla! κρίνεται απαραίτητη η συνδρομή από εξειδικευμένο άτομο για την παραμετροποίηση του, καθώς τα περισσότερα από τα διαθέσιμα είναι σχεδιασμένα βάση αναγκών συγκεκριμένου τύπου εταιρειών, πράγμα που σημαίνει ότι μπορεί να μην αρμόζει στη λειτουργία άλλων.

Σε κάθε περίπτωση, τα Issue Tracking Systems είναι πολύ χρήσιμα για την άμεση, ορθή και ομαλή αμφίδρομη κυκλοφορία πληροφοριών από πελάτες προς την επιχείρηση. Με αυτόν τον τρόπο επιτυγχάνεται η βελτιστοποίηση της αποτελεσματικότητας και της αποδοτικότητας στη λειτουργία της επιχείρησης και επιτυγχάνεται ο τελικός στόχος, ο οποίος δεν είναι άλλος από την ικανοποίηση του πελάτη.

ΚΕΦΑΛΑΙΟ 4

CIMaS: Καταγραφή και Ανάλυση Απαιτήσεων

4.1 Εισαγωγή

Στο κεφάλαιο αυτό, θα καταγραφούν και θα αναλυθούν οι απαιτήσεις της εφαρμογής CIMaS (Content Issue Management System). Το όνομα αυτό δόθηκε για δύο λόγους. Ο πρώτος είναι γιατί περιγράφει τη λειτουργία της εφαρμογής(Content Issue Management System). Ο δεύτερος είναι το άκουσμα της λέξης. Πιο συγκεκριμένα, το άκουσμα της λέξης CIMaS, παραπέμπει στην ελληνική λέξη «σιμά» που σημαίνει κοντά – δίπλα. Η ονομασία αυτή, φανερώνει το χαρακτήρα της υλοποίησης αυτής. Να είναι δηλαδή «δίπλα» σε οποιοδήποτε ζήτημα από τη στιγμή που δημιουργείται, μέχρι την πλήρη και ορθή ολοκλήρωσή του.

Η διαδικασία της ανάλυσης και της καταγραφής των απαιτήσεων, είναι ιδιαίτερος σημαντική, καθώς θα προσδιοριστούν ακριβώς οι ανάγκες που πρέπει να καλυφθούν από το CIMaS.

4.2 Ορολογία

Πριν την καταγραφή των εμπλεκόμενων χρηστών και πριν την ανάλυση των απαιτήσεων από τη σκοπιά του καθενός από αυτούς, πρέπει να οριστούν οι οντότητες που θα χρησιμοποιηθούν, για την αποφυγή παρεξηγήσεων και παρερμηνειών.

Δεδομένα (data): Στην επιστήμη υπολογιστών, τα δεδομένα είναι οτιδήποτε σε μορφή κατάλληλη για χρήση από τον υπολογιστή, αλλά όχι κώδικας¹⁴.

Πληροφορία (Information): Ο υπολογιστής επεξεργάζεται δεδομένα και παράγει επεξεργασμένα δεδομένα. Ο άνθρωπος προσδίδει νόημα στα επεξεργασμένα δεδομένα μετατρέποντας τα σε πληροφορία¹⁵.

Κλήση: Θεωρείται τρόπος εισροής δεδομένων τα οποία ορίζονται από την πηγή (ποιος, που, πότε), το θέμα – ζητούμενο, την κατηγορία και τον αποδέκτη του περιεχομένου.

Issue: Αποτελείται από δεδομένα και πληροφορίες, που στόχο έχουν να προκαλέσουν ενέργεια. Το Issue ορίζεται από τα περιεχόμενα της κλήσης, τις ενέργειες – διαδικασίες των αποδεκτών του, καθώς και το τελικό αποτέλεσμα μετά την επεξεργασία των δεδομένων και των πληροφοριών.

Task: Είναι οι ενέργειες – διαδικασίες που γίνονται για να μεταβεί το issue στην τελική του κατάσταση. Ένα Issue μπορεί να αποτελείται από ένα ή πολλά Tasks.

Εντολέας: Είναι αυτός που δίνει σαφής πληροφορίες και δεδομένα για την μετάβαση σε επιθυμητό αποτέλεσμα.

Εντολοδόχος: Είναι ο παραλήπτης του περιεχομένου της εντολής (από τον εντολέα), και από τον οποίο ζητείται να γίνουν δράσεις, ώστε να φτάσει ένα issue στην ολοκλήρωσή του.

¹⁴ en.wikipedia.org/wiki/Data_(computing)

¹⁵ http://el.wikipedia.org/wiki/Πληροφορία

Πελάτης (Customer): Είναι μια οντότητα, που χαρακτηρίζεται από δεδομένα, και μέσω της κλήσης δημιουργεί Issues. Ο πελάτης είναι πάντα εντολέας.

Διευθυντής (Manager): Είναι μια οντότητα, που χαρακτηρίζεται από δεδομένα. Είναι ο εντολοδόχος του πελάτη και εντολέας των τεχνικών. Καλείται να συντονίσει (μέσω tasks) τις ενέργειες που θα γίνουν για την καλύτερη και γρηγορότερη ανταπόκριση – ολοκλήρωση ενός Issue.

Υπάλληλος (Employee): Είναι οντότητα, που επίσης χαρακτηρίζεται από δεδομένα. Είναι εντολοδόχος του Manager και του πελάτη και καλείται να ενεργήσει κατάλληλα, για να φτάσει το Issue στην ορθή, πλήρη και έγκαιρη ολοκλήρωσή του.

4.3 Προσδιορισμός Απαιτήσεων

Οι απαιτήσεις που πρέπει να υλοποιηθούν στο σύστημα αυτό, σκοπό έχουν την μεγαλύτερη ικανοποίηση του πελάτη, τη μείωση φόρτου τηλεφωνικών ενημερώσεων, καθώς και την τυποποίηση στη διαχείριση της ροής εργασιών του συστήματος (για αποφυγή παράληψης ενημέρωσης ζητημάτων). Άρα οι βασικοί τομείς ενδιαφέροντος είναι η διαχείριση σχέσεων με πελάτες (Customer Relationship Management), η διαχείριση ροής επίλυσης ζητημάτων (Workflow Management), η σύνδεση online του συστήματος με βάση δεδομένων, η απόδοση δικαιωμάτων ανάλογα το χρήστη και όλα αυτά, να προβάλλονται μέσω μιας καλαίσθητης διεπαφής. Ποιο συγκεκριμένα οι ανάγκες που πρέπει να ικανοποιηθούν στην υλοποίηση του συστήματος είναι οι εξής:

1. Να μπορεί κάθε χρήστης να δημιουργεί νέα ζητήματα και να τα κατηγοριοποιεί αναλόγως του περιεχομένου τους (π.χ. ζήτημα πληρωμών, ζήτημα πωλήσεων, ζήτημα υπηρεσιών κτλ).
2. Παρακολούθηση εξελικτικής πορείας – επίλυσης των ζητημάτων.
3. Ύπαρξη προσωπικής περιοχής ανά τελικό χρήστη για την ενημέρωση νέων προϊόντων και υπηρεσιών που τον αφορούν.
4. Να παρέχεται η δυνατότητα στον πελάτη να στέλνει και να λαμβάνει αρχεία ενδιαφέροντος
5. Δυνατότητα προβολής στατιστικών στοιχείων.
6. Έρευνα ικανοποίησης του πελάτη.

4.4 Καταγραφή Εμπλεκόμενων χρηστών

Στο σύστημά μας, γίνεται διαχωρισμός των χρηστών στις εξής κατηγορίες:

- **Πελάτης (Customer):** Ο ρόλος του στο σύστημά μας είναι να δημιουργεί νέα Issues. Έχει σημασία να αντιληφθούμε τις ανάγκες του στην υλοποίηση της εφαρμογής μας, καθώς αυτή έχει ουσιαστικά σαν στόχο τη μεγιστοποίηση της ικανοποίησης του.
- **Διευθυντής (Manager):** Είναι ο κύριος διαχειριστής της ροής εργασιών. Έχει διαφορετικές ανάγκες από το σύστημά, από τις αυτές του πελάτη.
- **Υπάλληλος (Employee):** Είναι αποδέκτης των Issues και καλείται να κάνει τις ενέργειες ώστε να υπάρξει η ικανοποίηση του πελάτη. Η εφαρμογή πρέπει να ελεγχθεί και από τη δική του σκοπιά, καθώς τα κριτήρια ικανοποίησης των αναγκών του, είναι διαφορετικά από αυτά του πελάτη και του διευθυντή.

4.5 Καταγραφή απαιτήσεων

Στην ενότητα αυτή, θα καταγράψουμε όλες τις απαιτήσεις που υπάρχουν για το σύστημα, από τις διαφορετικές σκοπιές των άμεσα εμπλεκόμενων χρηστών.

4.5.1 Σκοπιά Πελάτη

- Δημιουργία λογαριασμού χρήσης
- Προβολή και επεξεργασία στοιχείων
- Δυνατότητα αποστολής και λήψης αρχείων ενδιαφέροντος
- Ευχρηστία συστήματος

- Απλότητα στην περιήγηση
- Λακωνικότητα στη συμπλήρωση δεδομένων – πληροφοριών
- Απόδοση σπουδαιότητας
- Άμεση ενημέρωση για μέγιστο χρόνο αποκατάστασης
- Ταχύτερη δυνατή ενημέρωση για έναρξη εργασιών
- Διαρκή ενημέρωση για εργασίες που είναι ανοιχτές
- Ενημέρωση για εργασίες που υπολείπονται της ολοκλήρωσης
- Ενημέρωση για το status του αιτήματος του
- Αποστολή ερωτημάτων
- Δυνατότητα εμφάνισης ιστορικού
- Εμφάνιση στατιστικών στοιχείων

4.5.2 Σκοπιά Διευθυντή

- Ενημέρωση πελατών για ύπαρξη νέων προϊόντων – υπηρεσιών
- Ενημέρωση συγκεκριμένου πελάτη για ζήτημα
- Παραλαβή αποτελεσμάτων από έρευνα για την ικανοποίηση πελατών
- Δυνατότητα διασύνδεσης συστήματος με άλλα συστήματα καταγραφής κινήσεων πελατών
- Έγκαιρη ενημέρωση για Issue
- Δυνατότητα εμφάνισης στοιχείων πελάτη (π.χ. ιστορικού κ.α)
- Δυνατότητα διάσπασης Issue σε Tasks
- Δυνατότητα ανάθεσης Task σε συγκεκριμένο υπάλληλο
- Δημιουργία σειράς προτεραιότητας εργασιών
- Δυνατότητα ενημέρωσης για την ικανοποίηση του πελάτη
- Διαχείριση χρηστών
- Εμφάνιση στατιστικών στοιχείων βάση κριτηρίων
- Δυνατότητα επεξεργασίας περιεχομένου Issue και Task
- Να μπορεί να δημιουργήσει νέο μήνυμα για κάποιο πελάτη
- Να ενημερώνει το σύστημα για το κόστος
- Ύπαρξη πεδίου για πληροφορίες / συμβουλές προς τους υπαλλήλους

4.5.3 Σκοπιά Υπαλλήλου

- Έγκαιρη ενημέρωση για Issue
- Δυνατότητα εμφάνισης στοιχείων πελάτη (π.χ. ιστορικού κ.α)
- Να εμφανίζονται με προκαθορισμένο από το σύστημα τρόπο οι πληροφορίες που έχει δώσει ο πελάτης για το Issue
- Να έχει τη δυνατότητα να ενημερώνει για εργασίες που έχει κάνει για τα Issues και το Tasks
- Να έχει τη δυνατότητα να ενημερώνει το status του Issue
- Να μπορεί να δημιουργήσει νέο μήνυμα για κάποιο πελάτη
- Να ενημερώνει το σύστημα για το κόστος

4.6 Ανάλυση Απαιτήσεων

Στην ενότητα αυτή, γίνεται η ανάλυση των απαιτήσεων που εντοπίστηκαν και καταγράφηκαν παραπάνω.

4.6.1 Σκοπιά Πελάτη

Δημιουργία λογαριασμού χρήσης: Ο πελάτης, την πρώτη φορά που θα χρησιμοποιήσει το CIMaS, θα πρέπει να μπορεί να δημιουργήσει λογαριασμό με δήλωση των στοιχείων του.

Προβολή και επεξεργασία στοιχείων: Να παρέχεται η δυνατότητα στον πελάτη να μπορεί να δει τα στοιχεία που έχει δηλώσει, και να κάνει κάποια αλλαγή σε αυτά (π. χ. αλλαγή διεύθυνσης).

Δυνατότητα αποστολής και λήψης αρχείων ενδιαφέροντος: Το σύστημα, θα πρέπει να παρέχει τη δυνατότητα στο χρήστη να μεταφορτώσει και να κατεβάσει αρχεία που τον ενδιαφέρουν.

Εύχρηστιά προγράμματος: Το CIMaS, θα πρέπει να είναι εύχρηστο και λειτουργικό, ώστε να δημιουργηθεί στον πελάτη ένα κίνητρο να χρησιμοποιήσει αυτό, αντί για να πάρει τηλέφωνο.

Απλότητα στην περιήγηση: Θα πρέπει να παρέχεται στο χρήστη απλότητα στην περιήγηση του CIMaS και να μην δυσκολευτεί να βρει πως θα μεταβεί στο επιθυμητό σημείο.

Λακωνικότητα στη συμπλήρωση δεδομένων – πληροφοριών: Ο πελάτης θα πρέπει μέσα από φόρμες, με τη λιγότερη δυνατή συμπλήρωση δεδομένων – πληροφοριών, να προσφέρει μέγιστη ενημέρωση για την ανάγκη του.

Απόδοση σπουδαιότητας: Να μπορεί ο πελάτης να καθορίσει πόσο σημαντικό είναι για αυτόν η υλοποίηση του Issue (π. χ. πολύ σημαντικό, μέτριο, λίγο σημαντικό κτλ.)

Άμεση ενημέρωση για μέγιστος χρόνος αποκατάστασης: Βάση προκαθορισμένων κειμένων (ίσως αλγορίθμων) να γίνεται άμεσα η ενημέρωση για το μέσο χρονικό διάστημα έως την υλοποίηση του Issue.

Ταχύτερη δυνατή ενημέρωση για έναρξη εργασιών: Να υπάρχει η δυνατότητα να ενημερώνεται για το πότε ξεκίνησαν οι ενέργειες υλοποίησης του αιτήματός του.

Διαρκής ενημέρωση για εργασίες που έχουν γίνει: Μετά την έναρξη των εργασιών, να υπάρχει η δυνατότητα ενημέρωσής του για τις διαδικασίες που έχουν γίνει έως εκείνη τη δεδομένη στιγμή.

Ενημέρωση για το status του αιτήματος του: Να ενημερώνεται για το status (αν έχει ανατεθεί κ.α.)

Αποστολή ερωτημάτων: Θα πρέπει να παρέχεται η δυνατότητα πέρα από τη δημιουργία και προβολή Issue, ο πελάτης να μπορεί να στείλει κάποιο μήνυμα ελευθέρου κειμένου, για κάποια απορία του (π. χ. καρτέλα υπολοίπου), ή κάποιο σχόλιό του ή καινούρια ανάγκη που του δημιουργήθηκε.

Δυνατότητα εμφάνισης ιστορικού: Ο πελάτης είναι χρήσιμο να μπορεί να κάνει μέσα από το σύστημά μας μια αναδρομή σε περασμένες υπηρεσίες που μας είχε ζητήσει και να θυμηθεί το αποτέλεσμά τους.

Εμφάνιση στατιστικών στοιχείων: Ο πελάτης να μπορεί να ενημερώνεται για προκαθορισμένο τύπο στατιστικών στοιχείων.

4.6.2 Σκοπιά Διευθυντή

Ενημέρωση πελατών για ύπαρξη νέων προϊόντων – υπηρεσιών: Η απαίτηση αυτή είναι πολύ σημαντική, για την κάλυψη των προωθητικών ενεργειών μέσω του συστήματος. Βάση της απαίτησης αυτής, ο διευθυντής, θα μπορεί να στέλνει ενημερώσεις μαζικά σε πελάτες, ή να ενημερώνει συγκεκριμένο πελάτη για κάποια πιθανή προσφορά που τον αφορά.

Παραλαβή αποτελεσμάτων από έρευνα για την ικανοποίηση πελατών: Θα πρέπει να διεξάγονται έρευνες, για την παρακολούθηση της ικανοποίησης των πελατών. Επίσης πολύ σημαντική απαίτηση, καθώς μέσω των αποτελεσμάτων των ερευνών, ο διευθυντής θα πρέπει να ενεργήσει κατάλληλα (προς την κατεύθυνση που οδηγείται αναλύοντας τα αποτελέσματα αυτά) για την μεγιστοποίηση της ικανοποίησης των πελατών.

Δυνατότητα διασύνδεσης συστήματος με άλλα συστήματα καταγραφής κινήσεων πελατών: Είναι πολύ χρήσιμο, με κάποιο τρόπο να μπορεί να συνδέεται το σύστημα διαχείρισης ζητημάτων με κάποιο άλλο ίσως οικονομικού περιεχομένου σύστημα, ή οποιουδήποτε παρακολουθεί τις κινήσεις των πελατών (χρήσιμο για τιμολόγηση όταν κλείσει κάποιο issue, για τη συγκεντρωτική παρακολούθηση των πελατών κ.α.)

Έγκαιρη ενημέρωση για Issue: Ο Διευθυντής θα πρέπει να ενημερώνεται μέσω του συστήματος το συντομότερο δυνατό για την είσοδο ενός νέου Issue.

Δυνατότητα εμφάνισης στοιχείων πελάτη (π.χ. ιστορικού κ.α.): Ο Manager πρέπει να έχει πρόσβαση στα στοιχεία του πελάτη, καθώς αυτό μπορεί να παίζει σημαντικό ρόλο για τη σειρά που θα αποδώσει στο εισερχόμενο αίτημα στην ουρά εργασιών.

Δυνατότητα διάσπασης Issue σε Tasks: Δυνατότητα διάσπασης issue σε tasks, για την καλύτερη και γρηγορότερη επίτευξη του τελικού στόχου.

Δυνατότητα ανάθεσης Task σε συγκεκριμένο υπάλληλο: Θα πρέπει να παρέχεται στο Manager η δυνατότητα να αναθέσει κάποιο Task σε συγκεκριμένο υπάλληλο, καθώς αυτός είναι ο υπεύθυνος για τη ροή εργασιών.

Δημιουργία σειράς προτεραιότητας εργασιών: Θα πρέπει να μπορεί να θέσει σειρά υλοποίησης εργασιών βάση ίσως της σημαντικότητας κάποιου πελάτη ή της καλύτερης διαχείρισης χρόνου για το προσωπικό.

Δυνατότητα ενημέρωσης για την ικανοποίηση του πελάτη: Ο Manager θα πρέπει να ενημερώνεται για την ικανοποίηση του πελάτη από τις προσφερόμενες υπηρεσίες, ώστε αν ο πελάτης δεν μείνει ευχαριστημένος, να προβεί στις απαραίτητες ενέργειες για την βελτιστοποίηση των προσφερόμενων αγαθών (τόσο υλικών όσο και υπηρεσιών).

Διαχείριση χρηστών: Ο Manager όπως έχουμε αναφέρει αρκετές φορές παραπάνω είναι ο κύριος υπεύθυνος για τη ροή των εργασιών. Άρα θα πρέπει να μπορεί να διαχειρίζεται τους χρήστες (π. χ. δημιουργία νέου υπαλλήλου, διαγραφή κτλ.) και να τους δίνει τα απαραίτητα δικαιώματα (π. χ. ο υπάλληλος δεν έχει τα ίδια δικαιώματα με τον πελάτη).

Εμφάνιση στατιστικών στοιχείων βάση κριτηρίων: Είναι πολύ σημαντικό για την εποπτεία και την ανάπτυξη της εταιρείας, ο Manager να μπορεί να παίρνει στατιστικά στοιχεία για την σύσταση στρατηγικού πλάνου.

Δυνατότητα επεξεργασίας περιεχομένου Issue και Task: Για την πιο ομαλή λειτουργία, ο Manager θα είναι ο μόνος από τους χρήστες ο οποίος θα μπορεί να παρέμβει σε κάποιο Issue τροποποιώντας το περιεχόμενό του (π.χ. μια λάθος αναφορά από τον τεχνικό κ.α.).

Να μπορεί να δημιουργήσει νέο μήνυμα για κάποιο πελάτη: Σε περίπτωση που κάποιος πελάτης έδωσε τηλεφωνικά το issue του, ο διευθυντής θα πρέπει να μπορεί να ενημερώσει το σύστημα για την πληροφορία αυτή.

Να ενημερώνει το σύστημα για το κόστος: Ο Manager αναλόγως τον πελάτη ενδέχεται να θέλει να ασκήσει μια διαφορετική τιμολογιακή πολιτική. Το σύστημα να μπορεί να του δίνει αυτή τη δυνατότητα.

Ύπαρξη πεδίου για πληροφορίες / συμβουλές προς τους υπαλλήλους: Ο Manager να έχει τη δυνατότητα να συμπληρώσει κάποιο πεδίο με προτάσεις για την υλοποίηση του Issue ή οτιδήποτε άλλο θέλει να το δει ο υπάλληλος βλέποντας το issue – task που του έχει ανατεθεί.

4.6.3 Σκοπιά Υπαλλήλου

Έγκαιρη ενημέρωση για Issue: Ο υπάλληλος θα πρέπει να ενημερώνεται μέσω του συστήματος το συντομότερο δυνατό για την ανάθεση ενός νέου Issue.

Δυνατότητα εμφάνισης στοιχείων πελάτη (π.χ. ιστορικού κ.α.): Ο υπάλληλος θα πρέπει να έχει πρόσβαση στα συγκεκριμένα στοιχεία του πελάτη, καθώς μπορεί να χρειαστεί να του τηλεφωνήσει για κάποιο λόγο ή να δει στο ιστορικό εργασίες σχετικές που είχαν γίνει κατά το παρελθόν

Να εμφανίζονται με προκαθορισμένο από το σύστημα τρόπο οι πληροφορίες που έχει δώσει ο πελάτης για το Issue: Οι πληροφορίες του Issue θα πρέπει να εμφανίζονται με προκαθορισμένο τρόπο, ώστε να μην ξεφεύγουν λεπτομέρειες.

Να έχει τη δυνατότητα να ενημερώνει για εργασίες που έχει κάνει για το Issue και το Task: Το σύστημα θα πρέπει να έχει πεδία, τα οποία ο υπάλληλος θα μπορεί να τα ενημερώνει εύκολα για την εξέλιξη του Issue.

Να έχει τη δυνατότητα να ενημερώνει το status του Issue: Θα πρέπει να υπάρχει πεδίο τέτοιο ώστε ο υπάλληλος να ενημερώνει για το status του task, καθώς επίσης και για το ποσοστό ολοκλήρωσής αυτού.

Να μπορεί να δημιουργήσει νέο μήνυμα για κάποιο πελάτη: Σε περίπτωση που κάποιος πελάτης έδωσε τηλεφωνικά το issue του, ο υπάλληλος θα πρέπει να μπορεί να ενημερώσει το σύστημα για την πληροφορία αυτή.

Να ενημερώνει το σύστημα για το κόστος: Ο υπάλληλος να μπορεί βάση των εργασιών που έκανε να ενημερώσει για το κόστος χρέωσης της υπηρεσίας, καθώς και για το κόστος υλικών που ίσως χρησιμοποίησε.

4.7 Περίληψη Κεφαλαίου

Στο κεφάλαιο αυτό, εντοπίστηκαν οι ανάγκες που υπάρχουν και χρήζουν υποστήριξης από το CIMaS. Επίσης έγινε η ανάλυση και η καταγραφή αυτών αναλόγως του χρήστη, καθώς καθένας από αυτούς, έχει διαφορετικές ανάγκες και διαφορετικά δικαιώματα εκτέλεσης εργασιών στο σύστημα.

ΚΕΦΑΛΑΙΟ 5

Σχεδιασμός CIMaS

5.1 Εισαγωγή

Από το κεφάλαιο αυτό και έπειτα, θα αρχίσει να δημιουργείται συγκεκριμένη λειτουργικότητα για το CIMaS, η οποία θα πρέπει να εφαρμοστεί για την ορθή και πλήρη υλοποίηση της εφαρμογής.

Πιο συγκεκριμένα, παρακάτω θα καταγραφούν και θα αναλυθούν τα σενάρια χρήσης, τα διαγράμματα ροής εργασιών, το διάγραμμα βάσης δεδομένων και τα διαγράμματα ροής δεδομένων για τον σαφή καθορισμό των επιθυμητών αποτελεσμάτων της εφαρμογής.

5.2 Αρχιτεκτονική Συστήματος

Το σύστημα διαχείρισης ζητημάτων, θα υλοποιηθεί σε λογισμικό διαχείρισης περιεχομένου ανοιχτής αρχιτεκτονικής (ανεξάρτητο από τη χρήση συγκεκριμένων προτύπων που θεσπίζονται από προμηθευτές), με υποστήριξη από Σχεσιακό Σύστημα Βάσης Δεδομένων (RDBMS) και με δυνατότητα ύπαρξης απεριόριστου αριθμού χρηστών.

Μέσα από το σύστημα, θα μπορεί να γίνει η διαχείριση των χρηστών, η κατηγοριοποίησή τους σε ομάδες και η απόδοση δικαιωμάτων τους, ώστε να επιτευχθεί βέλτιστη διαχείριση των περιεχομένων του συστήματος.

Επιπλέον, το σύστημα βασίζεται σε ανοιχτά πρότυπα υλοποίησης εφαρμογών και είναι διαθέσιμο ανεξαρτήτως λειτουργικού συστήματος, ώστε να επιτευχθεί το δυνατόν περισσότερη χρησιμοποίηση από τους χρήστες καθώς και η ευελιξία για μελλοντική επεκτασιμότητα του.

5.3 Καταγραφή Σεναρίων χρήσης

Τα σενάρια χρήσης, βοηθάνε στην αποσαφήνιση των επιλογών που παρέχονται για τη χρήση του CIMaS σε κάθε εμπλεκόμενο χρήστη. Αυτά είναι χρήσιμα, καθώς στο παρόν στάδιο της υλοποίησης, αποφασίζονται τα δικαιώματα που θα δίνονται στον εκάστοτε χρήστη (αναλόγως του γκρουπ που θα τοποθετηθεί) και βάση αυτών, τι εργασίες μπορεί να εκτελέσει.

Στις αμέσως επόμενες υποενότητες, συγκεντρώνονται τα σενάρια αυτά, ενώ στην συνέχεια, γίνεται η αναλυτική καταγραφή τους.

5.3.1 Πελάτης

Στην υποενότητα αυτή, καθορίζονται τα σενάρια χρήσης όσων ανήκουν στο γρουπ των Πελατών.

- Προβολή – επεξεργασία στοιχείων λογαριασμού
- Δημιουργία νέου Issue
- Ενημέρωση για πορεία εργασιών σε Issue
- Προσθήκη σχόλιου σε Issue
- Προβολή σχολίων σε Issue
- Προσθήκη σχολίων σε Task

- Προβολή σχολίων σε Task
- Προβολή ιστορικού
- Δημιουργία – αποστολή ερώτησης
- Εμφάνιση στατιστικών στοιχείων

5.3.2 Υπάλληλος

Στην υποενότητα αυτή, καθορίζονται τα σενάρια χρήσης όσων ανήκουν στο γρουπ των Υπαλλήλων (employee).

- Δημιουργία Issue
- Προβολή Issues
- Ενημέρωση Issue
- Προβολή σχολίων σε Issue
- Προσθήκη σχολίου σε Issue
- Προβολή Tasks
- Ενημέρωση πεδίων Task
- Προσθήκη σχολίου σε Task
- Προβολή σχολίων σε Task
- Προβολή ιστορικού πελάτη
- Προβολή και απάντηση σε τυχόν ερωτήσεις

5.3.3 Διευθυντής

Τέλος, στην υποενότητα αυτή, προβάλλονται τα σενάρια χρήσης όσων ανήκουν στο γρουπ Διευθυντή (manager).

- Προβολή Issue
- Δημιουργία Issue
- Διόρθωση σε Issue
- Ενημέρωση για πορεία εργασιών σε Issue
- Προσθήκη σχολίου σε Issue
- Προβολή σχολίων σε Issue
- Διάσπαση Issue σε Task – Δημιουργία Task
- Ανάθεση Task σε υπάλληλο
- Προβολή Tasks
- Διόρθωση σε Task
- Ενημέρωση για πορεία εργασιών σε Task
- Εισαγωγή σχολίου σε Task
- Προβολή σχολίων σε Task
- Εμφάνιση στατιστικών στοιχείων
- Προβολή ιστορικού πελάτη

5.4 Ανάλυση Σεναρίων Χρήσης

Στην ενότητα αυτή, θα αναλυθούν τα σενάρια χρήσης που καταγράψαμε στο προηγούμενο στάδιο του σχεδιασμού (στην καταγραφή σεναρίων).

Η ανάλυση των σεναρίων, εξυπηρετεί αφενός στο να καθοριστούν τα «βήματα» που θα ακολουθούνται για οποιαδήποτε διεργασία, αφετέρου στο να δημιουργηθεί για το κάθε σενάριο ένα σκαρίφημα που θα εμφανίζει τη χρησιμοποιούμενη διεπαφή που πρέπει να αναπτυχθεί. Η ανάλυση των «βημάτων» που αναφέραμε παραπάνω, θα μας βοηθήσει στο σαφή καθορισμό της επιχειρηματικής λογικής, ενώ τα σκαρίφηματα, θα μας προϊδεάσουν για την εικόνα που θα έχει το CIMaS μετά την ολοκλήρωση της υλοποίησής του.

5.4.1 Σενάρια Χρήσης για τον Πελάτη

Σε αυτή την υποενότητα, θα καταγραφούν αναλυτικά τα σενάρια χρήσης και τα βήματα ενεργειών, για την ομάδα χρηστών «Πελάτες», για το CIMaS.

5.4.1.1 Περίπτωση χρήσης: Δημιουργία Issue

Σενάριο: Δημιουργία Issue

Για να μπορέσει ο πελάτης να δημιουργήσει ένα καινούριο Issue, θα πρέπει να εκτελέσει τα παρακάτω βήματα:

Βήματα Σεναρίου

1. Εισαγωγή χρήστη στο σύστημα
2. Επιλογή δημιουργίας νέου Issue
3. Καταγραφή δεδομένων Issue
4. Υποβολή δεδομένων

Αναλυτική Περιγραφή Βημάτων Σεναρίου

Βήμα 1: Εισαγωγή χρήστη στο σύστημα

Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!

Βήμα 2: Επιλογή δημιουργίας νέου Issue

Μετά την εισαγωγή του χρήστη στο σύστημα, θα εμφανίζονται οι επιλογές (τα μενού) του CIMaS. Από αυτά θα επιλέγει Νέο Issue.

Βήμα 3: Καταγραφή δεδομένων Issue

Μετά την ολοκλήρωση των παραπάνω βημάτων, θα εμφανίζεται μια φόρμα καταγραφής για δημιουργία νέου Issue. Στο βήμα αυτό, θα καταχωρούνται τα δεδομένα.

Βήμα 4: Υποβολή δεδομένων

Μετά την ολοκλήρωση συμπλήρωσης, τα στοιχεία αυτά θα υποβάλλονται στη βάση δεδομένων, πατώντας το κουμπί submit.

Σκαρίφημα

Title	<input type="text"/>
Category	<input type="text"/>
Arrival Date	Τρέχουσα Ημ/νια
Priority	<input type="text"/>
Status	<input type="text"/>
User	<input type="text" value="Username Πελάτη"/>
Content	<input type="text"/>
Attachment	<input type="text"/> ...
<input type="button" value="Submit"/>	

Σκαρίφημα 1: Δημιουργία Issue

5.4.1.2 Περίπτωση Χρήσης: Ενημέρωση για πορεία εργασιών σε Issue

Σενάριο: Ενημέρωση για Πορεία Εργασιών σε Issue

Στο σενάριο αυτό, ο πελάτης θα μπορεί να δει τις εργασίες που έχουν γίνει σε κάποιο Issue. Κάθε issue έχει τουλάχιστον ένα task. Μπορεί επίσης να έχει και σχόλια. Από το σενάριο αυτό θα μπορεί να δει τα tasks και τα σχόλια του issue.

Βήματα Σεναρίου

1. Εισαγωγή χρήστη στο σύστημα
2. Επιλογή προβολής υπάρχοντων Issues
3. Επιλογή προβολής σχόλιων Issue
4. Επιλογή προβολής Tasks
5. Επιλογή προβολής Task
6. Επιλογή προβολής σχόλιων Task

Αναλυτική Περιγραφή Βημάτων Σεναρίου

Βήμα 1: Εισαγωγή χρήστη στο σύστημα

Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!

Βήμα 2: Επιλογή προβολής υπάρχοντων Issues

Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues» την προβολή υπάρχοντος issue. Στον πίνακα που θα προβληθεί, θα υπάρχει ένα πεδίο με το status του issue (δεν έχει ανατεθεί, έχει ανατεθεί, γίνονται εργασίες κτλ), ένας σύνδεσμος που θα οδηγεί στη λίστα των tasks και ένας σύνδεσμος που θα οδηγεί στη λίστα των σχόλιων που υπάρχουν για το συγκεκριμένο issue.

Βήμα 3: Επιλογή προβολής σχόλιων Issue

Στο βήμα αυτό, μετά την επιλογή του πελάτη, θα εμφανίζεται μια λίστα από τα σχόλια που υπάρχουν για το συγκεκριμένο issue.

Βήμα 4: Επιλογή προβολής Tasks

Στο βήμα αυτό, ο πελάτης θα βλέπει όλα τα tasks που έχουν δημιουργηθεί για το ζήτημά του. Στα tasks, θα υπάρχει επίσης ένας σύνδεσμος που θα οδηγεί στα σχόλια που υπάρχουν.

Βήμα 5: Επιλογή προβολής Task

Στη συνέχεια θα μπορεί να επιλέξει να εμφανιστεί μόνο του το task, για να προβληθούν με τον τρόπο αυτό όλες οι λεπτομέρειές του.

Βήμα 6: Επιλογή προβολής σχόλιων Task

Σε αυτό το βήμα, ο πελάτης θα μπορεί να αναγνώσει όλα τα σχόλια (λεπτομέρειες) που υπάρχουν για το συγκεκριμένο task.

Σκαριφήματα

#	Title	Category	Arrival	Priority	Status	Username	ID	Action	Comments	Attachments
1	blabla	blabla	blabla	blabla	blabla	blabla	68	blabla	blabla	blabla
2	blabla	blabla	blabla	blabla	blabla	blabla	83	blabla	blabla	blabla

Σκαρίφημα 2: Multi Issue View

5.4.1.3 Περίπτωση χρήσης: Προσθήκη – Προβολή Σχολίων Issue

<p>Σενάριο: Προβολή σχολίων Issue Για να μπορέσει ο πελάτης να δει τα σχόλια που υπάρχουν για το Issue, θα πρέπει να εκτελέσει τα παρακάτω βήματα:</p> <p>Βήματα Σεναρίου</p> <ol style="list-style-type: none">1. Εισαγωγή χρήστη στο σύστημα2. Επιλογή προβολής υπαρχόντων Issues3. Επιλογή προβολής σχολίων Issue <p>Αναλυτική Περιγραφή Βημάτων Σεναρίου</p> <p><u>Βήμα 1: Εισαγωγή χρήστη στο σύστημα</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!</p>	<p><u>Βήμα 2: Επιλογή προβολής υπαρχόντων Issues</u> Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues» την προβολή υπαρχόντος issue. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με το status του issue (δεν έχει ανατεθεί, έχει ανατεθεί, γίνονται εργασίες κτλ) και ένας σύνδεσμος που θα οδηγεί στη λίστα των σχολίων που υπάρχουν για το συγκεκριμένο issue.</p> <p><u>Βήμα 3: Επιλογή προβολής σχολίων Issue</u> Στο βήμα αυτό, μετά την επιλογή του πελάτη, θα προβάλλεται μια λίστα με τα σχόλια που αφορούν στο συγκεκριμένο Issue.</p>															
<p>Σκαριφήματα</p> <table border="1"><thead><tr><th>#</th><th>Comment</th><th>Creation Date</th><th>Issue</th><th>Username</th></tr></thead><tbody><tr><td>1</td><td><u>blabla</u></td><td>blabla</td><td>blabla</td><td>blabla</td></tr><tr><td>2</td><td><u>blabla</u></td><td>blabla</td><td>blabla</td><td>blabla</td></tr></tbody></table> <p>Σκαρίφημα 3: Προβολή Σχολίων σε Issue</p>		#	Comment	Creation Date	Issue	Username	1	<u>blabla</u>	blabla	blabla	blabla	2	<u>blabla</u>	blabla	blabla	blabla
#	Comment	Creation Date	Issue	Username												
1	<u>blabla</u>	blabla	blabla	blabla												
2	<u>blabla</u>	blabla	blabla	blabla												

Σενάριο: Δημιουργία σχόλιου σε Issue

Βάσει της περιγραφής του σεναρίου αυτού, ο πελάτης θα μπορεί να δημιουργήσει σχόλια μέσα σε κάποιο από τα ήδη υπάρχοντα issue.

Βήματα Σεναρίου

1. Εισαγωγή χρήστη στο σύστημα
2. Επιλογή προβολής υπάρχοντων Issues
3. Επιλογή προβολής σχόλιων Issue
4. Επιλογή δημιουργίας νέου σχόλιου
5. Υποβολή δεδομένων

Αναλυτική Περιγραφή Βημάτων Σεναρίου**Βήμα 1: Εισαγωγή χρήστη στο σύστημα**

Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!

Βήμα 2: Επιλογή προβολής υπάρχοντων Issues

Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues», την προβολή υπάρχοντος issue. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με το status του issue (δεν έχει ανατεθεί, έχει ανατεθεί, γίνονται εργασίες κτλ), καθώς επίσης και ένας σύνδεσμος που θα οδηγεί στη λίστα των σχόλιων που αφορούν στο συγκεκριμένο issue.

Βήμα 3: Επιλογή προβολής σχόλιων Issue

Στο βήμα αυτό, μετά την επιλογή του πελάτη, θα προβάλλεται μια λίστα με τα σχόλια που αφορούν στο συγκεκριμένο Issue.

Βήμα 4: Δημιουργία σχόλιου σε Issue

Στο βήμα αυτό, μετά την επιλογή του πελάτη, θα προβάλλεται μια φόρμα συμπλήρωσης για δημιουργία νέου σχόλιου.

Βήμα 5: Υποβολή δεδομένων

Μετά την ολοκλήρωση συμπλήρωσης, τα στοιχεία αυτά θα υποβάλλονται στη βάση δεδομένων, πατώντας το κουμπί submit.

Σκαριφήματα

The image shows a screenshot of a web form for creating a comment. It consists of a large rectangular text input field with the label 'Comment' to its left. Below this field, there is a smaller input field labeled 'Creation Date' containing the text 'blabla'. At the bottom center of the form is a button labeled 'Submit'.

Σκαρίφημα 4: Δημιουργία Comment σε Issue

5.4.1.4 Περίπτωση χρήσης: Προσθήκη – Προβολή Σχολίων σε Task

<p>Σενάριο: Προβολή σχολίων σε Task Για να μπορέσει ο πελάτης να δει τα σχόλια που υπάρχουν για το Task, θα πρέπει να εκτελέσει τα παρακάτω βήματα:</p> <p>Βήματα Σεναρίου</p> <ol style="list-style-type: none">1. Εισαγωγή χρήστη στο σύστημα2. Επιλογή προβολής υπαρχόντων Issues3. Επιλογή προβολής υπαρχόντων Tasks4. Επιλογή προβολής σχολίων του Task <p>Αναλυτική Περιγραφή Βημάτων Σεναρίου</p> <p><u>Βήμα 1: Εισαγωγή χρήστη στο σύστημα</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!</p>	<p><u>Βήμα 2: Επιλογή προβολής υπαρχόντων Issues</u> Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues», την προβολή υπάρχοντος issue. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με τις πληροφορίες που είναι σχετικές με αυτό. Ένας σύνδεσμος θα οδηγεί στη λίστα των σχολίων που υπάρχουν για το συγκεκριμένο issue, και ένας άλλος σύνδεσμος, θα οδηγεί στην προβολή των tasks που υπάρχουν για αυτό.</p> <p><u>Βήμα 3: Επιλογή προβολής υπαρχόντων Tasks</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να επιλέξει από τα Issues, το σύνδεσμο που θα οδηγεί στη λίστα των tasks που υπάρχουν για αυτό.</p> <p><u>Βήμα 4: Επιλογή προβολής σχολίων Task</u> Στο βήμα αυτό, μετά την επιλογή του πελάτη, θα προβάλλεται μια λίστα με τα σχόλια που υπάρχουν για το συγκεκριμένο task.</p>															
<p>Σκαριφήματα</p> <table border="1"><thead><tr><th>#</th><th>Comment</th><th>Creation Date</th><th>Task</th><th>Username</th></tr></thead><tbody><tr><td>1</td><td><u>blabla</u></td><td>blabla</td><td>blabla</td><td>blabla</td></tr><tr><td>2</td><td><u>blabla</u></td><td>blabla</td><td>blabla</td><td>blabla</td></tr></tbody></table> <p>Σκαρίφημα 5: Προβολή Σχολίων σε Task</p>		#	Comment	Creation Date	Task	Username	1	<u>blabla</u>	blabla	blabla	blabla	2	<u>blabla</u>	blabla	blabla	blabla
#	Comment	Creation Date	Task	Username												
1	<u>blabla</u>	blabla	blabla	blabla												
2	<u>blabla</u>	blabla	blabla	blabla												

Σενάριο: Δημιουργία σχόλιου σε Task

Στο σενάριο αυτού, ο πελάτης θα μπορεί να δημιουργήσει σχόλιο μέσα σε κάποιο από τα ήδη υπάρχοντα task.

Βήματα Σεναρίου

1. Εισαγωγή χρήστη στο σύστημα
2. Επιλογή προβολής υπαρχόντων Issues
3. Επιλογή προβολής υπαρχόντων Tasks
4. Επιλογή προβολής σχόλιων του Task
5. Επιλογή δημιουργίας νέου σχόλιου
6. Υποβολή δεδομένων

Αναλυτική Περιγραφή Βημάτων Σεναρίου

Βήμα 1: Εισαγωγή χρήστη στο σύστημα

Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!

Βήμα 2: Επιλογή προβολής υπαρχόντων

Issues

Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues» την προβολή ενός ήδη υπάρχοντος issue. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με το status του issue (δεν έχει ανατεθεί, έχει ανατεθεί, γίνονται εργασίες κτλ), καθώς επίσης και ένας σύνδεσμος που θα οδηγεί στη λίστα των σχόλιων που αφορούν στο συγκεκριμένο issue.

Βήμα 3: Επιλογή προβολής υπαρχόντων Tasks

Στο βήμα αυτό, ο χρήστης θα πρέπει να επιλέξει από τα Issues, το σύνδεσμο που θα οδηγεί στη λίστα των tasks που υπάρχουν για το συγκεκριμένο issue.

Βήμα 4: Επιλογή προβολής σχόλιων Task

Στο βήμα αυτό, μετά την επιλογή του πελάτη, θα προβάλλεται μια λίστα με τα σχόλια που αφορούν στο συγκεκριμένο task.

Βήμα 5: Επιλογή δημιουργίας νέου Task

Στο βήμα αυτό, μετά την επιλογή του πελάτη, θα προβάλλεται μια φόρμα συμπλήρωσης για δημιουργία νέου σχόλιου.

Βήμα 4: Υποβολή δεδομένων

Μετά την ολοκλήρωση συμπλήρωσης, τα στοιχεία αυτά θα υποβάλλονται στη βάση δεδομένων, πατώντας το κουμπί submit.

Σκαριφήματα

The screenshot shows a web form with a large text input field labeled 'Comment'. Below it, there is a label 'Creation Date' followed by the text 'blabla'. At the bottom of the form is a button labeled 'Submit'.

Σκαρίφημα 6: Δημιουργία Comment σε Task

5.4.1.5 Περίπτωση χρήσης: Προβολή Ιστορικού

<p>Σενάριο: Προβολή ιστορικού Για να μπορέσει ο πελάτης να δει τις εργασίες που έχουν γίνει στο παρελθόν κατόπιν αιτήματός του, θα πρέπει να εκτελέσει τα παρακάτω βήματα:</p> <p>Βήματα Σεναρίου</p> <ol style="list-style-type: none"> 1. Εισαγωγή χρήστη στο σύστημα 2. Επιλογή προβολής ιστορικού Issues 	<p>Αναλυτική Περιγραφή Βημάτων Σεναρίου</p> <p><u>Βήμα 1: Εισαγωγή χρήστη στο σύστημα</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!</p> <p><u>Βήμα 2: Επιλογή προβολής ιστορικού Issues</u> Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues» την προβολή ιστορικού. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με όλα τα issues που υπάρχουν στο σύστημά μας, τα οποία έχουν τιμή για το πεδίο status closed.</p>																																	
<p>Σκαριφήματα</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>#</th> <th>Title</th> <th>Category</th> <th>Arrival</th> <th>End date</th> <th>Comments</th> </tr> </thead> <tbody> <tr> <td rowspan="3">1</td> <td>Issue bla</td> <td>blabla</td> <td>blabla</td> <td></td> <td><u>blabla</u></td> </tr> <tr> <td>Task1 blabla</td> <td></td> <td>blabla</td> <td>blabla</td> <td><u>blabla</u></td> </tr> <tr> <td>Task2 blabla</td> <td></td> <td>blabla</td> <td>blabla</td> <td><u>blabla</u></td> </tr> <tr> <td rowspan="2">2</td> <td>Issue2 blabla</td> <td>blabla</td> <td>blabla</td> <td></td> <td><u>blabla</u></td> </tr> <tr> <td>Task blabla</td> <td></td> <td>blabla</td> <td>blabla</td> <td><u>blabla</u></td> </tr> </tbody> </table> <p style="text-align: center;">Σκαρίφημα 7: Προβολή Ιστορικού Πελάτη</p>		#	Title	Category	Arrival	End date	Comments	1	Issue bla	blabla	blabla		<u>blabla</u>	Task1 blabla		blabla	blabla	<u>blabla</u>	Task2 blabla		blabla	blabla	<u>blabla</u>	2	Issue2 blabla	blabla	blabla		<u>blabla</u>	Task blabla		blabla	blabla	<u>blabla</u>
#	Title	Category	Arrival	End date	Comments																													
1	Issue bla	blabla	blabla		<u>blabla</u>																													
	Task1 blabla		blabla	blabla	<u>blabla</u>																													
	Task2 blabla		blabla	blabla	<u>blabla</u>																													
2	Issue2 blabla	blabla	blabla		<u>blabla</u>																													
	Task blabla		blabla	blabla	<u>blabla</u>																													

5.4.1.6 Περίπτωση χρήσης: Δημιουργία - Προβολή – Επεξεργασία στοιχείων λογαριασμού

<p>Σενάριο: Δημιουργία Λογαριασμού Σε αυτό το σενάριο χρήσης, θα χρησιμοποιηθεί η έτοιμη λειτουργικότητα του Joomla! για τη δημιουργία λογαριασμού, και με μια φόρμα καταγραφής στοιχείων, θα κρατούνται οι επιπλέον πληροφορίες που χρειάζονται για κάθε χρήστη.</p> <p>Βήματα Σεναρίου</p> <ol style="list-style-type: none"> 1. Εισαγωγή χρήστη στο σύστημα 2. Επιλογή δημιουργίας λογαριασμού στο CIMaS 3. Υποβολή στοιχείων 	<p><u>Βήμα 1: Εισαγωγή χρήστη στο σύστημα</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!</p> <p><u>Βήμα 2: Δημιουργία λογαριασμού στο CIMaS</u> Στο βήμα αυτό, ο χρήστης θα καταγράψει κάποια από τα προσωπικά του στοιχεία, που είναι απαραίτητα για την ομαλή λειτουργία της ροής εργασιών του συστήματος.</p> <p><u>Βήμα 3: Υποβολή στοιχείων</u> Μετά την καταγραφή των στοιχείων του, θα πρέπει να κάνει υποβολή, για να αποθηκευτούν</p>
<p>Σκαριφήματα</p> <div style="border: 1px solid black; padding: 10px; width: fit-content; margin: auto;"> <p>Username <input type="text"/></p> <p>Phone <input type="text"/></p> <p>Adress <input type="text"/></p> <p style="text-align: center;"><input type="button" value="Submit"/></p> </div> <p style="text-align: center;">Σκαρίφημα 8: Στοιχεία Πελάτη</p>	

<p>Σενάριο: Προβολή Στοιχείων Λογαριασμού Σε αυτό το σενάριο, ο χρήστης θα μπορεί να κάνει επεξεργασία των στοιχείων που έχει δηλώσει στους πίνακες του CIMaS.</p> <p>Βήματα Σεναρίου</p> <ol style="list-style-type: none"> 1. Εισαγωγή χρήστη στο σύστημα 2. Επιλογή προβολής λογαριασμού στο CIMaS 	<p>Αναλυτική Περιγραφή Βημάτων Σεναρίου <u>Βήμα 1: Εισαγωγή χρήστη στο σύστημα</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!</p> <p><u>Βήμα 2: Προβολή λογαριασμού στο CIMaS</u> Στη συνέχεια, θα πρέπει να επιλέξει την προβολή των στοιχείων του λογαριασμού του και έπειτα να επεξεργαστεί όποια επιθυμεί.</p>
--	---

Σκαριφήματα

Username

Phone

Adress

Σκαρίφημα 9: Προβολή Στοιχείων Πελάτη

<p>Σενάριο: Επεξεργασία Στοιχείων Λογαριασμού Σε αυτό το σενάριο, ο χρήστης θα μπορεί να κάνει επεξεργασία των στοιχείων που έχει δηλώσει στους πίνακες του CIMaS.</p> <p>Βήματα Σεναρίου</p> <ol style="list-style-type: none"> 1. Εισαγωγή χρήστη στο σύστημα 2. Επιλογή επεξεργασίας λογαριασμού στο CIMaS 3. Υποβολή στοιχείων 	<p>Αναλυτική Περιγραφή Βημάτων Σεναρίου <u>Βήμα 1: Εισαγωγή χρήστη στο σύστημα</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!</p> <p><u>Βήμα 2: Προβολή λογαριασμού στο CIMaS</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να επιλέξει την εμφάνιση των στοιχείων του λογαριασμού του και έπειτα να επεξεργαστεί τα στοιχεία που επιθυμεί.</p> <p><u>Βήμα 3: Υποβολή στοιχείων</u> Στο βήμα αυτό, ο χρήστης μετά την επεξεργασία των στοιχείων του, θα πρέπει να κάνει υποβολή, για να αποθηκευτούν αυτά, στη βάση δεδομένων.</p>
--	---

Σκαριφήματα

Username

Phone

Adress

Σκαρίφημα 10: Προβολή Στοιχείων Πελάτη

5.4.2 Σενάρια Χρήσης για τον Υπάλληλο

Στο σημείο αυτό, θα αναλυθούν και θα καταγραφούν τα σενάρια χρήσης του CIMaS για το χρήστη «υπάλληλο».

5.4.2.1 Περίπτωση χρήσης: Δημιουργία Issue

Σενάριο: Δημιουργία Issue

Για να μπορέσει ο υπάλληλος να δημιουργήσει ένα καινούριο Issue, θα πρέπει να εκτελέσει τα παρακάτω βήματα:

Βήματα Σεναρίου

1. Εισαγωγή χρήστη στο σύστημα
2. Επιλογή δημιουργίας νέου Issue
3. Καταγραφή πληροφοριών Issue
4. Επιλογή απευθυνόμενου πελάτη
5. Υποβολή δεδομένων

Αναλυτική Περιγραφή Βημάτων Σεναρίου

Βήμα 1: Εισαγωγή χρήστη στο σύστημα

Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!

Βήμα 2: Επιλογή δημιουργίας νέου Issue

Μετά την εισαγωγή του χρήστη στο σύστημα, θα εμφανίζονται οι επιλογές (τα μενού) του CIMaS. Από αυτά θα επιλέγει «New Issue».

Βήμα 3: Καταγραφή πληροφοριών Issue

Σε αυτό το βήμα, ο υπάλληλος θα πρέπει να καταχωρεί τις πληροφορίες του Issue, στη φόρμα που εμφανίζεται.

Βήμα 5: Επιλογή απευθυνόμενου χρήστη

Μετά την συμπλήρωση των παραπάνω πεδίων, ο υπάλληλος θα πρέπει να ενημερώσει ποιόν αφορά το συγκεκριμένο Issue. Αυτό θα γίνει επιλέγοντας από μία λίστα πελατών, το σωστό πελάτη.

Βήμα 6: Υποβολή δεδομένων

Με την ολοκλήρωση της συμπλήρωσης για το νέο issue, ο υπάλληλος θα υποβάλλει τα στοιχεία στη βάση δεδομένων, πατώντας το κουμπί submit.

Σκαρίφημα

Title	<input type="text"/>
Category	<input type="text" value="▼"/>
Arrival Date	Τρέχουσα Ημ/νια
Priority	<input type="text" value="▼"/>
Status	<input type="text" value="▼"/>
User	<input type="text" value="▼"/>
Content	<input type="text"/>
Attachment	<input type="text" value="..."/>
<input type="button" value="Submit"/>	

Σκαρίφημα 11: Δημιουργία Issue - Υπάλληλος

5.4.2.2 Περίπτωση χρήσης: Ενημέρωση Issue

Σενάριο: Ενημέρωση issue

Για να μπορέσει ο υπάλληλος να επεξεργαστεί το στάδιο που βρίσκεται κάποιο issue, θα πρέπει να εκτελέσει τα παρακάτω βήματα :

Βήματα Σεναρίου

1. Εισαγωγή χρήστη στο σύστημα
2. Επιλογή προβολής υπαρχόντων Issues
3. Επιλογή προβολής Issue
4. Επεξεργασία κατάστασης (status) Issue
5. Υποβολή δεδομένων

Βήμα 1: Εισαγωγή χρήστη στο σύστημα

Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!

Βήμα 2: Επιλογή προβολής υπαρχόντων Issues

Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues», την προβολή υπάρχοντων issues. Θα εμφανιστεί μία λίστα από issues που υπάρχουν.

Βήμα 3: Επιλογή προβολής συγκεκριμένου Issue

Στο βήμα αυτό, ο χρήστης θα επιλέξει το συγκεκριμένο issue που θέλει να επεξεργαστεί. Στην φόρμα που θα προβληθεί, θα αλλάξει το status του (δεν έχει ανατεθεί, έχει ανατεθεί, γίνονται εργασίες, ολοκληρώθηκε κτλ).

Βήμα 4: Επεξεργασία κατάστασης (status) Issue

Στο βήμα αυτό, ο υπάλληλος αλλάζει το status του Issue.

Βήμα5: Υποβολή δεδομένων

Με την ολοκλήρωση της συμπλήρωσης για το issue, ο υπάλληλος θα υποβάλλει τα στοιχεία στη βάση δεδομένων, πατώντας το κουμπί submit.

Σκαρίφημα

Title	bla bla bla
Category	bla bla bla
Arrival Date	Τρέχουσα Ημ/να
Priority	bla bla bla
Status	<input type="text" value="bla bla bla"/> ▼
User	bla bla bla
Content	bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
Attachment	<input type="text" value="bla bla bla"/> ...
<input type="button" value="Submit"/>	

Σκαρίφημα 12: Ενημέρωση Issue - Υπάλληλος

5.4.2.3 Περίπτωση χρήσης: Προσθήκη – Προβολή Σχολίων Issue

<p>Σενάριο: Προβολή σχολίων Issue Για να μπορέσει ο υπάλληλος να δει τα σχόλια που υπάρχουν για το Issue (είτε τα έχει δημιουργήσει κάποιος πελάτης, είτε ο διευθυντής για καθοδήγηση), θα πρέπει να εκτελέσει τα παρακάτω βήματα:</p> <p>Βήματα Σεναρίου</p> <ol style="list-style-type: none">1. Εισαγωγή χρήστη στο σύστημα2. Επιλογή προβολής υπαρχόντων Issues3. Επιλογή προβολής σχολίων Issue <p>Αναλυτική Περιγραφή Βημάτων Σεναρίου</p> <p><u>Βήμα 1: Εισαγωγή χρήστη στο σύστημα</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!</p>	<p><u>Βήμα 2: Επιλογή προβολής υπαρχόντων Issues</u> Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues», την προβολή υπάρχοντος issue. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με το status του issue (δεν έχει ανατεθεί, έχει ανατεθεί, γίνονται εργασίες κτλ), καθώς επίσης και ένας σύνδεσμος που θα οδηγεί στη λίστα των σχολίων που αφορούν στο συγκεκριμένο issue.</p> <p><u>Βήμα 3: Επιλογή προβολής σχολίων Issue</u> Στο βήμα αυτό, μετά την επιλογή του υπαλλήλου, θα προβάλλεται μια λίστα με τα σχόλια που υπάρχουν για το συγκεκριμένο Issue. Να σημειωθεί, ότι ο υπάλληλος μπορεί να δει σχόλια που ο πελάτης δεν μπορεί.</p>															
<p>Σκαριφήματα</p> <table border="1"><thead><tr><th>#</th><th>Comment</th><th>Creation Date</th><th>Issue</th><th>Username</th></tr></thead><tbody><tr><td>1</td><td><u>blabla</u></td><td>blabla</td><td>blabla</td><td>blabla</td></tr><tr><td>2</td><td><u>blabla</u></td><td>blabla</td><td>blabla</td><td>blabla</td></tr></tbody></table> <p>Σκαρίφημα 13: Προβολή Σχολίων σε Issue</p>		#	Comment	Creation Date	Issue	Username	1	<u>blabla</u>	blabla	blabla	blabla	2	<u>blabla</u>	blabla	blabla	blabla
#	Comment	Creation Date	Issue	Username												
1	<u>blabla</u>	blabla	blabla	blabla												
2	<u>blabla</u>	blabla	blabla	blabla												

Σενάριο: Δημιουργία σχόλιου σε Issue

Βάσει της περιγραφής του σεναρίου αυτού, ο υπάλληλος θα μπορεί να δημιουργήσει σχόλια, μέσα σε κάποιο από τα ήδη υπάρχοντα issue.

Βήματα Σεναρίου

1. Εισαγωγή χρήστη στο σύστημα
2. Επιλογή προβολής υπάρχοντων Issues
3. Επιλογή προβολής σχόλιων Issue
4. Επιλογή δημιουργίας νέου σχόλιου
5. Υποβολή δεδομένων

Αναλυτική Περιγραφή Βημάτων Σεναρίου

Βήμα 1: Εισαγωγή χρήστη στο σύστημα

Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!

Βήμα 2: Επιλογή προβολής υπάρχοντων Issues

Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues» την προβολή υπάρχοντος issue. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με της πληροφορίες του issue, ένας σύνδεσμος που θα οδηγεί στη λίστα των σχόλιων που είναι σχετικά με το συγκεκριμένο issue, και ένας σύνδεσμος που θα οδηγεί στη λίστα των tasks που αφορούν στο συγκεκριμένο issue.

Βήμα 3: Επιλογή προβολής σχόλιων Issue

Στο βήμα αυτό, μετά την επιλογή του πελάτη, θα προβάλλεται μια λίστα με τα σχόλια που υπάρχουν για το συγκεκριμένο Issue.

Βήμα 4: Επιλογή δημιουργίας νέου σχόλιου σε Issue

Στο βήμα αυτό, μετά την επιλογή του πελάτη, θα προβάλλεται μια φόρμα συμπλήρωσης για δημιουργία νέου σχόλιου.

Βήμα 5: Υποβολή δεδομένων

Μετά την ολοκλήρωση συμπλήρωσης, τα στοιχεία αυτά θα υποβάλλονται στη βάση δεδομένων, πατώντας το κουμπί submit.

Σκαριφήματα

The screenshot shows a web form titled 'Comment' for creating a comment on an issue. It contains the following elements:

- A large text input field labeled 'Comment'.
- An 'Access Level' dropdown menu with the value 'bla bla bla' and a downward arrow.
- A 'Creation Date' field with the value 'blabla'.
- A 'Submit' button at the bottom.

Σκαρίφημα 14: Δημιουργία Comment σε Issue

5.4.2.4 Περίπτωση χρήσης: Ενημέρωση Task

<p>Σενάριο: Ενημέρωση Task Για να μπορέσει ο υπάλληλος να επεξεργαστεί και να ενημερώσει την κατάσταση κάποιου task, θα πρέπει να εκτελέσει τα παρακάτω βήματα :</p> <p>Βήματα Σεναρίου</p> <ol style="list-style-type: none">1. Εισαγωγή χρήστη στο σύστημα2. Επιλογή προβολής υπαρχόντων Issues3. Επιλογή προβολής υπαρχόντων Tasks για το Issue4. Επιλογή προβολής συγκεκριμένου Task5. Επεξεργασία πληροφοριών του Task6. Υποβολή <p>Αναλυτική Περιγραφή Βημάτων Σεναρίου</p> <p><u>Βήμα 1: Εισαγωγή χρήστη στο σύστημα</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!</p> <p><u>Βήμα 2: Επιλογή προβολής υπαρχόντων Issues</u> Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues», την προβολή αυτών που ήδη υπάρχουν. Θα εμφανιστεί μία λίστα από issues.</p>	<p><u>Βήμα 3: Επιλογή προβολής υπαρχόντων Tasks</u> Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues» την προβολή υπαρχόντων issues. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με τις πληροφορίες για αυτό. Ένας σύνδεσμος θα οδηγεί στη λίστα των σχόλιων που υπάρχουν για το συγκεκριμένο issue, και ένας άλλος σύνδεσμος θα οδηγεί στην προβολή των tasks που υπάρχουν για αυτό.</p> <p><u>Βήμα 4: Επιλογή προβολής συγκεκριμένου Task</u> Στο βήμα αυτό, ο υπάλληλος θα επιλέξει το Task στο οποίο θέλει να κάνει τις αλλαγές.</p> <p><u>Βήμα 5: Επεξεργασία πληροφοριών του Task</u> Θα επεξεργαστεί τις πληροφορίες-στοιχεία, όπου είναι αναγκαίο.</p> <p><u>Βήμα 6: Υποβολή</u> Με την ολοκλήρωση των αλλαγών, ο υπάλληλος θα πρέπει να κάνει υποβολή αυτών, για να τοποθετηθούν οι πληροφορίες στους πίνακες του CIMaS.</p>
<p>Σκαριφήματα</p> <div data-bbox="592 1272 997 1814" style="border: 1px solid black; padding: 10px; margin: 0 auto; width: fit-content;"><p>Title bla bla bla</p><p>Assigned To <input type="text" value="bla bla bla"/></p><p>Start Date bla bla bla</p><p>End Date</p><p>Status <input type="text" value="bla bla bla"/></p><p>User bla bla bla</p><p>Content bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla</p><p style="text-align: center;"><input type="button" value="Submit"/></p></div> <p>Σκαρίφημα 15: Ενημέρωση Task</p>	

5.4.2.5 Περίπτωση χρήσης: Προσθήκη – Προβολή Σχολίων Task

<p>Σενάριο: Προβολή σχολίων Task Για να μπορέσει ο υπάλληλος να δει τα σχόλια που υπάρχουν για το Task, θα πρέπει να εκτελέσει τα παρακάτω βήματα:</p> <p>Βήματα Σεναρίου</p> <ol style="list-style-type: none">1. Εισαγωγή χρήστη στο σύστημα2. Επιλογή προβολής υπαρχόντων Issues3. Επιλογή προβολής υπαρχόντων Tasks4. Επιλογή προβολής σχολίων του Task <p>Αναλυτική Περιγραφή Βημάτων Σεναρίου</p> <p><u>Βήμα 1: Εισαγωγή χρήστη στο σύστημα</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!</p>	<p><u>Βήμα 2: Επιλογή προβολής υπαρχόντων Issues</u> Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues» την προβολή υπάρχοντος issue. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με το status του issue (δεν έχει ανατεθεί, έχει ανατεθεί, γίνονται εργασίες κτλ), ένας σύνδεσμος που θα οδηγεί στη λίστα των σχολίων που αναφέρονται στο συγκεκριμένο issue, καθώς και ένας ακόμα σύνδεσμος που θα οδηγεί στα tasks του.</p> <p><u>Βήμα 3: Επιλογή προβολής υπαρχόντων Tasks</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να επιλέξει από τα Issues το σύνδεσμο που θα οδηγεί στη λίστα των tasks που υπάρχουν για το συγκεκριμένο issue.</p> <p><u>Βήμα 4: Επιλογή προβολής σχολίων Task</u> Στο βήμα αυτό, μετά την επιλογή του πελάτη, θα προβάλλεται μια λίστα με τα σχόλια που αναφέρονται στο συγκεκριμένο task.</p>															
<p>Σκαριφήματα</p> <table border="1"><thead><tr><th>#</th><th>Comment</th><th>Creation Date</th><th>Issue</th><th>Username</th></tr></thead><tbody><tr><td>1</td><td><u>blabla</u></td><td>blabla</td><td>blabla</td><td>blabla</td></tr><tr><td>2</td><td><u>blabla</u></td><td>blabla</td><td>blabla</td><td>blabla</td></tr></tbody></table> <p>Σκαρίφημα 16: Προβολή Σχολίων σε Issue</p>		#	Comment	Creation Date	Issue	Username	1	<u>blabla</u>	blabla	blabla	blabla	2	<u>blabla</u>	blabla	blabla	blabla
#	Comment	Creation Date	Issue	Username												
1	<u>blabla</u>	blabla	blabla	blabla												
2	<u>blabla</u>	blabla	blabla	blabla												

Σενάριο: Δημιουργία σχόλιου σε Task

Στο σενάριο αυτού, ο υπάλληλος θα δημιουργήσει σχόλιο μέσα σε κάποιο από τα υπάρχοντα tasks.

Βήματα Σεναρίου

1. Εισαγωγή χρήστη στο σύστημα
2. Επιλογή προβολής υπάρχοντων Issues
3. Επιλογή προβολής υπάρχοντων Tasks
4. Επιλογή προβολής σχόλιων του Task
5. Επιλογή δημιουργίας νέου σχόλιου
6. Υποβολή

Αναλυτική Περιγραφή Βημάτων Σεναρίου**Βήμα 1: Εισαγωγή χρήστη στο σύστημα**

Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!

Βήμα 2: Επιλογή προβολής υπάρχοντων Issues

Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues», την προβολή υπάρχοντος issue. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με το status του issue (δεν έχει ανατεθεί, έχει ανατεθεί, γίνονται εργασίες κτλ), ένας σύνδεσμος που θα οδηγεί στη λίστα των σχόλιων που υπάρχουν για το συγκεκριμένο issue, καθώς και ένας ακόμα σύνδεσμος που θα οδηγεί στα tasks του.

Βήμα 3: Επιλογή προβολής υπάρχοντων Tasks

Στο βήμα αυτό, ο χρήστης θα πρέπει να επιλέξει από τα «Issues», το σύνδεσμο που θα οδηγεί στη λίστα των tasks που υπάρχουν για το συγκεκριμένο issue.

Βήμα 4: Επιλογή προβολής σχόλιων του Task

Στο βήμα αυτό, μετά την επιλογή του πελάτη θα προβάλλεται μια λίστα με τα σχόλια που υπάρχουν για το συγκεκριμένο task.

Βήμα 5: Επιλογή δημιουργίας νέου Task

Στο βήμα αυτό, μετά την επιλογή του πελάτη, θα προβάλλεται μια λίστα με τα σχόλια που υπάρχουν για το συγκεκριμένο task.

Βήμα 6: Υποβολή

Με την ολοκλήρωση των αλλαγών, ο υπάλληλος θα πρέπει να κάνει την υποβολή τους, για να τοποθετηθούν οι πληροφορίες στους πίνακες του CIMaS.

Σκαριφήματα

The screenshot shows a web form with the following elements:

- A text input field labeled "Comment".
- A dropdown menu labeled "Access Level" with the value "bla bla bla" and a downward arrow.
- A text input field labeled "Creation Date" with the value "blabla".
- A "Submit" button at the bottom.

Σκαρίφημα 17: Δημιουργία Comment σε Task

5.4.2.6 Περίπτωση χρήσης: Προβολή Ιστορικού Πελάτη

<p>Σενάριο: Προβολή ιστορικού Για να μπορέσει ο υπάλληλος να δει τις εργασίες που έχουν γίνει στο παρελθόν σχετικά με κάποιον πελάτη, θα πρέπει να εκτελέσει τα παρακάτω βήματα:</p> <p>Βήματα Σεναρίου</p> <ol style="list-style-type: none">1. Εισαγωγή χρήστη στο σύστημα2. Επιλογή προβολής ιστορικού Issues3. Επιλογή συγκεκριμένου πελάτη για προβολή ιστορικού	<p>Αναλυτική Περιγραφή Βημάτων Σεναρίου</p> <p><u>Βήμα 1: Εισαγωγή χρήστη στο σύστημα</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!</p> <p><u>Βήμα 2: Επιλογή προβολής ιστορικού Issues</u> Στη συνέχεια, ο χρήστης θα επιλέξει από το μενού «Issues» την προβολή ιστορικού. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με όλα του τα issues που έχουν τιμή για το πεδίο status closed.</p> <p><u>Βήμα 3: Επιλογή συγκεκριμένου πελάτη για προβολή ιστορικού</u> Τέλος, μέσω του drop down menu, ο υπάλληλος θα μπορεί να επιλέξει κάποιο συγκεκριμένο πελάτη για την προβολή του ιστορικού του.</p>
---	---

Σκαριφήματα

User:

#	Title	Category	Arrival	End date	Comments
1	Issue bla	blabla	blabla		<u>blabla</u>
	Task1 blabla		blabla	blabla	<u>blabla</u>
	Task2 blabla		blabla	blabla	<u>blabla</u>
2	Issue2 blabla	blabla	blabla		<u>blabla</u>
	Task blabla		blabla	blabla	<u>blabla</u>

Σκαρίφημα 18: Ιστορικό Πελάτη

5.4.3 Σενάρια Χρήσης για τον Διευθυντή

Στο σημείο αυτό, θα αναλυθούν και θα καταγραφούν τα σενάρια χρήσης του CIMaS για το χρήστη διευθυντή. Να σημειωθεί ότι ο χρήστης «manager» θα έχει πρόσβαση στο σύστημα τόσο από το front, όσο και από το back end. Στο front end θα μπορεί να κάνει σχεδόν ό,τι και ο υπάλληλος, ενώ από το back end, θα του παρέχονται παραπάνω δυνατότητες. Στην ανάλυση των σεναρίων χρήσης που ακολουθεί, το περιβάλλον χρήσης είναι το back end.

5.4.3.1 Περίπτωση χρήσης: Δημιουργία Issue

Σενάριο: Δημιουργία Issue

Για να μπορέσει ο διευθυντής να δημιουργήσει ένα νέο ζήτημα(issue), θα πρέπει να εκτελέσει τα παρακάτω βήματα:

Βήματα Σεναρίου

1. Εισαγωγή χρήστη στο σύστημα
2. Επιλογή δημιουργίας νέου Issue
3. Καταγραφή πληροφοριών Issue
4. Επιλογή απευθυνόμενου πελάτη
5. Υποβολή δεδομένων

Αναλυτική Περιγραφή Βημάτων Σεναρίου

Βήμα 1: Εισαγωγή χρήστη στο σύστημα

Στο βήμα αυτό, ο χρήστης διευθυντής θα πρέπει να κάνει εισαγωγή στο σύστημα.

Βήμα 2: Επιλογή δημιουργίας νέου Issue

Μετά την εισαγωγή του χρήστη στο σύστημα, θα εμφανίζονται οι επιλογές (τα μενού) του CIMaS. Από αυτά θα επιλέγει «Νέο Issue».

Βήμα 3: Καταγραφή πληροφοριών Issue

Σε αυτό το βήμα, ο διευθυντής θα πρέπει να καταχωρεί τις πληροφορίες του Issue, στη φόρμα που εμφανίζεται.

Βήμα 5: Επιλογή απευθυνόμενου χρήστη

Μετά την συμπλήρωση των παραπάνω πεδίων, ο διευθυντής θα πρέπει να ενημερώσει ποιόν αφορά το συγκεκριμένο Issue. Αυτό θα γίνει επιλέγοντας από μία λίστα πελατών, το σωστό πελάτη.

Βήμα 6: Υποβολή δεδομένων

Με την ολοκλήρωση της συμπλήρωσης του νέου issue, ο διευθυντής θα υποβάλλει τα στοιχεία στη βάση δεδομένων, πατώντας το κουμπί submit.

Σκαρίφημα

Title	<input type="text"/>
Category	<input type="text" value="▼"/>
Arrival Date	Τρέχουσα Ημ/νια
Priority	<input type="text" value="▼"/>
Status	<input type="text" value="▼"/>
User	<input type="text" value="▼"/>
Content	<input type="text"/>
Attachment	<input type="text" value="..."/>
<input type="button" value="Submit"/>	

Σκαρίφημα 19: Δημιουργία Issue – Διευθυντής

5.4.3.2 Περίπτωση χρήσης: Ενημέρωση Issue – Προβολή Issue

Σενάριο: Προβολή issue

Για να μπορέσει ο διευθυντής να ενημερωθεί για το στάδιο εργασιών που βρίσκεται κάποιο issue, θα πρέπει να εκτελέσει τα παρακάτω βήματα :

Βήματα Σεναρίου

1. Εισαγωγή χρήστη στο σύστημα
2. Επιλογή προβολής υπαρχόντων Issues

Αναλυτική Περιγραφή Βημάτων Σεναρίου

Βήμα 1: Εισαγωγή χρήστη στο σύστημα

Στο βήμα αυτό, ο χρήστης θα πρέπει να εισέλθει στο σύστημα.

Βήμα 2: Επιλογή προβολής υπαρχόντων Issues

Έπειτα θα πρέπει να επιλέξει την επιλογή «Issues», για να του εμφανιστούν τα υπάρχοντα issues.

Σκαριφήματα

#	Title	Category	Arrival	Priority	Status	Username	ID	Action	Comments	Attachments
1	<u>blabla</u>	blabla	blabla	blabla	blabla	blabla	68	<u>blabla</u>	<u>blabla</u>	<u>blabla</u>
2	<u>blabla</u>	blabla	blabla	blabla	blabla	blabla	83	<u>blabla</u>	<u>blabla</u>	<u>blabla</u>

Σκαρίφημα 20: Multi Issue View

Σενάριο: Ενημέρωση issue

Για να μπορέσει ο manager να ενημερώσει-επεξεργαστεί κάποιο issue, θα πρέπει να εκτελέσει τα παρακάτω βήματα :

Βήματα Σεναρίου

1. Εισαγωγή χρήστη στο σύστημα
2. Επιλογή προβολής υπαρχόντων Issues (multi issue view)
3. Επιλογή προβολής Issue (single issue view)
4. Επεξεργασία κατάστασης (status) Issue
5. Υποβολή δεδομένων

Αναλυτική Περιγραφή Βημάτων Σεναρίου

Βήμα 1: Εισαγωγή χρήστη στο σύστημα

Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!

Βήμα 2: Επιλογή προβολής υπαρχόντων Issues

Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues» την προβολή υπαρχόντων issues. Θα εμφανιστεί μία λίστα από issues που υπάρχουν.

Βήμα 3: Επιλογή προβολής Issue

Στο βήμα αυτό, ο χρήστης θα επιλέξει το συγκεκριμένο issue που θέλει να επεξεργαστεί. Στην φόρμα που θα προβληθεί, θα αλλάξει το status του (δεν έχει ανατεθεί, έχει ανατεθεί, γίνονται εργασίες, ολοκληρώθηκε κτλ).

Βήμα 4: Επεξεργασία κατάστασης (status) Issue

Στο βήμα αυτό, ο διευθυντής αλλάζει το status του Issue.

Βήμα 5: Υποβολή δεδομένων

Με την ολοκλήρωση της συμπλήρωσης για το status του issue, θα υποβάλλει τα στοιχεία στη βάση δεδομένων, πατώντας το κουμπί submit.

Σκαρίφημα

Title	<input type="text" value="bla bla bla"/>
Category	<input type="text" value="bla bla bla"/> ▼
Arrival Date	<input type="text" value="bla bla bla"/>
Priority	<input type="text" value="bla bla bla"/> ▼
Status	<input type="text" value="bla bla bla"/> ▼
User	<input type="text" value="bla bla bla"/>
Content	<input type="text" value="bla bla bla"/> <input type="text" value="bla bla"/> <input type="text" value="bla bla bla"/> <input type="text" value="bla bla"/> <input type="text" value="bla bla bla"/>
Attachment	<input type="text" value="bla bla bla"/> ...
<input type="button" value="Submit"/>	

Σκαρίφημα 21: Ενημέρωση Issue - Διευθυντής

5.4.3.3 Περίπτωση χρήσης: Προσθήκη – Προβολή Σχολίων Issue

<p>Σενάριο: Προβολή σχολίων Issue Για να μπορέσει ο διευθυντής να δει τα σχόλια που υπάρχουν για το Issue (είτε τα έχει δημιουργήσει κάποιος πελάτης, είτε κάποιος υπάλληλος, είτε ο ίδιος), θα πρέπει να εκτελέσει τα παρακάτω βήματα:</p> <p>Βήματα Σεναρίου</p> <ol style="list-style-type: none"> 1. Εισαγωγή χρήστη στο σύστημα 2. Επιλογή προβολής υπαρχόντων Issues 3. Επιλογή προβολής σχολίων Issue <p>Αναλυτική Περιγραφή Βημάτων Σεναρίου <u>Βήμα 1: Εισαγωγή χρήστη στο σύστημα</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!</p>	<p><u>Βήμα 2: Επιλογή προβολής υπαρχόντων Issues</u> Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues» την προβολή υπαρχόντος issue. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με το status του issue (δεν έχει ανατεθεί, έχει ανατεθεί, γίνονται εργασίες κτλ), καθώς και ένας σύνδεσμος που θα οδηγεί στη λίστα των σχολίων που αναφέρονται στο συγκεκριμένο issue.</p> <p><u>Βήμα 3: Επιλογή προβολής σχολίων Issue</u> Στο βήμα αυτό, μετά την επιλογή του διευθυντή, θα προβάλλεται μια λίστα με τα σχόλια που υπάρχουν για το συγκεκριμένο Issue. Να σημειωθεί, ότι ο διευθυντής μπορεί να δει σχόλια που ο πελάτες και υπάλληλοι δεν μπορούν.</p>																		
<p>Σκαριφήματα</p> <table border="1" data-bbox="406 981 1212 1205"> <thead> <tr> <th>#</th> <th>Comment</th> <th>Creation Date</th> <th>Issue</th> <th>Username</th> <th>Access Level</th> </tr> </thead> <tbody> <tr> <td>1</td> <td><u>blabla</u></td> <td>blabla</td> <td>blabla</td> <td>blabla</td> <td>blabla</td> </tr> <tr> <td>2</td> <td><u>blabla</u></td> <td>blabla</td> <td>blabla</td> <td>blabla</td> <td>blabla</td> </tr> </tbody> </table> <p>Σκαρίφημα 22: Προβολή Σχολίων σε Issue</p>		#	Comment	Creation Date	Issue	Username	Access Level	1	<u>blabla</u>	blabla	blabla	blabla	blabla	2	<u>blabla</u>	blabla	blabla	blabla	blabla
#	Comment	Creation Date	Issue	Username	Access Level														
1	<u>blabla</u>	blabla	blabla	blabla	blabla														
2	<u>blabla</u>	blabla	blabla	blabla	blabla														

Σενάριο: Δημιουργία σχόλιου σε Issue

Βάση της περιγραφής του σεναρίου αυτού, ο διευθυντής θα μπορεί να δημιουργήσει σχόλια μέσα σε κάποιο από τα ήδη υπάρχοντα issue.

Βήματα Σεναρίου

1. Εισαγωγή χρήστη στο σύστημα
2. Επιλογή προβολής υπαρχόντων Issues
3. Επιλογή προβολής σχόλιων Issue
4. Επιλογή δημιουργίας νέου σχόλιου
5. Υποβολή δεδομένων

Αναλυτική Περιγραφή Βημάτων Σεναρίου

Βήμα 1: Εισαγωγή χρήστη στο σύστημα

Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!

Βήμα 2: Επιλογή προβολής υπαρχόντων Issues

Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues» την προβολή υπάρχοντος issue. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με τις πληροφορίες του issue, ένας σύνδεσμος που θα οδηγεί στη λίστα των σχόλιων που αφορούν στο συγκεκριμένο issue και ένας σύνδεσμος που θα οδηγεί στη λίστα των tasks που είναι σχετικά με το συγκεκριμένο issue.

Βήμα 3: Επιλογή προβολής σχόλιων Issue

Στο βήμα αυτό, μετά την επιλογή του διευθυντή, θα προβάλλεται μια λίστα με τα σχόλια που υπάρχουν για το συγκεκριμένο Issue.

Βήμα 4: Δημιουργία σχόλιου σε Issue

Στο βήμα αυτό, μετά την επιλογή του διευθυντή, θα προβάλλεται μια φόρμα συμπλήρωσης για δημιουργία νέου σχόλιου.

Βήμα 5: Υποβολή δεδομένων

Μετά την ολοκλήρωση συμπλήρωσης, τα στοιχεία αυτά θα υποβάλλονται στη βάση δεδομένων, πατώντας το κουμπί submit.

Σκαριφήματα

The screenshot shows a web form titled 'Σκαριφήματα'. It contains the following elements:

- A text area labeled 'Comment'.
- A dropdown menu labeled 'Access Level' with the value 'bla bla bla' and a downward arrow.
- A text field labeled 'Creation Date' with the value 'blabla'.
- A button labeled 'Submit'.

Σκαρίφημα 23: Δημιουργία Comment σε Issue

5.4.3.4 Διάσπαση Issue σε Task

<p>Σενάριο: Διάσπαση Issue σε Task Βάσει της περιγραφής του σεναρίου αυτού, ο διευθυντής θα μπορεί να δημιουργήσει task, μέσα σε κάποιο από τα υπάρχοντα issue.</p> <p>Βήματα Σεναρίου</p> <ol style="list-style-type: none">1. Εισαγωγή χρήστη στο σύστημα2. Επιλογή προβολής υπαρχόντων Issues3. Επιλογή προβολής Tasks4. Επιλογή δημιουργίας Task σε issue.5. Υποβολή δεδομένων <p>Αναλυτική Περιγραφή Βημάτων Σεναρίου</p> <p><u>Βήμα 1: Εισαγωγή χρήστη στο σύστημα</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!</p>	<p><u>Βήμα 2: Επιλογή προβολής υπαρχόντων Issues</u> Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues» την προβολή υπάρχοντος issue. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με τις πληροφορίες του issue, ένας σύνδεσμος που θα οδηγεί στη λίστα των σχόλιων που αναφέρονται στο συγκεκριμένο issue και ένας σύνδεσμος που θα οδηγεί στη λίστα των tasks που αφορούν στο συγκεκριμένο issue.</p> <p><u>Βήμα 3: Επιλογή προβολής Tasks</u> Στο βήμα αυτό, μετά την επιλογή του διευθυντή, θα προβάλλεται το σημείο στο οποίο εμφανίζεται η λίστα με τα tasks.</p> <p><u>Βήμα 4: Επιλογή δημιουργίας Task σε Issue</u> Στο βήμα αυτό, μετά την επιλογή του διευθυντή, θα προβάλλεται μια φόρμα συμπλήρωσης για δημιουργία νέου task.</p> <p><u>Βήμα 5: Υποβολή δεδομένων</u> Μετά την ολοκλήρωση συμπλήρωσης, τα στοιχεία αυτά θα υποβάλλονται στη βάση δεδομένων, πατώντας το κουμπί submit.</p>
<p>Σκαριφήματα</p> <div data-bbox="580 1267 1000 1816" style="border: 1px solid black; padding: 10px; margin: 0 auto; width: fit-content;"><p>Title <input type="text"/></p><p>Assigned To <input type="text"/></p><p>Start Date Τρέχουσα Ημ/νια</p><p>End Date</p><p>Status <input type="text"/></p><p>Content <input type="text"/></p><p style="text-align: center;"><input type="button" value="Submit"/></p></div> <p>Σκαρίφημα 24: Δημιουργία Task</p>	

5.4.3.5 Περίπτωση χρήσης: Προβολή - Ενημέρωση Task

Σενάριο: Προβολή Task

Στο σενάριο αυτό, ο διευθυντής θα μπορεί να δει τις εργασίες που έχουν γίνει σε κάποιο Issue. Κάθε issue έχει τουλάχιστον ένα task, επίσης μπορεί να έχει και σχόλια. Από το σενάριο αυτό θα μπορεί να δει τα tasks και τα σχόλια του issue.

Βήματα Σεναρίου

1. Εισαγωγή χρήστη στο σύστημα
2. Επιλογή προβολής υπαρχόντων Issues
3. Επιλογή προβολής σχόλιων Issue
4. Επιλογή προβολής Tasks
5. Επιλογή προβολής Task
6. Επιλογή προβολής σχόλιων Tasks

Αναλυτική Περιγραφή Βημάτων Σεναρίου

Βήμα 1: Εισαγωγή χρήστη στο σύστημα

Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!

Βήμα 2: Επιλογή προβολής υπαρχόντων Issues

Στο βήμα 2, ο χρήστης θα επιλέξει από το μενού Issues την προβολή υπάρχοντος issue. Στον πίνακα που θα προβληθεί, θα υπάρχει ένας πεδίο με το status του issue (δεν έχει ανατεθεί, έχει ανατεθεί, γίνονται εργασίες κτλ), ένας σύνδεσμος που θα οδηγεί στη λίστα των tasks και ένας σύνδεσμος που θα οδηγεί στη λίστα των σχόλιων που υπάρχουν για το συγκεκριμένο issue.

Βήμα 3: Επιλογή προβολής σχόλιων Issue

Στο βήμα αυτό, μετά την επιλογή του πελάτη θα εμφανίζεται μια λίστα από τα σχόλια που υπάρχουν για το συγκεκριμένο issue.

Βήμα 4: Επιλογή προβολής Tasks

Στο βήμα αυτό, ο πελάτης θα βλέπει όλα τα tasks που έχουν δημιουργηθεί για το ζήτημά του. Στα tasks, επίσης θα υπάρχει ένας σύνδεσμος που θα οδηγεί στα σχόλια που υπάρχουν.

Βήμα 5: Επιλογή προβολής Task

Στη συνέχεια θα μπορεί να επιλέξει να εμφανιστεί μόνο του το task, για να προβληθούν με τον τρόπο αυτό όλες οι λεπτομέρειες του.

Βήμα 6: Επιλογή προβολής σχόλιων Task

Σε αυτό το βήμα, ο πελάτης θα μπορεί να αναγνώσει όλα τα σχόλια (λεπτομέρειες) που υπάρχουν για το συγκεκριμένο task.

Σκαριφήματα

#	Title	Start Date	End Date	Status	Issue	Username	Assigned To	Comments
1	blabla	blabla	blabla	blabla	blabla	blabla	blabla	blabla
2	blabla	blabla	blabla	blabla	blabla	blabla	blabla	blabla

Σκαρίφημα 25: Multi Task View

Σενάριο: Ενημέρωση Task

Στο σενάριο αυτό, ο διευθυντής θα μπορεί να κάνει αλλαγές σε κάποιο υπάρχον task.

Βήματα Σεναρίου

1. Εισαγωγή χρήστη στο σύστημα
2. Επιλογή προβολής υπαρχόντων Issues
3. Επιλογή προβολής Tasks
4. Επιλογή προβολής Task
5. Επεξεργασία στοιχείων Task
6. Υποβολή στοιχείων

Αναλυτική Περιγραφή Βημάτων Σεναρίου

Βήμα 1: Εισαγωγή χρήστη στο σύστημα

Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!

Βήμα 2: Επιλογή προβολής υπαρχόντων Issues

Στο βήμα 2, ο χρήστης θα επιλέξει από το μενού Issues την προβολή υπάρχοντος issue. Στον πίνακα που θα προβληθεί, θα υπάρχει ένας πεδίο με το status του issue (δεν έχει ανατεθεί, έχει ανατεθεί, γίνονται εργασίες κτλ), ένας σύνδεσμος που θα οδηγεί στη λίστα των tasks και ένας σύνδεσμος που θα οδηγεί στη λίστα των σχόλιων που υπάρχουν για το συγκεκριμένο issue.

Βήμα 3: Επιλογή προβολής Tasks

Στο βήμα αυτό, ο πελάτης θα βλέπει όλα τα tasks που έχουν δημιουργηθεί για το ζήτημά του. Στα tasks, επίσης θα υπάρχει ένας σύνδεσμος που θα οδηγεί στα σχόλια που υπάρχουν.

Βήμα 4: Επιλογή προβολής Task

Στη συνέχεια θα μπορεί να επιλέξει να εμφανιστεί μόνο του το task, για να προβληθούν με τον τρόπο αυτό όλες οι λεπτομέρειες του.

Βήμα 5: Επεξεργασία στοιχείων Task

Στη φόρμα που εμφανίζεται, ο διευθυντής επεξεργάζεται τα στοιχεία που επιθυμεί.

Βήμα 6: Υποβολή στοιχείων

Μετά τις αλλαγές τις επιλογής του, ο διευθυντής κάνει αποθήκευση αυτών.

Σκαριφήματα

Title	<input type="text" value="bla bla bla"/>
Assigned To	<input type="text" value="bla bla bla"/>
Start Date	Τρέχουσα Ημέρα
End Date	
Status	<input type="text" value="bla bla bla"/>
Content	<input type="text" value="bla bla bla bla bla bla bla bla bla bla"/>
	<input type="button" value="Submit"/>

Σκαρίφημα 26: Επεξεργασία Task

5.4.3.6 Περίπτωση χρήσης: Προσθήκη – Προβολή Σχολίων Task

<p>Σενάριο: Προβολή σχολίων Task Για να μπορέσει ο διευθυντής να δει τα σχόλια που υπάρχουν για κάποιο Task, θα πρέπει να εκτελέσει τα παρακάτω βήματα:</p> <p>Βήματα Σεναρίου</p> <ol style="list-style-type: none">1. Εισαγωγή χρήστη στο σύστημα2. Επιλογή προβολής υπαρχόντων Issues3. Επιλογή προβολής υπαρχόντων Tasks4. Επιλογή προβολής σχολίων του Task <p>Αναλυτική Περιγραφή Βημάτων Σεναρίου</p> <p><u>Βήμα 1: Εισαγωγή χρήστη στο σύστημα</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!</p>	<p><u>Βήμα 2: Επιλογή προβολής υπαρχόντων Issues</u> Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues» την προβολή υπάρχοντος issue. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με το status του issue (δεν έχει ανατεθεί, έχει ανατεθεί, γίνονται εργασίες κτλ), ένας σύνδεσμος που θα οδηγεί στη λίστα των σχολίων που αναφέρονται στο συγκεκριμένο issue, καθώς και ένας ακόμα σύνδεσμος που θα οδηγεί στα tasks του issue.</p> <p><u>Βήμα 3: Επιλογή προβολής υπαρχόντων Tasks</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να επιλέξει από τα Issues το σύνδεσμο που θα οδηγεί στη λίστα των tasks που υπάρχουν για το συγκεκριμένο issue.</p> <p><u>Βήμα 4: Επιλογή προβολής σχολίων του Task</u> Στο βήμα αυτό, μετά την επιλογή του διευθυντή, θα προβάλλεται μια λίστα με τα σχόλια που υπάρχουν για το συγκεκριμένο task.</p>															
<p>Σκαριφήματα</p> <table border="1"><thead><tr><th>#</th><th>Comment</th><th>Creation Date</th><th>Issue</th><th>Username</th></tr></thead><tbody><tr><td>1</td><td><u>blabla</u></td><td>blabla</td><td>blabla</td><td>blabla</td></tr><tr><td>2</td><td><u>blabla</u></td><td>blabla</td><td>blabla</td><td>blabla</td></tr></tbody></table> <p>Σκαρίφημα 27: Προβολή Σχολίων σε Issue</p>		#	Comment	Creation Date	Issue	Username	1	<u>blabla</u>	blabla	blabla	blabla	2	<u>blabla</u>	blabla	blabla	blabla
#	Comment	Creation Date	Issue	Username												
1	<u>blabla</u>	blabla	blabla	blabla												
2	<u>blabla</u>	blabla	blabla	blabla												

Σενάριο: Δημιουργία σχόλιου σε Task

Στο σενάριο αυτό, ο διευθυντής μπορεί να δημιουργήσει σχόλιο μέσα σε κάποιο από τα ήδη υπάρχοντα tasks.

Βήματα Σεναρίου

1. Εισαγωγή χρήστη στο σύστημα
2. Επιλογή προβολής υπάρχόντων Issues
3. Επιλογή προβολής υπάρχόντων Tasks
4. Επιλογή προβολής σχόλιων του Task
5. Επιλογή δημιουργίας νέου σχόλιου σε Task
6. Υποβολή

Αναλυτική Περιγραφή Βημάτων Σεναρίου**Βήμα 1: Εισαγωγή χρήστη στο σύστημα**

Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!

Βήμα 2: Επιλογή προβολής υπάρχόντων Issues

Στο βήμα αυτό, ο χρήστης θα επιλέξει από το μενού «Issues», την προβολή υπάρχοντος issue. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με το status του issue (δεν έχει ανατεθεί, έχει ανατεθεί, γίνονται εργασίες κτλ), ένας σύνδεσμος που θα οδηγεί στη λίστα των σχόλιων που αναφέρονται στο συγκεκριμένο issue, καθώς και ένας ακόμα σύνδεσμος που θα οδηγεί στα tasks του issue.

Βήμα 3: Επιλογή προβολής υπάρχόντων Issues

Στο βήμα αυτό, ο χρήστης θα πρέπει να επιλέξει από τα Issues το σύνδεσμο που θα οδηγεί στη λίστα των tasks που υπάρχουν για το συγκεκριμένο issue.

Βήμα 4: Επιλογή προβολής σχόλιων του Task

Στο βήμα αυτό, μετά την επιλογή του διευθυντή, θα προβάλλεται μια λίστα με τα σχόλια που υπάρχουν για το συγκεκριμένο task.

Βήμα 5: Επιλογή δημιουργίας νέου σχόλιου σε Task

Στο βήμα αυτό, μετά την επιλογή του χρήστη, θα προβάλλεται μια λίστα με τα σχόλια που υπάρχουν για το συγκεκριμένο task.

Βήμα 6: Υποβολή

Με την ολοκλήρωση των αλλαγών, ο διευθυντής θα πρέπει να κάνει την υποβολή αυτών, για να τοποθετηθούν οι πληροφορίες στους πίνακες του CIMaS.

Σκαριφήματα

The screenshot shows a form with the following elements:

- Comment:** A large empty text input area.
- Access Level:** A dropdown menu with the text 'bla bla bla' and a downward arrow.
- Creation Date:** A text input field containing the text 'blabla'.
- Submit:** A button at the bottom of the form.

Σκαρίφημα 28: Δημιουργία Comment σε Task

5.4.3.7 Περίπτωση χρήσης: Προβολή Ιστορικού Πελάτη

<p>Σενάριο: Προβολή ιστορικού Για να μπορέσει ο διευθυντής να δει τις εργασίες που έχουν γίνει στο παρελθόν για κάποιον πελάτη, θα πρέπει να εκτελέσει τα παρακάτω βήματα:</p> <p>Βήματα Σεναρίου</p> <ol style="list-style-type: none">1. Εισαγωγή χρήστη στο σύστημα2. Επιλογή προβολής ιστορικού Issues3. Επιλογή συγκεκριμένου πελάτη για προβολή ιστορικού	<p>Αναλυτική Περιγραφή Βημάτων Σεναρίου</p> <p><u>Βήμα 1: Εισαγωγή χρήστη στο σύστημα</u> Στο βήμα αυτό, ο χρήστης θα πρέπει να κάνει εισαγωγή στο σύστημα μέσω της έτοιμης εφαρμογής login του Joomla!</p> <p><u>Βήμα 2: Επιλογή προβολής ιστορικού Issues</u> Στη συνέχεια, ο χρήστης θα επιλέξει από το μενού «Issues» την προβολή ιστορικού. Στην φόρμα που θα προβληθεί, θα υπάρχει ένας πίνακας με όλα τα issues που έχουν τιμή για το πεδίο status closed.</p> <p><u>Βήμα 3: Επιλογή συγκεκριμένου πελάτη για προβολή ιστορικού</u> Τέλος, μέσω του drop down menu, ο διευθυντής θα μπορεί να επιλέξει κάποιο συγκεκριμένο πελάτη για την προβολή του ιστορικού του.</p>
---	--

Σκαριφήματα

User:

#	Title	Category	Arrival	End date	Comments
1	Issue bla	blabla	blabla		<u>blabla</u>
	Task1 blabla		blabla	blabla	<u>blabla</u>
	Task2 blabla		blabla	blabla	<u>blabla</u>
2	Issue2 blabla	blabla	blabla		<u>blabla</u>
	Task blabla		blabla	blabla	<u>blabla</u>

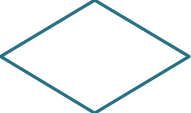


Σκαρίφημα 29: Ιστορικό Πελάτη

5.5 Διάγραμμα Ροής Εργασιών

Στα Διαγράμματα Ροής Εργασιών(Work Flow Diagrams), καθορίζονται τα βήματα με τα οποία εκτελούνται οι επιλογές για τη χρήση του συστήματος. Δηλαδή σε ποιά σειρά βημάτων είναι η καταχώρηση ενός νέου Issue, τι ενέργειες πρέπει να έχουν γίνει πριν από αυτό κτλ.

Στην ενότητα αυτή, παρουσιάζονται τα διαγράμματα ροής εργασιών για τον πελάτη, τον υπάλληλο και το διευθυντή.

Για την καλύτερη κατανόηση των διαγραμμάτων που ακολουθούν, στον Πίνακα 2 παρουσιάζονται τα σχήματα που χρησιμοποιήθηκαν και η ερμηνεία τους.

Σχήμα	Ερμηνεία
	Απόφαση
	Δεδομένα
	Ροή εργασιών που αναπτύσσεται σε άλλο διάγραμμα

Πίνακας 2: Ερμηνεία Χρησιμοποιημένων Σχημάτων των Work Flows

5.5.1 Για τον Πελάτη

Αρχικά θα πρέπει ο χρήστης «πελάτης» να κάνει εισαγωγή στο σύστημα. Με την είσοδό του στο σύστημα, φορτώνονται οι προσωπικές του ρυθμίσεις και τα δεδομένα που αφορούν τον ίδιο.

Στη συνέχεια, με τις επιλογές του, ενημερώνει το σύστημα, για το τι θα ήθελε να κάνει σε σχέση με την εφαρμογή εποπτείας των ζητημάτων CIMaS, η οποία θα υπάρχει εγκατεστημένη στη σελίδα (από το σχετικό μενού εργασιών που έχει δημιουργηθεί).

Στο σημείο αυτό να υπενθυμίσουμε, ότι ο πελάτης από το CIMaS μπορεί να δημιουργεί νέο Issue, να ζητάει να του προβληθούν όλες οι ενέργειες που έχουν γίνει για την επίλυση του ζητήματος (tasks και σχόλιων σε task), να δημιουργεί νέα σχόλια και να επεξεργάζεται τα προσωπικά δεδομένα που έχει δηλώσει.

Αν επιλέξει να επεξεργαστεί τα προσωπικά δεδομένα που έχει υποβάλλει, επιλέγει την αντίστοιχη ενέργεια, κάνει τις αλλαγές που επιθυμεί και στη συνέχεια υποβάλλει τα νέα επεξεργασμένα στοιχεία.

Αν θέλει να δημιουργήσει νέο Issue, επιλέγει τη συγκεκριμένη ενέργεια και συμπληρώνει τα δεδομένα. Στη συνέχεια, ενημερώνει το σύστημα, κάνοντας υποβολή της φόρμας που έχει συμπληρώσει.

Στην περίπτωση που δεν θέλει να δημιουργήσει κάποιο νέο issue, επιλέγει την ενέργεια προβολής υπάρχοντος Issue, μέσω της οποίας, προβάλλονται οι εργασίες που δεν έχουν κλείσει, καθώς και το ιστορικό από παλιότερες ενέργειες. Στο επόμενο βήμα, μπορεί να επιλέξει μια από τις εξής ενέργειες:

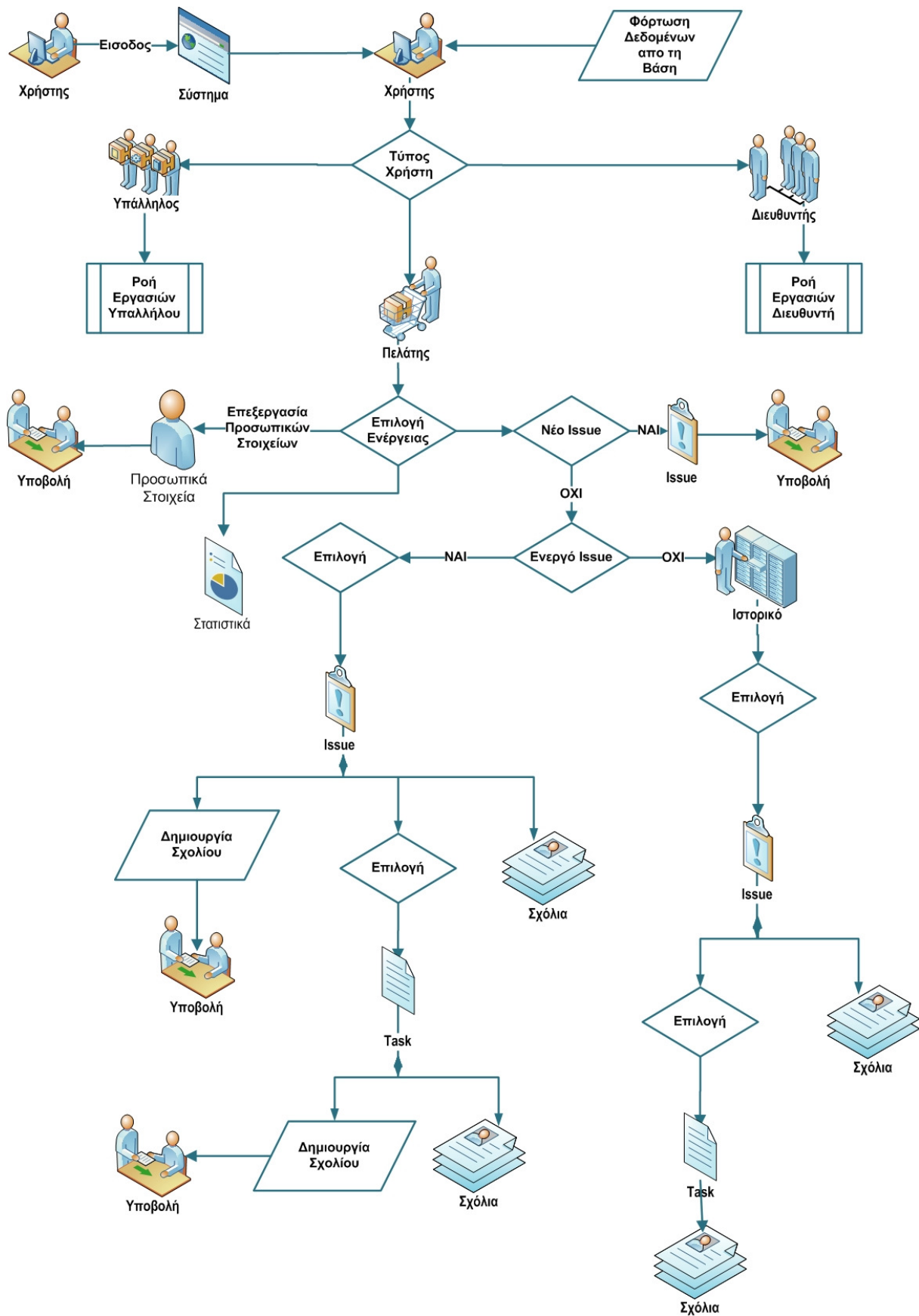
- Να δημιουργήσει νέο σχόλιο (για κάποια νέα πληροφορία που πιθανώς υπάρχει ή για κάποια ερώτηση που θέλει να υποβάλλει) και στη συνέχεια να το καταχωρήσει.
- Να προβάλλει τα σχόλια που ήδη υπάρχουν.
- Να επιλέξει την εμφάνιση των εργασιών (tasks) που έχουν γίνει για το συγκεκριμένο Issue.

Στην περίπτωση που επιλέξει να εμφανιστούν τα tasks στα οποία έχει διασπαστεί το αρχικό μήνυμα, για την διεκπεραίωσή του θα πρέπει να επιλέξει από τη λίστα κάποιο συγκεκριμένο task, για να περιγραφθούν αναλυτικά οι εργασίες που έχουν καταχωρηθεί και έχουν σχέση με αυτό. Επίσης μπορεί να επιλέξει μια από τις παρακάτω ενέργειες:

- Να δημιουργήσει νέο σχόλιο (για κάποια νέα πληροφορία που πιθανώς υπάρχει ή για κάποια ερώτηση που θέλει να υποβάλλει) και στη συνέχεια να το καταχωρήσει.
- Να προβάλλει τα σχόλια που ήδη υπάρχουν

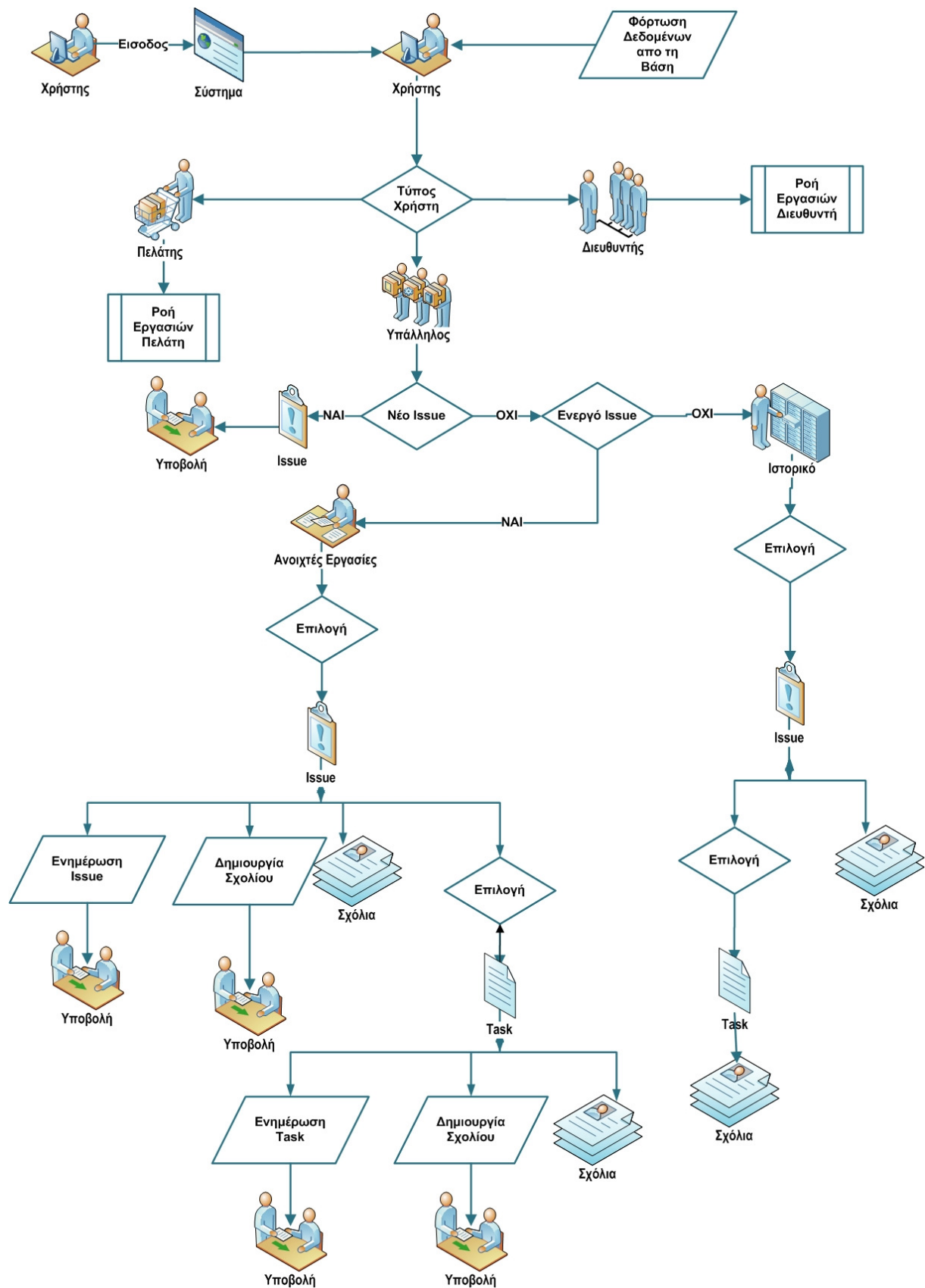
Όλα αυτά που αναφέρουμε για τη ροή εργασιών του πελάτη, περιγράφονται σχηματικά στο Διάγραμμα 1.

Όλες οι ενέργειες από αυτή την ομάδα χρηστών, γίνονται από το front end της ιστοσελίδας, όπου έχουν δημιουργηθεί links για την επικοινωνία του πελάτη με το σύστημα.



Διάγραμμα 1: Ροή Εργασιών Πελάτη

5.5.2 Για τον Υπάλληλο



Διάγραμμα 2: Ροή Εργασιών Υπαλλήλου

Στο Διάγραμμα 2, παρουσιάζεται σχηματικά η ακολουθία της ροής των εργασιών που μπορεί να πραγματοποιήσει ο χρήστης υπάλληλος.

Αρχικά θα πρέπει να γίνει η είσοδος του στο σύστημα. Η εισαγωγή του γίνεται με το να αποσταλούν τα στοιχεία του για έλεγχο ορθότητας, γίνεται ταυτοποίηση χρήστη, φορτώνονται τα προσωπικά του δεδομένα και τέλος φορτώνονται οι ρυθμίσεις και τα δικαιώματά του από το σύστημα.

Στη συνέχεια, από τις επιλογές του, ενημερώνει το σύστημα τι θα ήθελε να κάνει σε σχέση με την εφαρμογή CIMaS που θα υπάρχει εγκατεστημένη.

Αν θέλει να δημιουργήσει νέο Issue, επιλέγει τη συγκεκριμένη ενέργεια και συμπληρώνει τα στοιχεία που χρειάζονται. Μετά ενημερώνει το σύστημα, κάνοντας υποβολή της φόρμας που έχει συμπληρώσει.

Στην περίπτωση που δεν θέλει να δημιουργήσει κάποιο νέο issue, υπάρχουν οι εξής επιλογές προβολή ανοιχτών εργασιών ή προβολή ιστορικού.

Στην προβολή ιστορικού, μπορεί να ελέγξει εργασίες που έχουν γίνει στο παρελθόν για κάποιο πελάτη, ή για να θυμηθεί τα στάδια –βήματα που είχαν εκτελεστεί. Αν επιλέξει προβολή ιστορικού, του παρέχονται οι εξής επιλογές:

- Να επιλέξει συγκεκριμένο issue για προβολή
- Να επιλέξει προβολή σχολίων του συγκεκριμένου issue
- Να επιλέξει προβολή συγκεκριμένου Task
- Να επιλέξει προβολή σχολίων του συγκεκριμένου Task

Στην προβολή ανοιχτών εργασιών, προβάλλονται οι εργασίες που δεν έχουν κλείσει. Στο επόμενο βήμα, μπορεί να επιλέξει μια από τις εξής ενέργειες:

- Να δημιουργήσει νέο σχόλιο (για κάποια νέα πληροφορία) και στη συνέχεια να την υποβάλλει
- Να προβάλλει σχόλια που ήδη υπάρχουν
- Να ενημερώσει το Issue για την ολοκλήρωσή του, μόνο στην περίπτωση που έχουν ολοκληρωθεί οι εργασίες (tasks).
- Να επιλέξει να εμφανιστούν οι εργασίες (tasks) που υπάρχουν για το συγκεκριμένο Issue.

Στην περίπτωση που επιλέξει να εμφανιστούν τα tasks στα οποία έχει διασπαστεί το αρχικό μήνυμα, για την διεκπεραίωσή του, θα μπορεί να επιλέξει από τη λίστα κάποιο συγκεκριμένο task. Θα εμφανιστεί η αναλυτική περιγραφή των εργασιών που έχουν καταχωρηθεί για το συγκεκριμένο task. Επίσης μπορεί να επιλέξει μια από τις παρακάτω ενέργειες:

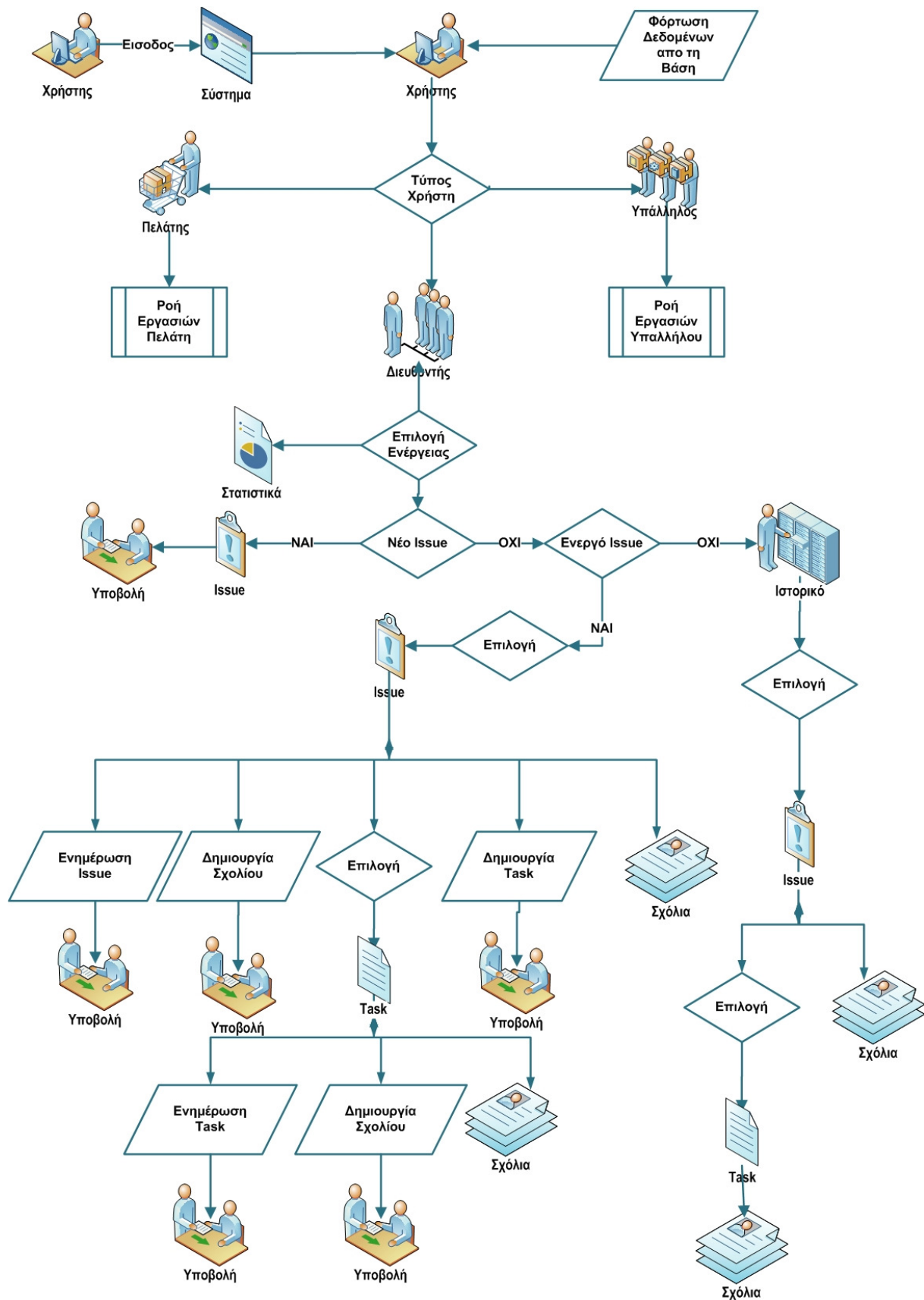
- Να ενημερώσει το task με ενέργειες που έχουν γίνει.
- Να δημιουργήσει νέο σχόλιο (για κάποια νέα πληροφορία που πιθανώς υπάρχει ή για κάποια ερώτηση που θέλει να κάνει) και στη συνέχεια να το καταχωρήσει.
- Να προβάλλει τα σχόλια που ήδη υπάρχουν.

Όλες οι ενέργειες από αυτή την ομάδα χρηστών, γίνονται από το front end της ιστοσελίδας, όπου έχουν δημιουργηθεί links για την επικοινωνία του υπαλλήλου με το σύστημα.

5.5.3 Για τον Διευθυντή

Ο διευθυντής (manager) μπορεί να εκμεταλλευτεί τη λειτουργικότητα του CIMaS τόσο από το front όσο και από το back end. Αφού θα κάνει εισαγωγή στη σελίδα (στην περίπτωση αυτή αναλύουμε τη ροή από το διαχειριστικό μέρος, αφού από εκεί μπορεί να κάνει κάποια παραπάνω πράγματα απ' ότι από το front end), θα επιλέξει την προβολή του CIMaS component, και από τον πίνακα εργασιών θα μπορεί να κάνει τις παρακάτω ενέργειες:

- Να ελέγξει τα στατιστικά που έχουν συγκεντρωθεί βάσει κριτηρίων που θα επιλέξει.
- Να δημιουργήσει ένα νέο issue για κάποιο πελάτη και στη συνέχεια να ενημερώσει το σύστημα για αυτό κάνοντας υποβολή.
- Να δημιουργήσει ένα νέο χρήστη ή να επεξεργαστεί τις πληροφορίες που υπάρχουν σχετικά με κάποιο χρήστη.
- Να επεξεργαστεί υπάρχον issue.
- Να δει ιστορικό



Διάγραμμα 3: Ροή Εργασιών Διευθυντή

Για να επεξεργαστεί κάποιο issue που ήδη υπάρχει, θα πρέπει πρώτα να το επιλέξει από τη λίστα που υπάρχουν αυτά, και στη συνέχεια να το επεξεργαστεί. Έπειτα πρέπει να το υποβάλλει, αποθηκεύοντάς το στους πίνακες της βάσης δεδομένων. Στην ίδια περίπτωση (δηλαδή σε κάποιο ήδη υπάρχον issue), ο διευθυντής, μπορεί να κάνει και τα εξής:

- Δημιουργία σχόλιου

- Επεξεργασία υπάρχοντος σχόλιου
- Προβολή σχόλιων που ήδη υπάρχουν
- Δημιουργία νέου task
- Επιλογή προβολής των υπαρχόντων tasks και την επεξεργασία τους.

Σε όλες τις περιπτώσεις πρέπει μετά την ολοκλήρωση της επεξεργασίας των πληροφοριών, να γίνει υποβολή, για να αποθηκευτούν οι πληροφορίες. Στην περίπτωση που αναφέρεται σε επεξεργασία υπάρχοντος task, η ροή των εργασιών ακολουθεί τα παρακάτω βήματα:

- Επιλέγεται από τη λίστα των tasks το επιθυμητό
- Επιλέγεται η επόμενη ενέργεια. Δηλαδή, χρειάζεται να προστεθεί κάποιο σχόλιο; Το δημιουργεί και στη συνέχεια το υποβάλλει. Χρειάζεται να προβληθούν τα σχόλια που ήδη υπάρχουν; Επιλέγει προβολή σχολίων και αυτά εμφανίζονται. Χρειάζεται να γίνει επεξεργασία στις πληροφορίες που υπάρχουν ήδη στο task; Επιλέγει επεξεργασία task, γίνεται αλλαγή στις επιθυμητές πληροφορίες και στην συνέχεια αποθηκεύει.

Όσα έχουν αναφερθεί περιγραφικά σε αυτή την υποενότητα, εμφανίζονται και γραφικά στο Διάγραμμα 3.

5.6 Class Diagram

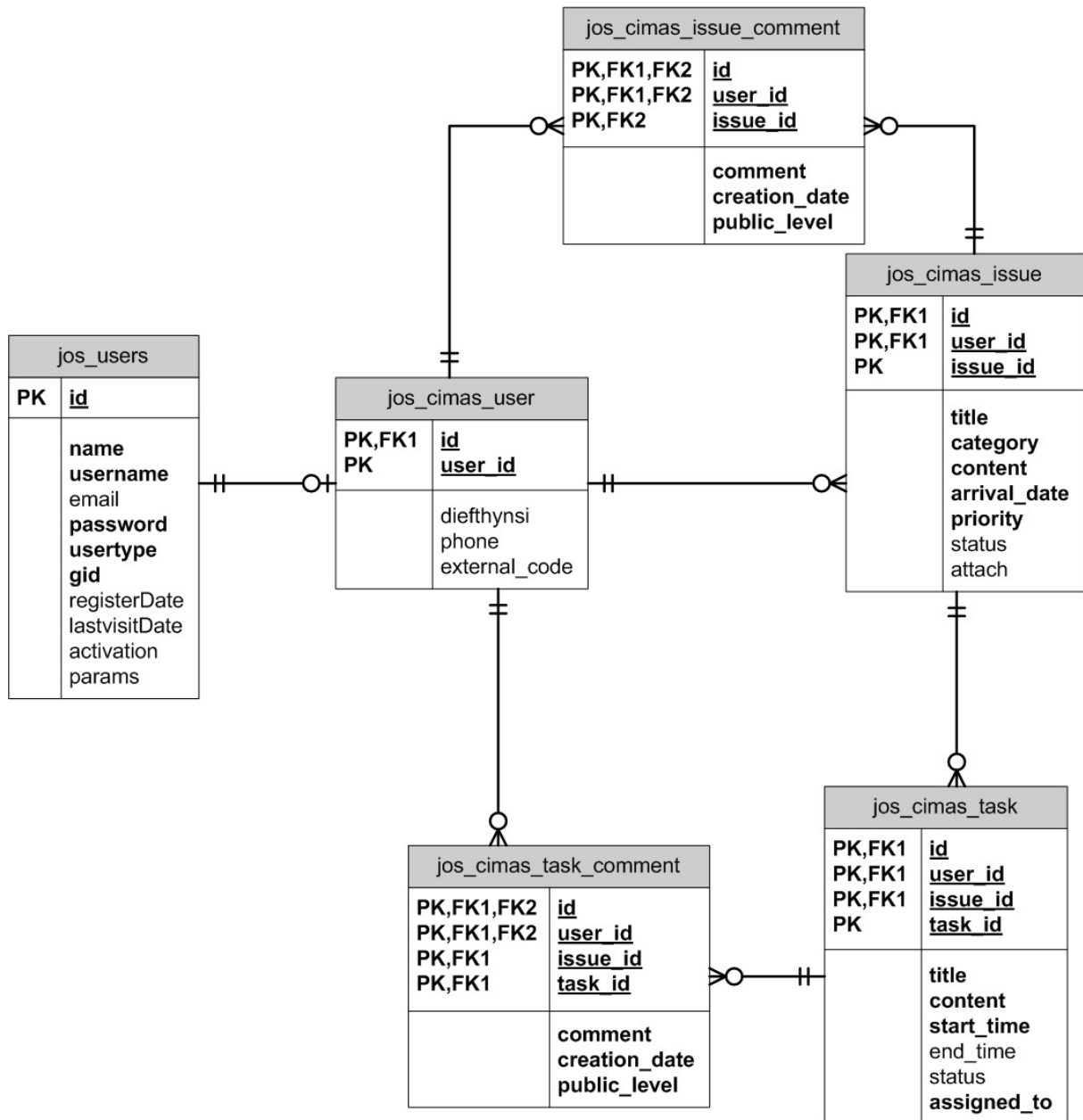
Στην ενότητα αυτή, παρουσιάζεται το διάγραμμα της βάσης δεδομένων που θα χρησιμοποιηθεί για την εφαρμογή CIMaS. Η αποσαφήνιση της βάσης των δεδομένων που θα περιέχει αυτή, είναι πολύ σημαντική, καθώς καθορίζεται η αποθήκευση των δεδομένων και ο τρόπος με τον οποίο, μέσω ερωτημάτων, μπορούν να ανακτηθούν δεδομένα από τη βάση αυτή.

Ποιο συγκεκριμένα στο **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**, βλέπουμε ότι η βάση που χρειάζεται για την υλοποίηση του CIMaS, αποτελείται από πέντε πίνακες, το jos_cimas_issue_comment, το jos_cimas_task_comment, το jos_cimas_issue, το jos_cimas_task, το jos_cimas_user και τον jos_users.

Το πρόθεμα jos_ που υπάρχει σε όλους του πίνακες, είναι το πρόθεμα που τοποθετείται στους πίνακες που χρησιμοποιούνται από το Joomla!. Επίσης, το δεύτερο συνθετικό που υπάρχει στους περισσότερους από αυτούς, σημαίνει ότι είναι πίνακας που έχει εγκατασταθεί για την εφαρμογή CIMaS.

Ο πίνακας jos_users, δημιουργείται κατά την αρχική εγκατάσταση του Joomla! και εκεί αποθηκεύονται όλοι οι εγγεγραμμένοι χρήστες του συστήματος. Οι πίνακες που έχουν το πρόθεμα jos_cimas_, θα δημιουργηθούν όταν θα εγκατασταθεί το component CIMaS. Η λειτουργικότητα αυτή παρέχεται, δημιουργώντας ένα αρχείο install.mysql.sql και τοποθετώντας μέσα σε αυτό για τον κάθε πίνακα τις εξής γραμμές εντολών:

```
CREATE TABLE `#__cimas_issue_comment` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `comment` text NOT NULL default '',
  `creation_date` datetime NOT NULL default '0000-00-00 00:00:00',
  `user_id` int(11) NOT NULL default 0,
  `issue_id` int(11) NOT NULL default 0,
  `public_level` varchar(20) NOT NULL default '',
  PRIMARY KEY (`id`)
) ENGINE= MyISAM AUTO_INCREMENT=0 DEFAULT CHARSET=utf8;
```



Διάγραμμα 4: Σχήμα Βάσης Δεδομένων

Οι εντολές αυτές πιο αναλυτικά, λένε στη μηχανή MyISAM, να δημιουργήσει ένα πίνακα, που θα είναι ο επόμενος από τον τελευταίο που είχε δημιουργηθεί στη βάση δεδομένων, με κωδικοποίηση χαρακτήρων utf8.

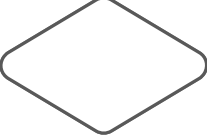







Η εντολή CREATE TABLE `#__cimas_issue_comment` ενημερώνει για τη δημιουργία ενός πίνακα με το όνομα jos_cimas_issue_comment. Το #__ είναι το γενικό πρόθεμα, που όταν το component εγκατασταθεί στο Joomla! μετατρέπεται στο πρόθεμα jos_. Ο πίνακας αυτός θα περιέχει τα πεδία που υπάρχουν ανάμεσα στην παρένθεση που ανοίγει στην πρώτη γραμμή, και κλείνει στην τελευταία. Τέλος μέσα στις παρενθέσεις αυτές, υπάρχει μια αναλυτική περιγραφή για τα χαρακτηριστικά του κάθε πεδίου, όπως όνομα, τύπο δεδομένων που θα υπάρχουν σε αυτό, το μέγεθος των δεδομένων που μπορούν να αποθηκευτούν, το αν η έλλειψη απόδοσης τιμής είναι αποδεκτή και τέλος η προκαθορισμένη του τιμή.

Με τον τρόπο αυτό, στο ίδιο αρχείο καταχωρούνται όλοι οι πίνακες, ώστε στην εγκατάσταση του component, να δημιουργηθούν χωρίς κάποιο πρόβλημα.

5.7 Data Flow Diagrams

Στην ενότητα αυτή παρουσιάζεται η ροή των δεδομένων (πως γίνεται η διακίνησή τους) στο CIMaS.

Για την καλύτερη κατανόηση των διαγραμμάτων που ακολουθούν, στον Πίνακα 1 παρουσιάζονται τα σχήματα που χρησιμοποιούνται και η ερμηνεία τους.

Σχήμα	Ερμηνεία
	Απόφαση
	Αποθηκευμένα Δεδομένα
	Δεδομένα
	Ακολουθία επεξηγούμενη σε άλλο διάγραμμα
	Επεξεργασία δεδομένων βάσης (προσθήκη – τροποποίηση – διαγραφή)
	Παρουσίαση
	Τερματικό
	Σύνδεση

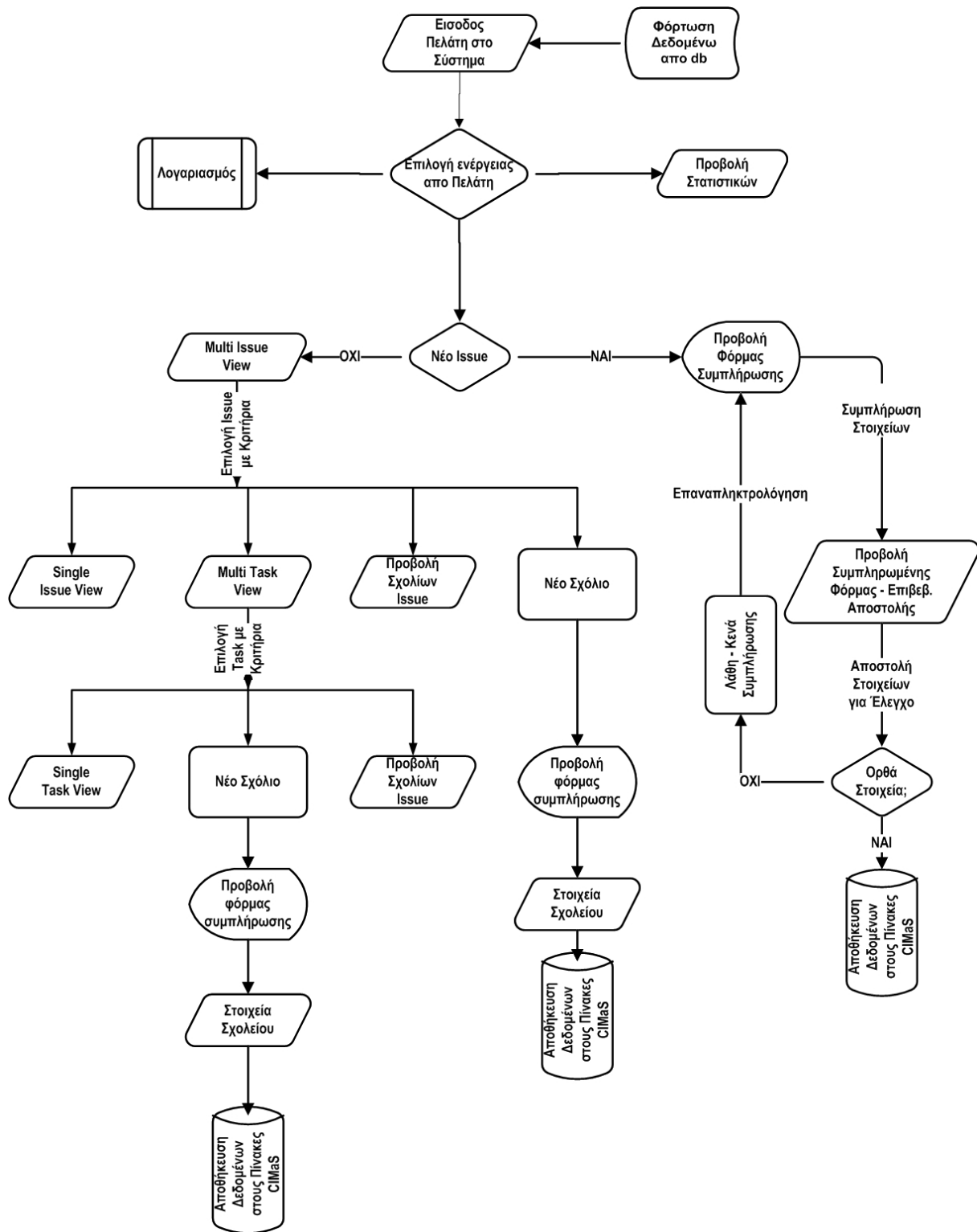
Πίνακας 3: Ερμηνεία Χρησιμοποιηθέντων Σχημάτων των Data Flow Diagrams

5.7.1 Data Flow Diagram για τον Πελάτη

Στο Διάγραμμα 5, παρουσιάζεται η ροή που θα έχουν οι πληροφορίες για τη διακίνησή τους από τη σκοπιά του πελάτη.

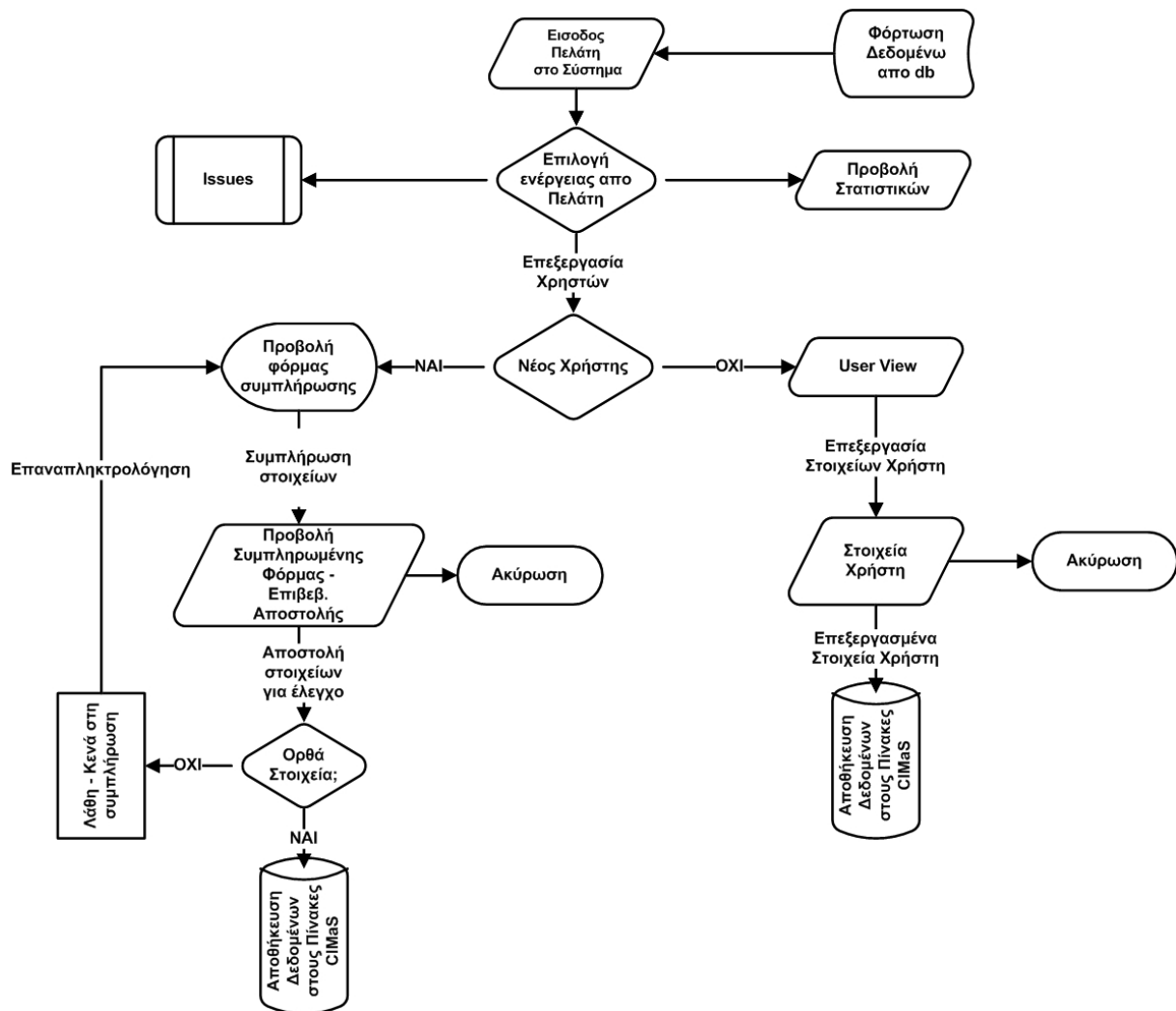
Πιο συγκεκριμένα, με την είσοδο του χρήστη στο σύστημα, φορτώνονται οι προσωπικές του ρυθμίσεις και τα δικαιώματα που έχουν καταχωρηθεί για αυτόν, στη βάση δεδομένων.

Στη συνέχεια, επιλέγει τι θα ήθελε να κάνει σε σχέση με το CIMaS. Μπορεί να επιλέξει να προβληθούν πληροφορίες από στατιστικά δεδομένα, να επεξεργαστεί τα στοιχεία που έχει δηλώσει, να ενημερωθεί για ένα υπάρχον ζήτημα ή να δημιουργήσει κάποιο καινούριο ζήτημα. Σε κάθε περίπτωση, όταν ο πελάτης ζητάει πληροφορίες, αυτές ανακτώνται από τη βάση δεδομένων, ενώ όταν θέλει να υποβάλλει πληροφορίες, αυτές αποθηκεύονται στους προκαθορισμένους πίνακες (στα σημεία που έχουν δηλωθεί μέσω του κώδικα).



Διάγραμμα 5: Ροή Δεδομένων Πελάτη

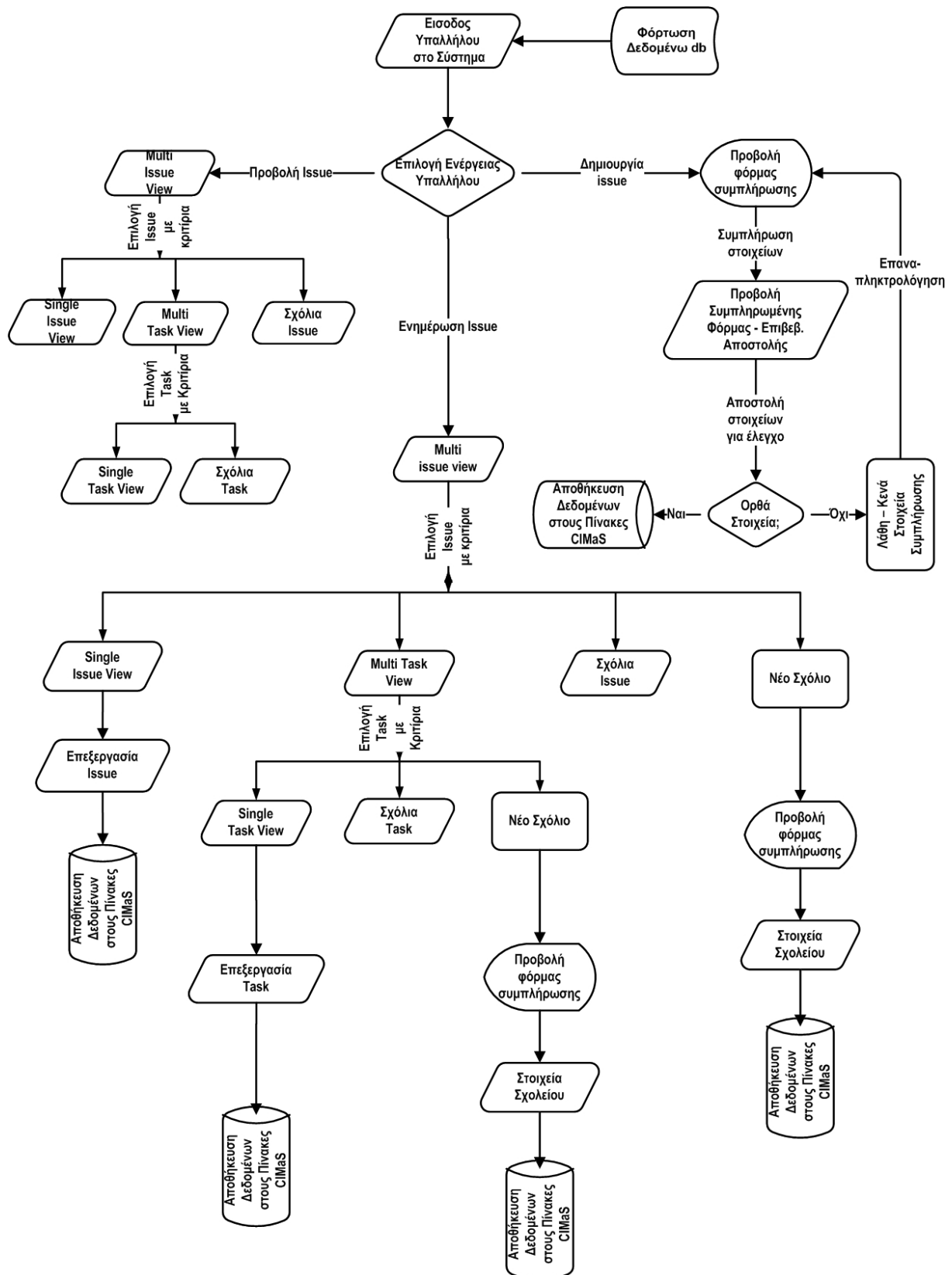
Σύμφωνα λοιπόν με τα παραπάνω, στο Διάγραμμα 5 ο πελάτης ζητάει να του εμφανιστούν πληροφορίες από τη βάση, μέσω των επιλογών multi issue view (πολλαπλή προβολή για issues), single issue view (προβολή ενός μόνο issue), multi task view (πολλαπλή προβολή των tasks), single task view (προβολή ενός μόνο task), προβολή σχολίων issue ή task και προβολή δηλωμένων στοιχείων. Επίσης στέλνει δεδομένα προς τη βάση, μέσω των επιλογών δημιουργίας νέου issue, μέσω δημιουργίας σχολίων τόσο σε issue όσο και σε task, αλλά και μέσω δημιουργίας λογαριασμού – προφίλ χρήστη (Διάγραμμα 6).



Διάγραμμα 6: Ροή Δεδομένων για Δημιουργία Λογαριασμού Πελάτη

5.7.2 Data Flow Diagram για τον Υπάλληλο

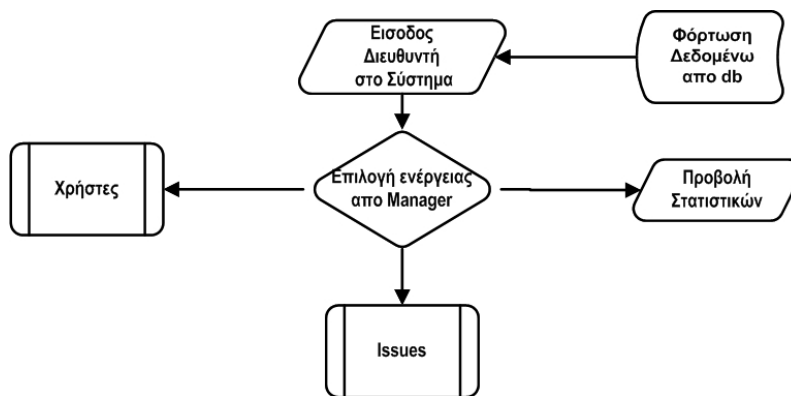
Στο Διάγραμμα 7, φαίνεται διαγραμματικά η διακίνηση των δεδομένων για το χρήστη υπάλληλο. Ο υπάλληλος με τις επιλογές για δημιουργία νέου issue, ενημέρωση εργασιών σε issue, ενημέρωση εργασιών σε task, δημιουργία σχόλιου τόσο σε issue όσο και σε task, μεταφέρει δεδομένα στη βάση. Αντίστοιχα, όταν εκτελεί κάποια από τις επιλογές, είσοδος χρήστη στο σύστημα, προβολή υπαρχόντων issues (multi issue view), προβολή issue (single issue view), προβολή υπαρχόντων tasks (multi task view), προβολή task (single task view), προβολή σχόλιων τόσο για tasks όσο και για issues, ζητάει από τη βάση να του μεταφέρει δεδομένα, για να ενημερωθεί και να επεξεργαστεί όποια από αυτά απαιτούν ενέργεια.



Διάγραμμα 7: Ροή Δεδομένων για Υπάλληλο

5.7.3 Data Flow Diagrams για τον Διευθυντή

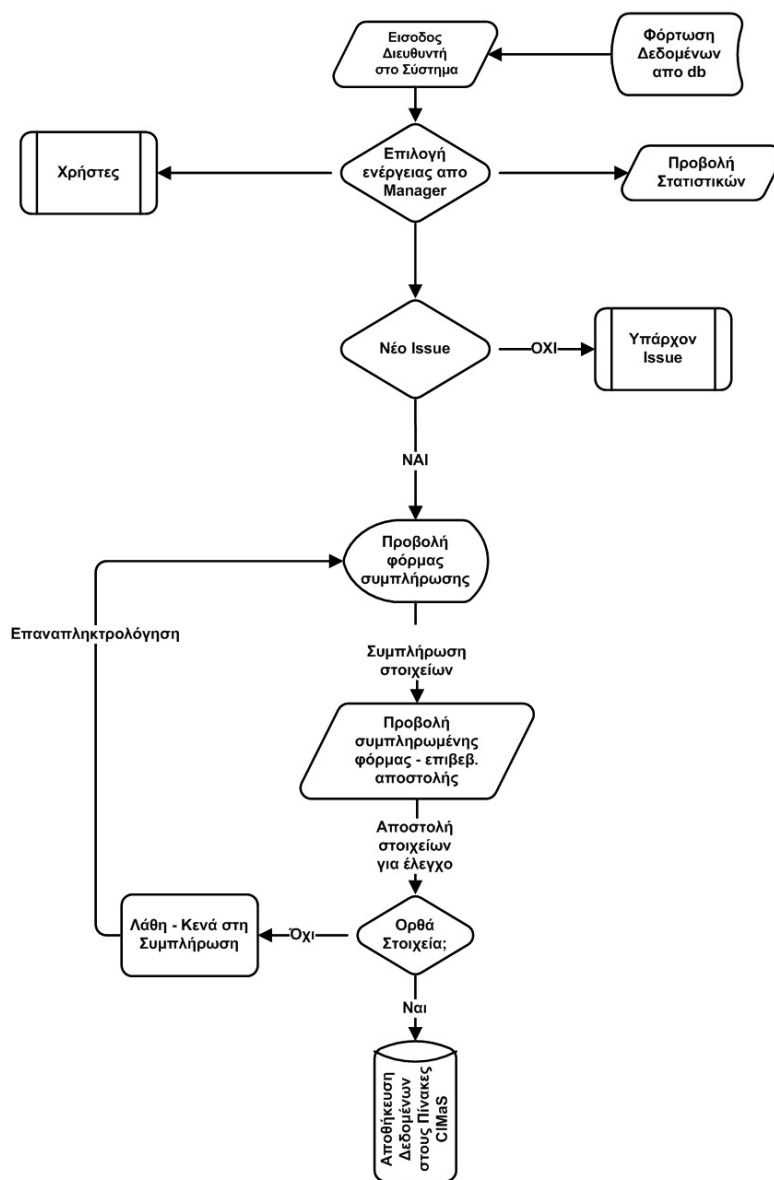
Στο παρακάτω Διάγραμμα 8, φαίνονται οι επιλογές για το διευθυντή (manager) μέσω των οποίων, στη συνέχεια θα αναπτυχθούν τα διαγράμματα ροής των δεδομένων για τις εργασίες που μπορεί να εκτελέσει.



Διάγραμμα 8: Αρχικό Διάγραμμα Ροής Δεδομένων Διευθυντή

Οι ενέργειες μέσω των οποίων κάνει διακίνηση δεδομένων, είναι η επιλογή προβολής στατιστικών στοιχείων βάσει κριτηρίων, η επιλογή επεξεργασίας κάποιου χρήστη ή, η επιλογή ενεργειών στα issues.

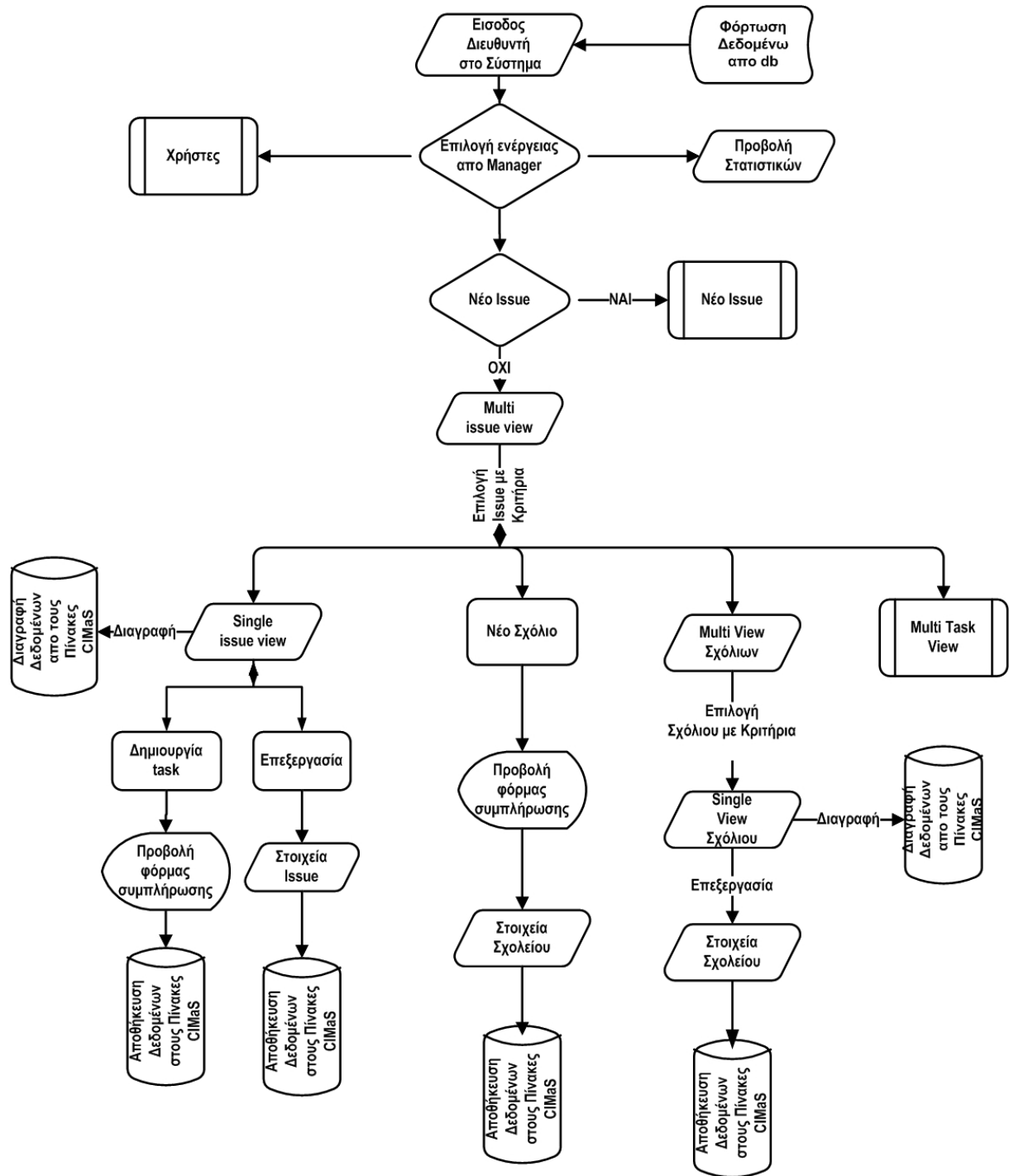
5.7.3.1 Νέο Issue



Διάγραμμα 9: Ροή Δεδομένων Διευθυντή - Δημιουργία Issue

Στο Διάγραμμα 9, παρουσιάζεται η διακίνηση των πληροφοριών από το χρήστη manager σχετικά με το CIMaS για τη δημιουργία νέου ζητήματος. Συμπληρώνοντας τη φόρμα δημιουργίας νέου Issue, μέσω του submit button, τα δεδομένα αποστέλλονται για αποθήκευση στη βάση δεδομένων.

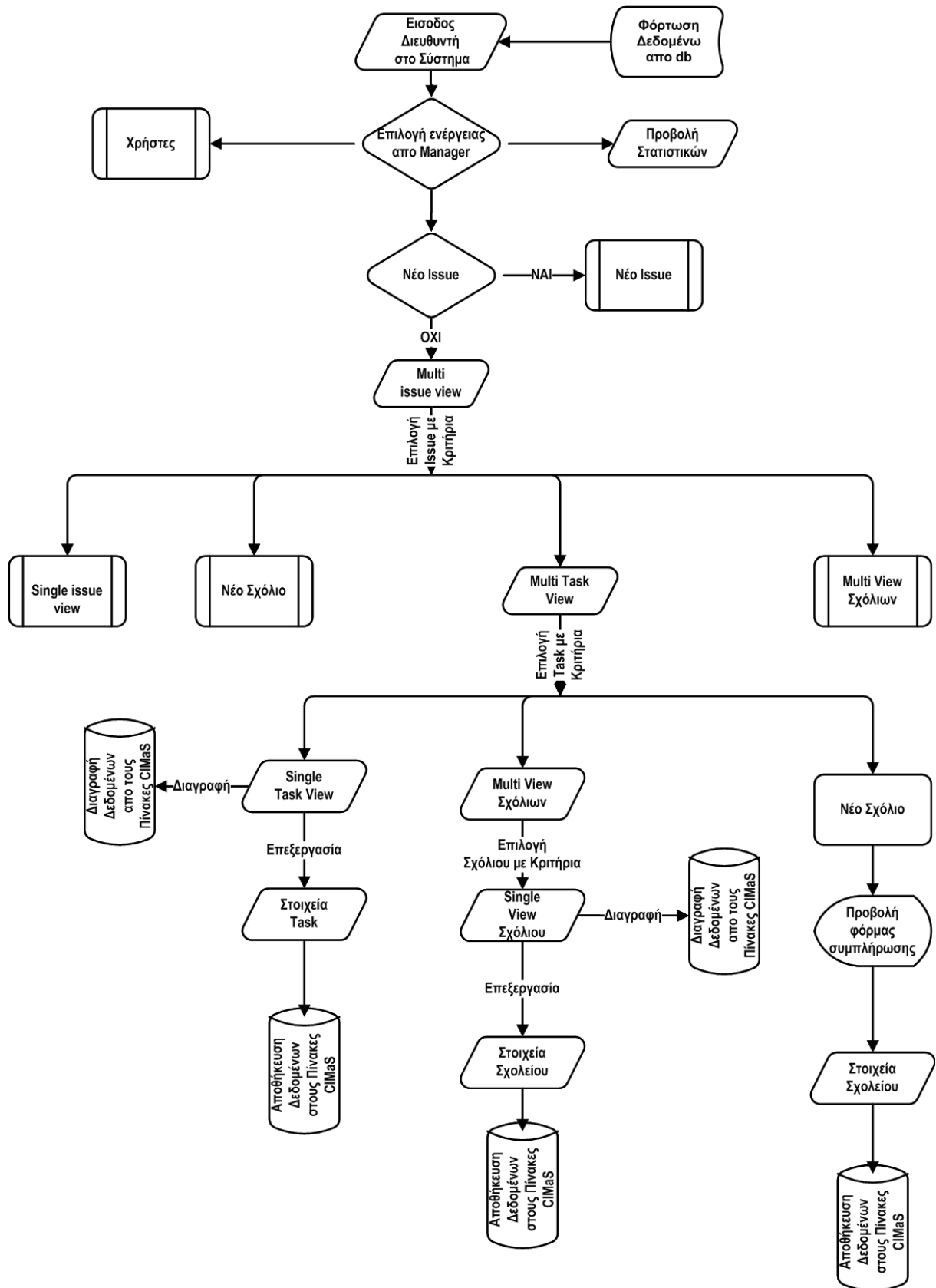
5.7.3.2 Υπάρχον Issue



Διάγραμμα 10: Ροή Δεδομένων Διευθυντή για Υπάρχον Issue

Στο Διάγραμμα 10 και Διάγραμμα 11 παρουσιάζεται η ανταλλαγή των δεδομένων ανάμεσα στη βάση και το CIMaS για το χρήστη διευθυντή, στην περίπτωση που ενημερώνεται – επεξεργάζεται κάποιο υπάρχον issue.

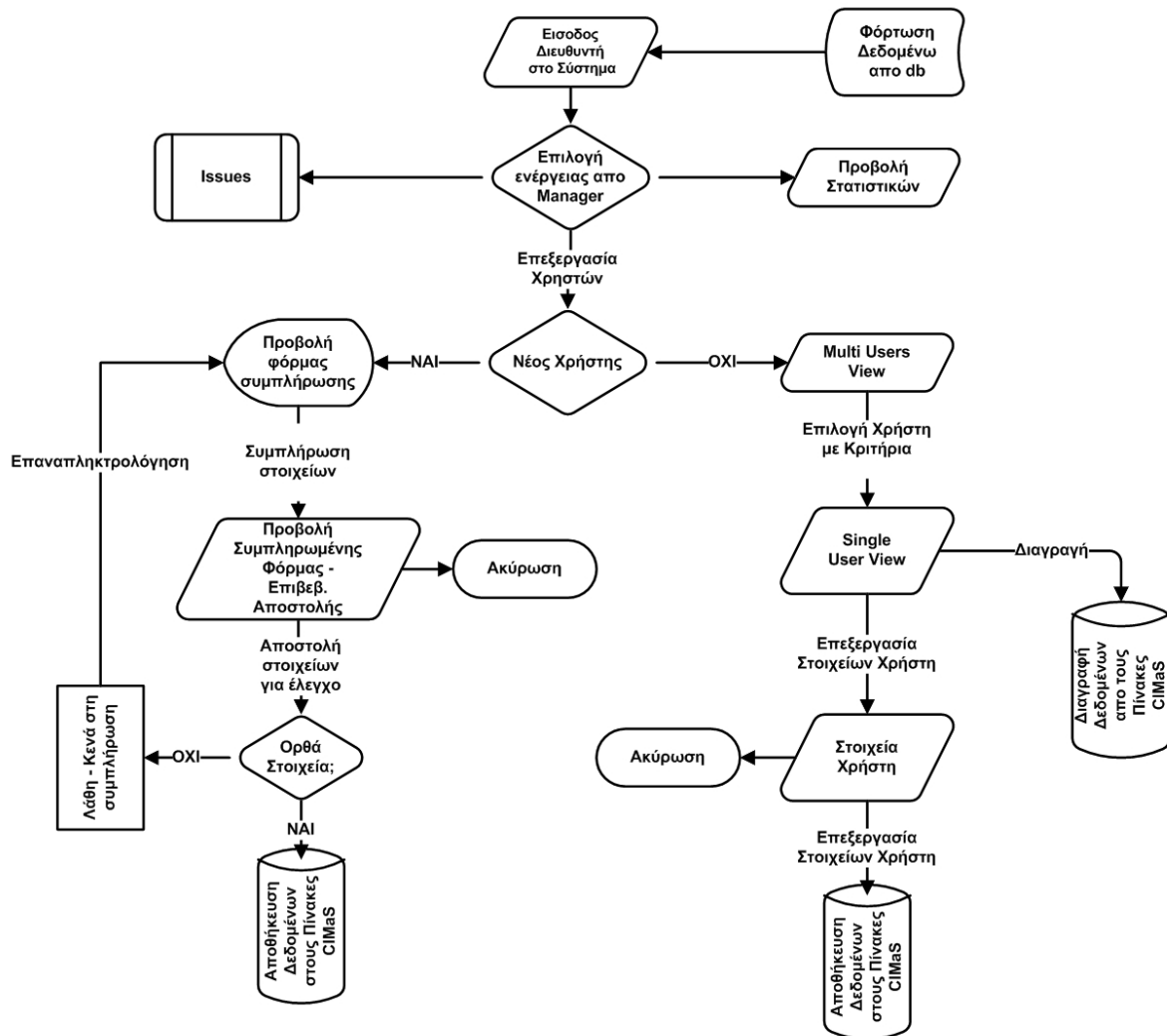
Ο διευθυντής, ενώ κάποιο issue υπάρχει, μπορεί να στέλνει στη βάση δεδομένα, όταν επεξεργάζεται tasks, issues, σχόλια για issues ή tasks και όταν δημιουργεί καινούριο task για κάποιο issue.



Διάγραμμα 11: Ροή Δεδομένων Διευθυντή για Tasks

Επίσης ενημερώνει τη βάση, όταν διαγράφει κάποιο issue, task ή σχόλιο.

5.7.3.3 Διαχείριση Χρηστών



Διάγραμμα 12: Ροή Δεδομένων Διευθυντή για Χρήστες

Στο τελευταίο Διάγραμμα 12, εμφανίζεται η ανταλλαγή των δεδομένων για το χρήστη διευθυντή σε περίπτωση που δημιουργεί νέο χρήστη ή που επεξεργάζεται δεδομένα κάποιου άλλου υπάρχοντος χρήστη.

Πιο συγκεκριμένα, στην περίπτωση δημιουργίας νέου χρήστη, συμπληρώνει μια φόρμα με τα στοιχεία του, και στη συνέχεια στέλνει τα δεδομένα για αποθήκευσή τους στη βάση. Σε περίπτωση επεξεργασίας στοιχείων κάποιου υπάρχοντος χρήστη, ζητάει από τη βάση να παραλάβει τα δεδομένα για το συγκεκριμένο χρήστη και μετά την επεξεργασία τους, τα στέλνει ξανά για αποθήκευση στη βάση.

5.8 Περίληψη Κεφαλαίου

Στην ενότητα αυτή, έγινε η καταγραφή των σεναρίων και η σχηματική απεικόνιση της ροής των εργασιών, της ροής των δεδομένων, της δομής της βάσης δεδομένων μας και του τρόπου ανταλλαγής δεδομένων χρηστών με το σύστημα. Η ανάλυση αυτή, είναι απαραίτητη για την αποσαφήνιση των διαδικασιών, έτσι ώστε να περάσουμε στο επόμενο βήμα, που είναι η υλοποίηση του CIMaS.

Στο επόμενο κεφάλαιο, θα παρουσιαστεί ο κώδικας που θα χρησιμοποιηθεί για την υλοποίηση της εφαρμογής, καθώς και οθόνες από την υλοποίηση.

ΚΕΦΑΛΑΙΟ 6

Υλοποίηση Συστήματος

6.1 Εισαγωγή

Μετά την μελέτη που έχει γίνει παραπάνω για το CIMaS, στο παρόν κεφάλαιο θα γίνει η υλοποίησή του. Στα πλαίσια της υλοποίησης αυτής, έχει δημιουργηθεί ένα component με τη χρήση γλώσσας προγραμματισμού php, που υποστηρίζει τη διαχείριση και παρακολούθηση ζητημάτων (του οποίου η ανάλυση γίνεται στα κεφάλαια 4-5). Επίσης το σύστημα θα επεκταθεί με κάποιες έτοιμες λειτουργικότητες ανοιχτού κώδικα, ώστε να διασφαλιστεί η ικανοποίηση του πελάτη στο μεγαλύτερο δυνατό βαθμό.

Στην ενότητα αυτή, θα ασχοληθούμε επιπλέον με το Model View Controller design, με το οποίο θα υλοποιηθεί το component CIMaS και θα παρουσιαστούν κομμάτια κώδικα και εικόνες από την τελική μορφή του συστήματος.

6.2 Επιλογή και Εγκατάσταση CMS

Στο Κεφάλαιο 2, όπου αναφέρονται τα συστήματα διαχείρισης περιεχομένου, αιτιολογήθηκε η απόφαση χρησιμοποίησης του Joomla! για την υλοποίηση του συστήματος, στηριζόμενοι στις επεκτατικές του δυνατότητες. Για τη χρήση όμως του εργαλείου αυτού (εγκατάστασή του), πρέπει να πληρούνται κάποιες προϋποθέσεις (απαιτήσεις) ώστε να διασφαλιστεί η ορθή και πλήρης λειτουργικότητά του. Πιο συγκεκριμένα, χρειάζονται:

PHP: έκδοση τουλάχιστον 4.3.10

Τι είναι όμως PHP; Η PHP (Hypertext Preprocessor) είναι μια ευρέως χρησιμοποιούμενη, γενικού σκοπού scripting γλώσσα προγραμματισμού, που αρχικά, σχεδιάστηκε για την ανάπτυξη Ιστού για την παραγωγή δυναμικών ιστοσελίδων. Για το σκοπό αυτό, ο PHP κώδικας, είναι ενσωματωμένος στο έγγραφο πηγαίου κώδικα HTML και μεταφράζεται από έναν web server μέσω μιας μονάδας επεξεργαστή PHP, η οποία παράγει το έγγραφο ως ιστοσελίδα. Ως γενικής χρήσης γλώσσα προγραμματισμού, ο κώδικάς της επεξεργάζεται από μια εφαρμογή διερμηνέα, με χρήση command-lines, που εκτελούν τις επιθυμητές εργασίες στο σύστημα και παράγει τυποποιημένα αντίγραφα του προγράμματος, στο κανάλι εξόδου της. Μπορεί επίσης να λειτουργήσει ως γραφική εφαρμογή. Η PHP είναι διαθέσιμη ως επεξεργαστής για τους πιο σύγχρονους web servers και ως αυτόνομος διερμηνέας για τα περισσότερα λειτουργικά συστήματα και πλατφόρμες υπολογιστών¹⁶.

¹⁶ en.wikipedia.org/wiki/Php

MySQL: έκδοση από 3.23.x και πάνω

Γιατί χρειάζεται η MySQL; Η MySQL, είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων που τρέχει ως server και επιτρέπει την πρόσβαση πολλών χρηστών σε βάσεις δεδομένων. Το project MySQL, διαθέτει τον πηγαίο κώδικά του, υπό τους όρους του GNU (General Public Licence), καθώς και υπό τους όρους διαφόρων ιδιωτικών συμφωνητικών. Έργα ελεύθερου λογισμικού που απαιτούν ένα πλήρως εξοπλισμένο σύστημα διαχείρισης της βάσης των δεδομένων τους, κάνουν χρήση της MySQL¹⁷.

Apache: έκδοση 1.3 και πάνω

Τι βοήθεια μπορεί να προσφέρει ο Apache; Ο Apache HTTP server είναι ένα λογισμικό για τη δημιουργία web servers και έχει διαδραματίσει πολύ σημαντικό ρόλο στην ανάπτυξη του παγκόσμιου ιστού. Χρησιμοποιείται κυρίως για την εξυπηρέτηση τόσο στατικού περιεχομένου, όσο και δυναμικών ιστοσελίδων. Βοηθάει στη διανομή του περιεχομένου με ασφαλή και αξιόπιστο τρόπο. Ένα παράδειγμα είναι η κοινή χρήση αρχείων από έναν προσωπικό υπολογιστή μέσω του διαδικτύου.

Ένα εργαλείο που περιέχει τα παραπάνω και μπορεί να εγκατασταθεί σε λειτουργικό σύστημα Windows, είναι ο WampServer. Ο WampServer, είναι ένα web περιβάλλον ανάπτυξης για Windows. Παρέχει τη δυνατότητα να δημιουργηθούν web εφαρμογές χρησιμοποιώντας Apache, PHP και βάση δεδομένων MySQL. Επίσης παρέχει και το εργαλείο phpMyAdmin, για την ευκολότερη διαχείριση των MySQL data bases. Η έκδοση WampServer 2.0i, περιλαμβάνει Apache 2.2.11, MySQL 5.1.36 και PHP 5.3.0. Το εργαλείο αυτό είναι το καταλληλότερο, καθώς θα μπορεί να στηρίξει το σύστημα που θα υλοποιηθεί, ακόμα και σε πιθανές μελλοντικές αναβαθμίσεις¹⁸.

6.2.1 Εγκατάσταση WampServer

Για να γίνει η εγκατάσταση του WampServer, πρέπει να ακολουθηθούν τα εξής βήματα:

Βήμα 1. Μεταφόρτωση αρχείου για εγκατάσταση από τη σελίδα www.wampserver.com.

Βήμα 2. Μετά το τέλος της μεταφόρτωσης, πρέπει να ανοιχτεί το αρχείο για να ξεκινήσει η εγκατάσταση του server.



Εικόνα 22: Wizard Παράθυρο Διαλόγου για Εγκατάσταση WampServer 2

¹⁷ en.wikipedia.org/wiki/MySQL

¹⁸ en.wikipedia.org/wiki/Apache_HTTP_Server

Βήμα 3. Στη συνέχεια, από το wizard παράθυρο που εμφανίζεται στην Εικόνα 22, επιλέγεται η εντολή next.

Βήμα 4. Στο επόμενο παράθυρο διαλόγου που εμφανίζεται, αναγράφονται οι όροι της GNU General Public License, που επιτρέπει τη χρήση του λογισμικού αυτού. Μετά την ανάγνωση της άδειας, επιλέγεται το radiobutton: I accept the agreement και στη συνέχεια πατείται το κουμπί next.

Βήμα 5. Στο παράθυρο που θα εμφανιστεί μετά, γίνεται επιλογή του επιθυμητού σημείου εγκατάστασης όλων των αρχείων. Το προκαθορισμένο σημείο είναι c:\wamp. Μπορεί να δηλωθεί κάποιο άλλο μέρος αποθήκευσης, πατώντας το κουμπί browse και επιλέγοντας το μονοπάτι που θέλουμε να τοποθετηθεί.

Βήμα 6. Στο επόμενο παράθυρο επιλέγονται τα σημεία στα οποία είναι επιθυμητό, να δημιουργηθούν επιπλέον συντομεύσεις. Η επιλογή γίνεται τσεκάροντας τα σημεία ενδιαφέροντος. Μετά από αυτό, η διαδικασία συνεχίζεται πατώντας το κουμπί next.


Βήμα 7. Επόμενο παράθυρο που εμφανίζεται, είναι ένας container που ενημερώνει για τις επιλογές που έγιναν στο 5^ο βήμα (επιλογή σημείου εγκατάστασης φακέλου αρχείων) και στο 6^ο βήμα (επιλογή σημείων δημιουργίας συντομεύσεων). Πατώντας την επιλογή Install, ξεκινάει η εγκατάσταση του WampServer.

Βήμα 8. Εμφανίζεται ένα παράθυρο, που ενημερώνει για την πορεία εγκατάστασης του προγράμματος.

Βήμα 9. Το επόμενο παράθυρο, εμφανίζεται μόλις ολοκληρωθεί η εγκατάσταση και ρωτάει αν είναι επιθυμητή η ρύθμιση του FireFox σαν προκαθορισμένο browser για το WampServer. Αν αυτό είναι επιθυμητό, τότε επιλέγεται το κουμπί Yes, διαφορετικά, πατώντας το κουμπί No, ρυθμίζεται ως προεπιλεγμένος browser ο Internet Explorer.

Βήμα 10. Το επόμενο παράθυρο επιτρέπει τη ρύθμιση του server για προώθηση email που δημιουργεί η php στο κατάλληλο server, και εισάγεται ένας λογαριασμός email για να πηγαίνουν στο σωστό άτομο.

Βήμα 11. Τέλος εμφανίζεται ένα παράθυρο που ενημερώνει για την ολοκλήρωση της εγκατάστασης. Σε αυτό το παράθυρο διαλόγου, επιλέγεται το κουμπί Finish.

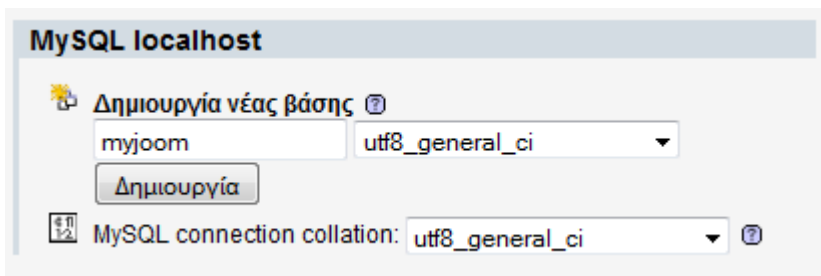
Με την ολοκλήρωση της εγκατάστασης, έχει δημιουργηθεί στην κάτω δεξιά μπάρα του υπολογιστή το εικονίδιο  που ενημερώνει ότι ο server είναι σε λειτουργία.

6.2.2 Εγκατάσταση Joomla!

Για την εγκατάσταση του Joomla! θα πρέπει πρώτα να έχει δημιουργηθεί μία βάση δεδομένων MySQL στην οποία θα αποθηκεύονται τα δεδομένα για το CMS αυτό, και ένας χρήστης με πλήρη δικαιώματα για τη βάση αυτή.

6.2.2.1 Δημιουργία Βάσης Δεδομένων

Για τη δημιουργία της βάσης δεδομένων, θα χρησιμοποιήσουμε το εργαλείο phpMyAdmin που έχει εγκατασταθεί προηγουμένως μέσα στο WampServer.



Εικόνα 23: Δημιουργία Βάσης Δεδομένων στην MySQL

Βήμα 1. Τρέξιμο του WampServer

Βήμα 2: Επιλογή phpMyAdmin

Βήμα 3: Στο σημείο που φαίνεται «Δημιουργία νέας βάσης», συμπληρώνεται το όνομα της βάσης που είναι επιθυμητό για τη δημιουργία της, και επιλέγεται για Collation (από το list box) το utf8_general_ci.

Βήμα 4. Μετά τη συμπλήρωση των παραπάνω στοιχείων, με το πάτημα του κουμπιού «Δημιουργία», δημιουργείται η MySQL βάση δεδομένων.

Για την εγκατάσταση του Joomla! στο σύστημα διαχείρισης ζητημάτων, δημιουργήθηκε η βάση με το όνομα myjoomla.

6.2.2.2 Προσθήκη Νέου Χρήστη

Βήμα 1. Για να δημιουργηθεί ένας νέος χρήστης, με δικαιώματα στη βάση που δημιουργήθηκε στην 6.2.2.1 υποενότητα, επιλέγονται τα εξής: Προνόμια→Προσθήκη νέου Χρήστη. Στις πληροφορίες σύνδεσης πρέπει να δοθούν οι ακόλουθες πληροφορίες:

Διακομιστής: localhost

Βάσεις Δεδομένων SQL Κατάσταση Μεταβλητές Σειτ χαρακτήρων Engines Προνόμια

Binary log Διεργασίες Εξαγωγή Import

Επεξεργασία Προνομίων: Χρήστης 'rena'@'localhost'

Γενικά προνόμια (Επιλογή όλων / Απεπιλογή όλων)

Σημείωση: Τα ονόματα προνομίων της MySQL εκφράζονται στα Αγγλικά

Δεδομένα	Δομή	Διαχείριση
<input checked="" type="checkbox"/> SELECT	<input checked="" type="checkbox"/> CREATE	<input checked="" type="checkbox"/> GRANT
<input checked="" type="checkbox"/> INSERT	<input checked="" type="checkbox"/> ALTER	<input checked="" type="checkbox"/> SUPER
<input checked="" type="checkbox"/> UPDATE	<input checked="" type="checkbox"/> INDEX	<input checked="" type="checkbox"/> PROCESS
<input checked="" type="checkbox"/> DELETE	<input checked="" type="checkbox"/> DROP	<input checked="" type="checkbox"/> RELOAD
<input checked="" type="checkbox"/> FILE	<input checked="" type="checkbox"/> CREATE TEMPORARY TABLES	<input checked="" type="checkbox"/> SHUTDOWN
	<input checked="" type="checkbox"/> SHOW VIEW	<input checked="" type="checkbox"/> SHOW DATABASES
	<input checked="" type="checkbox"/> CREATE ROUTINE	<input checked="" type="checkbox"/> LOCK TABLES
	<input checked="" type="checkbox"/> ALTER ROUTINE	<input checked="" type="checkbox"/> REFERENCES
	<input checked="" type="checkbox"/> EXECUTE	<input checked="" type="checkbox"/> REPLICATION CLIENT
	<input checked="" type="checkbox"/> CREATE VIEW	<input checked="" type="checkbox"/> REPLICATION SLAVE
	<input checked="" type="checkbox"/> EVENT	<input checked="" type="checkbox"/> CREATE USER
	<input checked="" type="checkbox"/> TRIGGER	

Όρια πόρων

Σημείωση: Αν ορίσετε αυτές τις επιλογές σε 0 (μηδέν) αφαιρείτε ο περιορισμός.

MAX QUERIES PER HOUR 0

MAX UPDATES PER HOUR 0

MAX CONNECTIONS PER HOUR 0

MAX USER_CONNECTIONS 0

Εκτέλεση

Εικόνα 24: MySQL Απόδοση Δικαιωμάτων Χρήστη

1. Όνομα Χρήστη για τη βάση (με λατινικούς χαρακτήρες)
 2. Σύστημα: localhost (καθώς η εγκατάσταση θα γίνει σε τοπικό server)
 3. Κωδικός πρόσβασης.
- Τέλος εκτελείται η επιλογή «εκτέλεση», που βρίσκεται στο τέλος της σελίδας.

Βήμα 2. Σε αυτό το βήμα, ο χρήστης πρέπει να συνδεθεί με τη βάση. Από την αρχική σελίδα του phpMyAdmin, γίνονται οι εξής επιλογές: Προνόμια→Περίληψη Χρηστών→αναζήτηση χρήστη που δημιουργήθηκε→ επιλογή εικονιδίου επεξεργασίας χρήστη.

Βήμα 3: Στην οθόνη που εμφανίζεται, επιλέγεται το πλαίσιο «Προνόμια Βάσης Δεδομένων» και στο πλαίσιο αυτό, επιλέγεται το όνομα της βάσης με την οποία πρέπει να συνδεθεί ο χρήστης. Στη

συνέχεια, πρέπει να γίνει ανανέωση στον browser. Μετά την ανανέωση, εμφανίζεται το πλαίσιο επιλογής προνομίων, όπου γίνεται «Επιλογή Όλων» και στη συνέχεια με την εντολή «εκτέλεση», εμφανίζεται μήνυμα από το πρόγραμμα ότι η αλλαγή πραγματοποιήθηκε.

Στο σημείο αυτό, έχει δημιουργηθεί μια βάση δεδομένων με ένα χρήστη που έχει πλήρη δικαιώματα σε αυτή.

6.2.2.3 Εγκατάσταση Joomla!

Αφού έχουν γίνει οι παραπάνω ενέργειες, σειρά έχει η εγκατάσταση του Joomla!.

Βήμα 1. Μεταφόρτωση της τελευταίας έκδοσης του Joomla από τη σελίδα joomla.org.

Βήμα 2. Πρέπει να αποσυμπεστεί το αρχείο που ελήφθη, σε ένα φάκελο με όνομα π.χ. Joomla. Στη συνέχεια ο φάκελος αυτός τοποθετείται στο φάκελο www του Wamp.

Βήμα 3. Σε αυτό το βήμα πρέπει να ανοιχτεί ένα browser, και δίνεται η διεύθυνση <http://localhost/Joomla>. Η σελίδα εγκατάστασης του Joomla! θα εμφανιστεί στην οθόνη.

Βήμα 4. Επιλέγεται η γλώσσα εγκατάστασης και συνεχίζεται η διαδικασία με την εκτέλεση της εντολής «επόμενο».

Προληπτικός Έλεγχος

Προληπτικός έλεγχος για Joomla! 1.5.22 Stable [senu takaa ama woi] 04-November-2010 18:00 GMT:

Εάν κάποιο από τα στοιχεία δεν υποστηρίζεται (σημειώνεται ως **Όχι**), το σύστημά σας δεν κανονισεί τις ελάχιστες απαιτήσεις. Παρακαλώ, προβείτε στις απαραίτητες διαρρυθμίσεις ώστε να εξυλιωθούν τα σφάλματα. Σε αντίθετη περίπτωση, το Joomla! ίσως να μη λειτουργεί σωστά.

Έκδοση PHP >= 4.3.10	Ναι
- Υποστήριξη Συμπίεσης zlib	Ναι
- Υποστήριξη XML	Ναι
- Υποστήριξη MySQL	Ναι
Η γλώσσα για τις συναρτήσεις MB είναι προκαθορισμένη	Ναι
Η υποφόρτωση αλφαριθμητικών για τις συναρτήσεις MB είναι απενεργοποιημένη	Ναι
configuration.php Εγγράψιμο	Ναι

Συνιστώμενες Ρυθμίσεις:

Αυτές οι ρυθμίσεις προτείνονται για την PHP ώστε να υπάρχει πλήρης συμβατότητα με το Joomla!

Πάντως, το Joomla! θα λειτουργεί, ακόμα και αν δεν υπάρχει πλήρης τύπηση με τις απαιτούμενες ρυθμίσεις.

Οδηγία	Συνιστάται	Προσμετρώ
Αποβολή Λειτουργία (Safe Mode):	Ανενεργό	Ανενεργό
Προβολή Σφαλμάτων:	Ανενεργό	Ενεργό
Μεταφόρτωση Αρχείων:	Ενεργό	Ενεργό
Χρήση Αυτόματων Εισαγωγικών (Magic Quotes):	Ανενεργό	Ανενεργό
Γενικός Μεταβλητός (Register Globals):	Ανενεργό	Ανενεργό
Εκτροπή Εξόδου στη Μνήμη (output buffering):	Ανενεργό	Ανενεργό
Αυτόματη Έναρξη Συνεδρίας:	Ανενεργό	Ανενεργό

Joomla! είναι Ελεύθερο Λογισμικό και διανέμεται σύμφωνα με την Άδεια Χρήσης GNU/GPL v2.0

Εικόνα 25: Εγκατάσταση Joomla! Προληπτικός Έλεγχος

Βήμα 5. Στην οθόνη που εμφανίζεται σε αυτό το βήμα, γίνεται ένας προληπτικός έλεγχος των ρυθμίσεων που υπάρχουν. Πρέπει όλες οι επιλογές να είναι πράσινες. Σε περίπτωση που κάποια από τις επιλογές είναι κόκκινη, πρέπει να γίνει αλλαγή.

Βήμα 6. Στο παράθυρο αυτό εμφανίζεται η άδεια χρήσης. Μετά την ανάγνωσή της, η εγκατάσταση συνεχίζεται στο επόμενο βήμα.

Βήμα 7. Σε αυτό το παράθυρο, πρέπει να δοθούν οι ρυθμίσεις που έχουν γίνει στη βάση δεδομένων. Συμπληρώνονται λοιπόν τα παρακάτω στοιχεία:

Είδος βάσης: mysql

Όνομα Server: localhost (καθώς η εγκατάσταση γίνεται τοπικά).

Όνομα χρήστη: Το όνομα του χρήστη που δημιουργήθηκε στην υποενότητα 6.2.2.2.

Κωδικός: Ο κωδικός του χρήστη της βάσης

Όνομα Βάσης Δεδομένων: Το όνομα τη βάσης δεδομένων που δημιουργήθηκε στην υποενότητα 6.2.2.1.

Οι υπόλοιπες ρυθμίσεις, καλό είναι να παραμείνουν ως έχουν.

Βήμα 8. Ρυθμίσεις FTP. Στην περίπτωση που αναφέρονται στο παρόν έγγραφο, η εγκατάσταση γίνεται τοπικά, άρα δεν χρειάζεται να γίνουν ρυθμίσεις. Άρα το βήμα αυτό παρακάμπτεται.

Βήμα 9. Ρυθμίσεις ιστοτόπου. Στο βήμα αυτό, καθορίζεται το επιθυμητό όνομα για τον ιστότοπο που θα δημιουργηθεί. Επίσης πρέπει να δοθούν email και κωδικός πρόσβασης για το διαχειριστή.

Βήμα 10. Σε αυτό το βήμα, εμφανίζεται ένα μήνυμα διαγραφής από το φάκελο joomla, του υποφακέλου installation. Μετά από τη διαγραφή αυτή, το Joomla! έχει εγκατασταθεί και ο ιστότοπος είναι έτοιμος.

Στη συνέχεια του κεφαλαίου αυτού, θα επεκταθεί το σύστημα διαχείρισης περιεχομένου Joomla! με σκοπό να ικανοποιούνται οι ανάγκες που έχουν αναλυθεί στο κεφάλαιο 4.

6.3 Δημιουργία και Παρακολούθηση Ζητημάτων

Ο κύριος λόγος της εκτενούς ανάλυσης στα κεφάλαια 4 και 5 του συστήματος, έχει σχέση κυρίως με το component CIMaS που δημιουργήθηκε όπως έχει προαναφερθεί σε προηγούμενο κεφάλαιο για εκπαιδευτικούς λόγους, αλλά και για λόγους κάλυψης εξειδικευμένων αναγκών. Το component αυτό, δημιουργήθηκε βασισμένο στο μοντέλο Model View Controller.

6.3.1 Model View Controller Patern⁴

Το Model View Controller, είναι ένα πρότυπο σχεδίασης, το οποίο κάνει διαχωρισμό ανάμεσα στην επιχειρησιακή λογική (business logic) και την παρουσίαση (presentation). Η αρχιτεκτονική MVC, βοηθάει αρκετά στο σχεδιασμό και τη δημιουργία νέων εφαρμογών. Πιο αναλυτικά:

Model

Το Model, παρέχει ρουτίνες για τη χειραγώγηση και τη διαχείριση των δεδομένων στην εφαρμογή. Στις περισσότερες περιπτώσεις, τα δεδομένα ανακτούνται από τη βάση δεδομένων (παρόλα αυτά μπορούν να χρησιμοποιηθούν και άλλες πηγές δεδομένων). Για την εφαρμογή των Models, το Joomla! framework παρέχει την abstract (αφηρημένη) κλάση JModel.

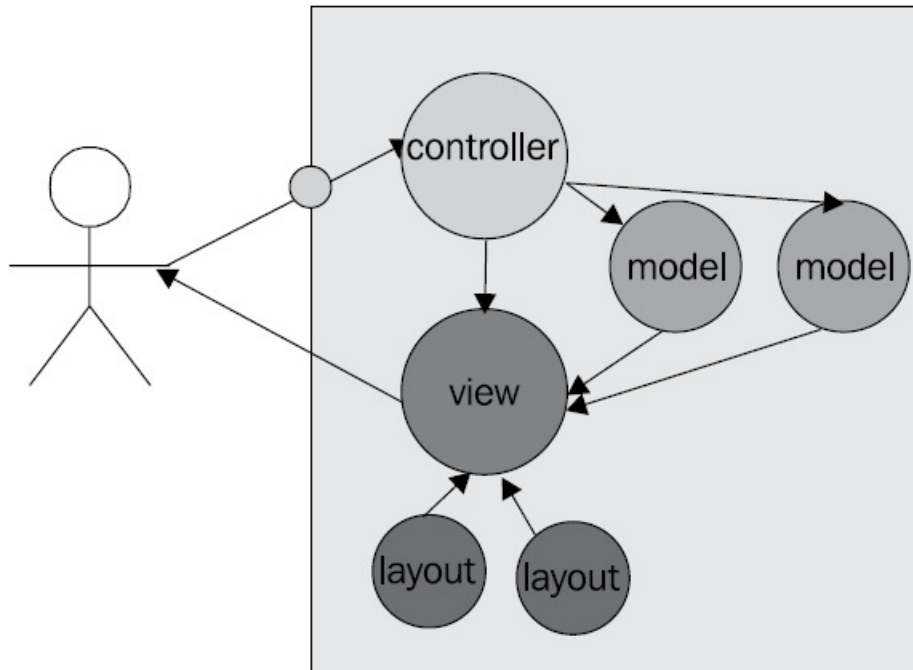
View

Ο ρόλος του View στην εφαρμογή, είναι η εμφάνιση δεδομένων. Τα views δεν τροποποιούν δεδομένα. Το μόνο που κάνουν είναι να εμφανίζουν τα δεδομένα που έχουν ανακτηθεί. Τα views είναι τεχνικά μέρη του παρουσιαστικού επιπέδου και περιέχουν ένα ή περισσότερα templates που παρέχουν τη μορφοποίηση των δεδομένων. Να σημειωθεί ότι τα views μπορούν να εμφανίσουν δεδομένα από πολλά models. Για την εφαρμογή των Views, το Joomla! framework παρέχει την abstract (αφηρημένη) κλάση JView.

Controller

Ο Controller έχει τον διαχειριστικό ρόλο. Δουλειά του είναι να συνδέει το model με τα views. Αντιδρά στις ενέργειες που επιφέρουν την ανάκτηση των δεδομένων (μέσω του model) και τα περνάει στα views για την παρουσίασή τους. Για την εφαρμογή των Controllers, το Joomla! framework παρέχει την abstract (αφηρημένη) κλάση JController.

Στην Εικόνα 26 που ακολουθεί, εμφανίζεται η σχέση των τριών παραπάνω στοιχείων (Model – View – Controller). Παρατηρείται ότι βάσει των επιλογών του χρήστη (το entry point), ενεργοποιείται ο controller, ο οποίος με τη σειρά του αναζητάει και ενεργοποιεί τα κατάλληλα models και views για να εμφανιστεί στο χρήστη η ζητούμενη πληροφορία.



Εικόνα 26: Σχήμα Λειτουργία Model View Controller pattern¹⁹

6.3.2 Απαραίτητες ενέργειες για την υλοποίηση του component

Για να επιτευχθεί η υλοποίηση και εγκατάσταση ενός component, πέρα από τη συγγραφή σωστού κώδικα, θα πρέπει να γίνουν και κάποιες άλλες ενέργειες, ώστε να μην χαθεί ο έλεγχος κατά τη διάρκεια υλοποίησής του.

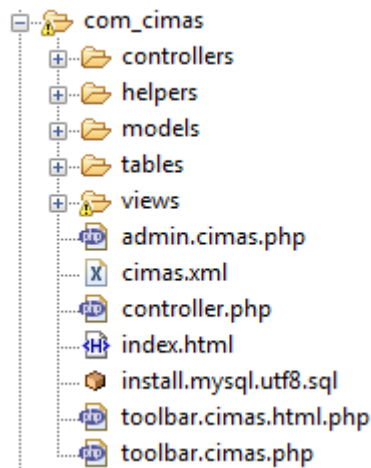
6.3.2.1 Αναγκαία αρχεία και φάκελοι για την λειτουργία του component

Πρώτα, θα πρέπει να δημιουργηθεί ένας φάκελος που θα περιέχει όλα τα αρχεία και τους υποφακέλους που χρειάζονται. Ο φάκελος αυτός, για την περίπτωση που περιγράφεται σε αυτή την πτυχιακή, θα ονομάζεται `com_cimas`. Δίνεται αυτή η ονομασία στο φάκελο για να ενημερωθεί το Joomla! όταν θα γίνει η εγκατάσταση, ότι αυτή αφορά σε component (αυτό θα το καταλάβει από το πρόθεμα του φακέλου `com_`) και το όνομα του component αυτού είναι `cimas` (το όνομα που έχει δοθεί για το component).

Στη συνέχεια, μέσα στο φάκελο αυτό, δημιουργούνται δύο υποφάκελοι. Ένας με την ονομασία «`administrator`» όπου εκεί θα τοποθετηθούν όλα τα αρχεία που χρειάζονται για τη λειτουργικότητα του back end, και ένας φάκελος με την ονομασία «`component`» όπου εκεί θα δημιουργηθούν όλα τα αρχεία που χρειάζονται για τη λειτουργικότητα του component στο front end. Στον ίδιο φάκελο θα τοποθετηθούν επίσης τα αρχεία «`cimas.xml`» και «`index.html`». Στο αρχείο `cimas.xml` θα συμπεριληφθούν οι εντολές για το πώς θα εγκαταστήσει το Joomla! από την περιοχή `extensions` → `install` το συγκεκριμένο component. Τα περιεχόμενα του `cimas.xml` φαίνονται στο παράρτημα. Το αρχείο `index.html`, μόνο δημιουργείται, χωρίς να εισάγεται κώδικας σε αυτό. Είναι απαραίτητο να υπάρχει σε κάθε φάκελο και υποφάκελο, αφενός για να μην φαίνονται τα αρχεία που περιέχονται στους φακέλους των χρηστών που περιηγούνται στη σελίδα μας, και αφετέρου επειδή μέσω αυτού περνάνε όλα τα ερωτήματα προς το root.

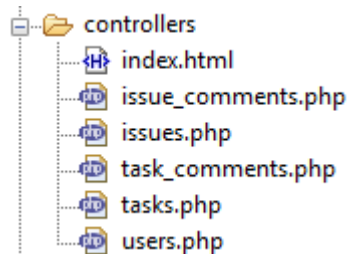
Στο φάκελο `administrator`, πρέπει να δημιουργηθούν τα εξής αρχεία και υποφάκελοι (Εικόνα 27):

¹⁹ Packt, Mastering Joomla! 1.5 Extension and Framework Development, 2010



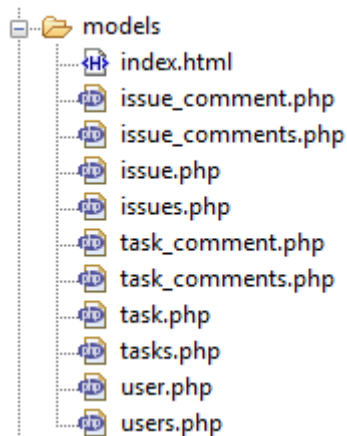
Εικόνα 27: Φάκελος com_cimas\administrator

- Φάκελος controllers: Εκεί θα τοποθετηθούν όλοι οι controllers που χρειάζονται για τη λειτουργία του component (Εικόνα 28).



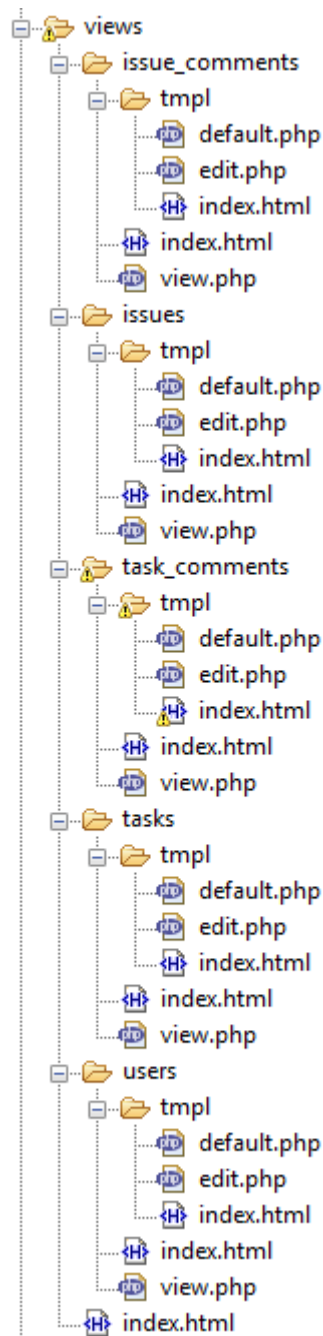
Εικόνα 28: Περιεχόμενα Φακέλου com_cimas\administrator\controllers

- Φάκελος models: Στο φάκελο αυτό, θα αποθηκευτούν όλα τα models.php αρχεία που χρειάζονται για τη λειτουργία του component (Εικόνα 29).



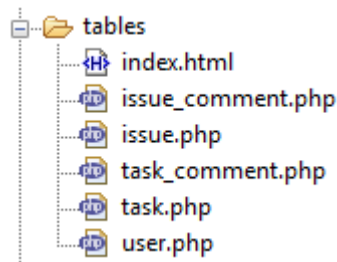
Εικόνα 29: Περιεχόμενα Φακέλου com_cimas\administrator\models

- Φάκελος views: Είναι ο φάκελος στον οποίο θα τοποθετηθούν όλα τα views που χρειάζονται (Εικόνα 30).



Εικόνα 30: Περιεχόμενα Φακέλου `com_cimas\administrator\views`

- Φάκελος helpers: Στο φάκελο αυτό θα τοποθετηθούν όλοι οι helpers που χρειάζονται για τη λειτουργία του component. Τα αρχεία που περιέχονται στο φάκελο helpers, είναι βοηθητικά, καθώς αποθηκεύονται σε αυτά, κομμάτια κώδικα που χρησιμοποιούνται απόφια από διαφορετικά σημεία της εφαρμογής.
- Φάκελος Tables: μέσα στον φάκελο αυτό, περιέχονται όλοι οι πίνακες που έχουν δημιουργηθεί στη βάση δεδομένων για τη λειτουργία του `cimas` component. Ο φάκελος αυτός χρησιμεύει κυρίως για την ενημέρωση του Joomla! σχετικά με την ύπαρξη των πινάκων και των πεδίων που περιέχουν (Εικόνα 31).



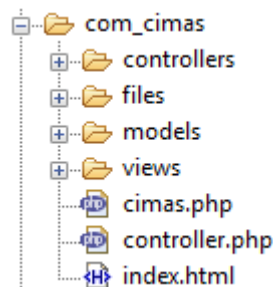
Εικόνα 31: Περιεχόμενα Φακέλου com_cimas\administrator\tables

- Αρχείο admin.cimas.php: Είναι το αρχείο εισόδου για τον administrator. Από αυτό το αρχείο κατευθύνεται το πρόγραμμα για το ποιο controller θα επιλέξει.
- Αρχείο index.html: Το γνωστό πλέον αρχείο
- Αρχείο install.mysql.utf8.sql: Είναι το αρχείο που περιέχει τις εντολές για τη δημιουργία των πινάκων του component cimas στη βάση δεδομένων.
- Αρχείο toolbar.cimas.html.php: Στο αρχείο αυτό, γράφεται κώδικας, για να φορτωθούν οι helpers των toolbars του back end. Οι helpers που καλούνται σε αυτό το αρχείο, έχουν εγκατασταθεί με την αρχική εγκατάσταση του Joomla!.
- Αρχείο toolbar.cimas.php: Στο αρχείο αυτό, επιλέγονται τα κουμπιά που είναι επιθυμητά να φαίνονται στη γραμμή των toolbars του backend (Εικόνα 32).



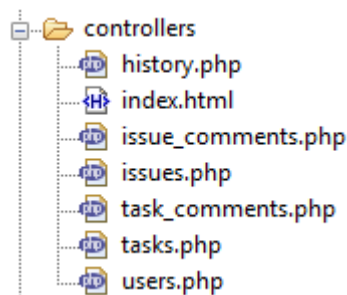
Εικόνα 32: Παράδειγμα toolbar

Για τον φάκελο components, χρειάζονται τα εξής αρχεία και οι υποφακέλοι (Εικόνα 33):



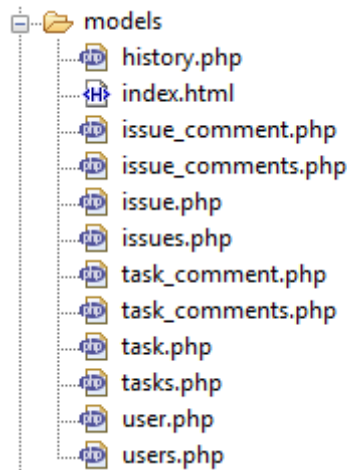
Εικόνα 33: Φάκελος com_cimas\components

- Φάκελος controllers: Εκεί θα τοποθετηθούν όλοι οι controllers που χρειάζονται για τη λειτουργία του component (Εικόνα 34).



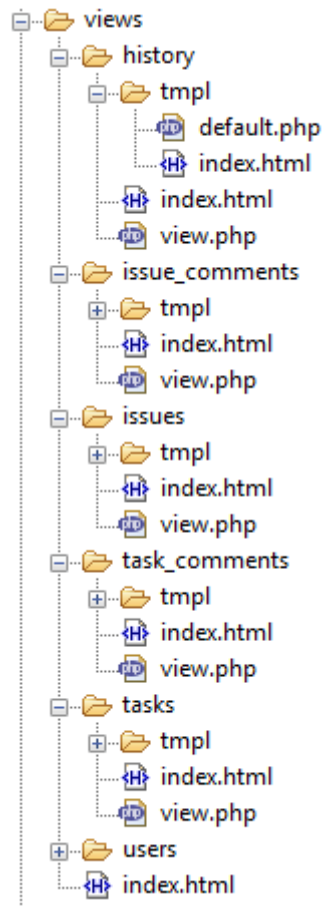
Εικόνα 34: Περιεχόμενα Φακέλου com_cimas\components\controllers

- Φάκελος models: Στο φάκελο αυτό, θα αποθηκευτούν όλα τα models.php αρχεία που χρειάζονται για τη λειτουργία του component (Εικόνα 35).



Εικόνα 35: Περιεχόμενα Φακέλου com_cimas\components\models

- Φάκελος views: Είναι ο φάκελος στον οποίο θα τοποθετηθούν όλα τα views που χρειάζονται (Εικόνα 36).



Εικόνα 36: Περιεχόμενα Φακέλου com_cimas\components\views

- Αρχείο cimas.php: Είναι το αρχείο εισόδου για τον χρήστη του front end. Βάσει αυτού του αρχείου, κατευθύνεται το πρόγραμμα για το ποιον controller θα επιλέξει.
- Αρχείο index.html: Το γνωστό κενό αρχείο

Παρατηρείται, ότι στο backend υπάρχουν ακόμα, οι φάκελοι helpers και tables. Τα tables, αρκεί να δηλωθούν μόνο μια φορά στο backend για την ομαλή λειτουργία του Joomla!. Όσο για το φάκελο helpers, δεν δημιουργείται πρόβλημα αν θα τοποθετηθεί στο front end ή στο back end. Οι συναρτήσεις που περιέχουν τα αρχεία του, μπορούν να καλεστούν και από τα δύο σημεία, ανεξαρτήτως του σημείου αποθήκευσης.

Όλα τα αρχεία που περιγράφονται στην ενότητα αυτή, είναι απαραίτητα για την ομαλή λειτουργία του component βάσει της ανάλυσης απαιτήσεων που πραγματοποιήθηκε στο κεφάλαιο 4.

Ο κώδικας που έχει δημιουργηθεί τόσο για το front end όσο και για το back end της υλοποίησης του component με τη βοήθεια του model view controller pattern, παρουσιάζεται στο παράρτημα Α.

6.3.3 Τρόπος γραφής των Model – View - Controller

Στην παραπάνω ενότητα, παρουσιάστηκαν τα αρχεία που θα χρειαστούν για την υλοποίηση του συστήματος παρακολούθησης ζητημάτων. Στην ενότητα αυτή θα περιγράψουμε τι χρειάζεται ένα σύστημα που χρησιμοποιεί αυτό τον τρόπο ανάπτυξης, για να λειτουργήσει.

6.3.3.1 Model

Τα απαραίτητα κομμάτια κώδικα για να υλοποιηθεί είναι model είναι τα ακόλουθα:

```
<?php
defined('_JEXEC') or die('Restricted access');
\\Αυτή η πρώτη γραμμή κώδικα, είναι μια δήλωση ελέγχου για να βεβαιωθεί
\\αν το αρχείο καλείται μέσα από το session του Joomla. Αυτό προστατεύει
\\την περιοχή, δυσκολεύοντας ένα cracker/hacker να βλάψει το site.
jimport( 'joomla.application.component.model' );
\\Αυτή η γραμμή του κώδικα, χρησιμοποιείται για να φορτωθούν οι
\\βιβλιοθήκες του Joomla! για το model
class cimasModelHistory extends JModel
\\Δημιουργεί ένα αντικείμενο model που ονομάζεται History για το
component \\cimas που κληρονομεί την αφηρημένη (abstract) κλάση JModel.
{
    function __construct() {
        parent::__construct();
    }
    \\εδώ τοποθετείται ο επιθυμητός κώδικας
}
?>
```

Στη συνέχεια τοποθετούμε τα υπόλοιπα κομμάτια επιθυμητού.

6.3.3.2 View

Το view χρειάζεται τις αντίστοιχες εντολές για την ομαλή λειτουργία του. Στο view επίσης δηλώνεται ποιο αρχείο θα φορτωθεί ανα περίπτωση από το φάκελο tmpl.

```
<?php
defined('_JEXEC') or die('Restricted access');
\\Η γνωστή πλέον εντολή ασφαλείας.
jimport( 'joomla.application.component.view' );
\\Αυτή η γραμμή του κώδικα, χρησιμοποιείται για να φορτωθούν οι
\\βιβλιοθήκες του Joomla! για το view
class cimasViewHistory extends JView
\\Δημιουργεί ένα view αντικείμενο που ονομάζεται History για το
\\component cimas που κληρονομεί την αφηρημένη (abstract) κλάση JView.
{
    function display($tmpl=null) \\αποφασίζει ποιο template θα
        \\χρησιμοποιηθεί
    {
        \\εδώ τοποθετείται ο επιθυμητός κώδικας
    }
}
?>
```

6.3.3.3 Controller

Ο controller όπως έχει αναφερθεί, είναι ο «εγκέφαλος». Κάνει την επικοινωνία μεταξύ του model και του view. Αποφασίζει δηλαδή ποιο model και ποιο view θα χρησιμοποιηθεί.

Τα βασικά κομμάτια κώδικά του είναι

```
defined( '_JEXEC' ) or die( 'Restricted access' );
jimport( 'joomla.application.component.controller' );
\\Αυτή η γραμμή του κώδικα, χρησιμοποιείται για να φορτωθούν οι
\\βιβλιοθήκες του Joomla! για τον controller
class cimasControllerHistory extends JController
\\Δημιουργεί ένα αντικείμενο τύπου controller που ονομάζεται History για το
\\component cimas που κληρονομεί την αφηρημένη (abstract) κλάση JController.
{

    \\εδώ τοποθετείται ο επιθυμητός κώδικας

}
```

?>

6.4 CIMaS Component

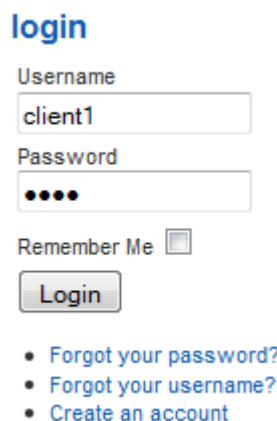
Από τα κομμάτια κώδικα που εμφανίζονται στο Παράρτημα Α, τα αποτελέσματα είναι τα εμφανιζόμενα στις παρακάτω εικόνες. Όπως έχει αναφερθεί και παραπάνω, έχει γίνει υλοποίηση τόσο για τη διαχείριση των ζητημάτων από το front end, όσο και από το back end. Παρακάτω παρουσιάζονται εικόνες και από τα δύο σημεία.

6.4.1 Front End

Στο front end έχουν δικαιώματα εισόδου και επεξεργασίας όλοι οι εγγεγραμμένοι εμπλεκόμενοι χρήστες.

6.4.1.1 Είσοδος στο σύστημα

Η είσοδος στο σύστημα, γίνεται από το default login του Joomla!. Βάσει των στοιχείων που θα δοθούν για είσοδο, αποδίδονται τα αντίστοιχα δικαιώματα.



Εικόνα 37: Είσοδος Χρήστη στο Σύστημα

6.4.1.2 Μενού CIMaS

Μετά την είσοδο οποιουδήποτε χρήστη στο σύστημα, εμφανίζεται το μενού της Εικόνα 38.

CIMaS Menu

- New Issue
- Issues
- Profile
- History
- Statistics

Εικόνα 38: CIMaS Μενού

6.4.1.3 Δημιουργία Νέου Issue

Για τη λειτουργία αυτή, παρέχονται δικαιώματα στην επεξεργασία των δεδομένων, αναλόγως του χρήστη που συνδέεται.

Πελάτης

Για τη δημιουργία ενός νέου ζητήματος για τον πελάτη, εμφανίζεται η φόρμα της Εικόνα 39

The screenshot shows the Joomla! CMS interface. At the top left is the Joomla! logo with the tagline "...because open source matters". Below it is a "Main Menu" with a "Home" link. To the right is a "login" section with "Hi client1," and a "Log out" button. The "CIMaS Menu" is displayed in the sidebar, listing "New Issue", "Issues", "Profile", "History", and "Statistics". The main content area is titled "New Issue" and contains a form with the following fields: "Title" (Laptop Problem), "Category" (Service), "Arrival date" (2010-12-27 21:18:28), "Priority" (Very High), "Status" (Pending assignment), "User" (client1), and "Content" (I can't open a specific file...). There is an "Attachment" field with a file path and a "Submit" button. At the bottom, there is a copyright notice: "Copyright © 2010 thend. All Rights Reserved. Joomla! is Free Software released under the GNU/GPL License." and a footer: "Powered by Joomla!, valid XHTML and CSS."

Εικόνα 39: Δημιουργία Issue από Πελάτη

Οι διαφοροποιήσεις που υπάρχουν αναλόγως του χρήστη που έχει εισαχθεί, μαρκάρονται για να είναι πιο εμφανείς.

Μόλις ο πελάτης εκτελέσει την εντολή «submit», έχει δημιουργηθεί το ζήτημα, και τα στοιχεία του έχουν αποσταλεί στη βάση δεδομένων όπως φαίνεται και στην Εικόνα 40.

	id	title	category	content	arrival_date	priority	status	user_id	attach
	14	Laptop Problem	1	I can't open a specific file	2010-12-27 21:31:54	4	1	64	DSC03140.JPG

Εικόνα 40: Αποθήκευση στη Βάση Δεδομένων

Υπάλληλος – Διευθυντής

Στην περίπτωση δημιουργίας νέου Issue από κάποιο υπάλληλο η διευθυντή, εμφανίζεται η φόρμα της Εικόνα 41.

The screenshot shows the Joomla! interface for creating a new issue. The Joomla! logo is at the top left. Below it, there are two menus: 'Main Menu' with a 'Home' link, and 'CIMaS Menu' with links for 'New Issue', 'Issues', 'Profile', 'History', and 'Statistics'. A 'login' box shows 'Hi tech!', a 'Log out' button, and a 'Log in' button. The main content area is titled 'New Issue' and contains a form with the following fields: Title (text input: 'Problem at the scanner of our company'), Category (dropdown: 'Service'), Arrival date (text input: '2010-12-27 21:41:44'), Priority (dropdown: 'Very High'), Status (dropdown: 'Pending assignment'), User (dropdown: 'rena'), and Content (text area: 'Rena, we have a serious problem with the lamp of the scanner'). There is an 'Attachment' field with a file path and a 'Submit' button. At the bottom, there is a copyright notice: 'Copyright © 2010 thend. All Rights Reserved. Joomla! is Free Software released under the GNU/GPL License.' and a footer: 'Powered by Joomla!, valid XHTML and CSS.'

Εικόνα 41: Δημιουργία νέου Issue από Υπάλληλο – Διευθυντή

Αμέσως μόλις εκτελεστεί η εντολή «submit», εμφανίζονται στη βάση τα issues της Εικόνα 42

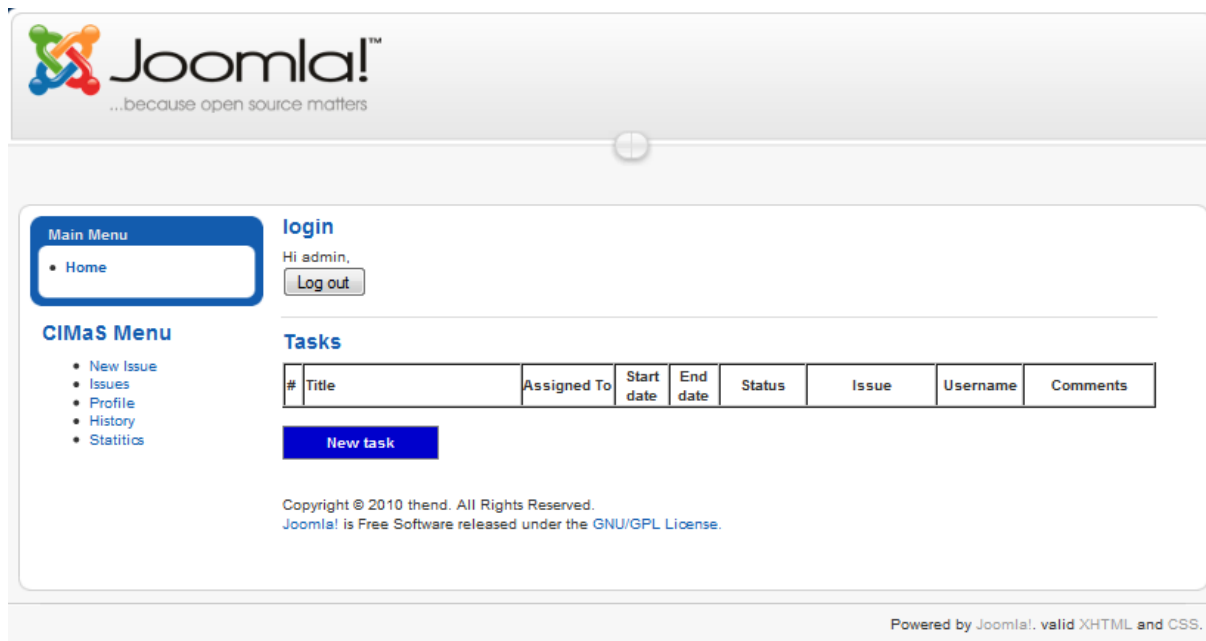
			id	title	category	content	arrival_date	priority	status	user_id	attach
			14	Laptop Problem	1	I can't open a specific file	2010-12-27 21:31:54	4	1	64	DSC03140.JPG
			16	Problem at the scanner of our company	1	Rena, we have a serious problem with the lamp of t...	2010-12-27 21:46:33	4	1	63	IMG_0659.JPG

Εικόνα 42: Αποθήκευση στη Βάση Δεδομένων

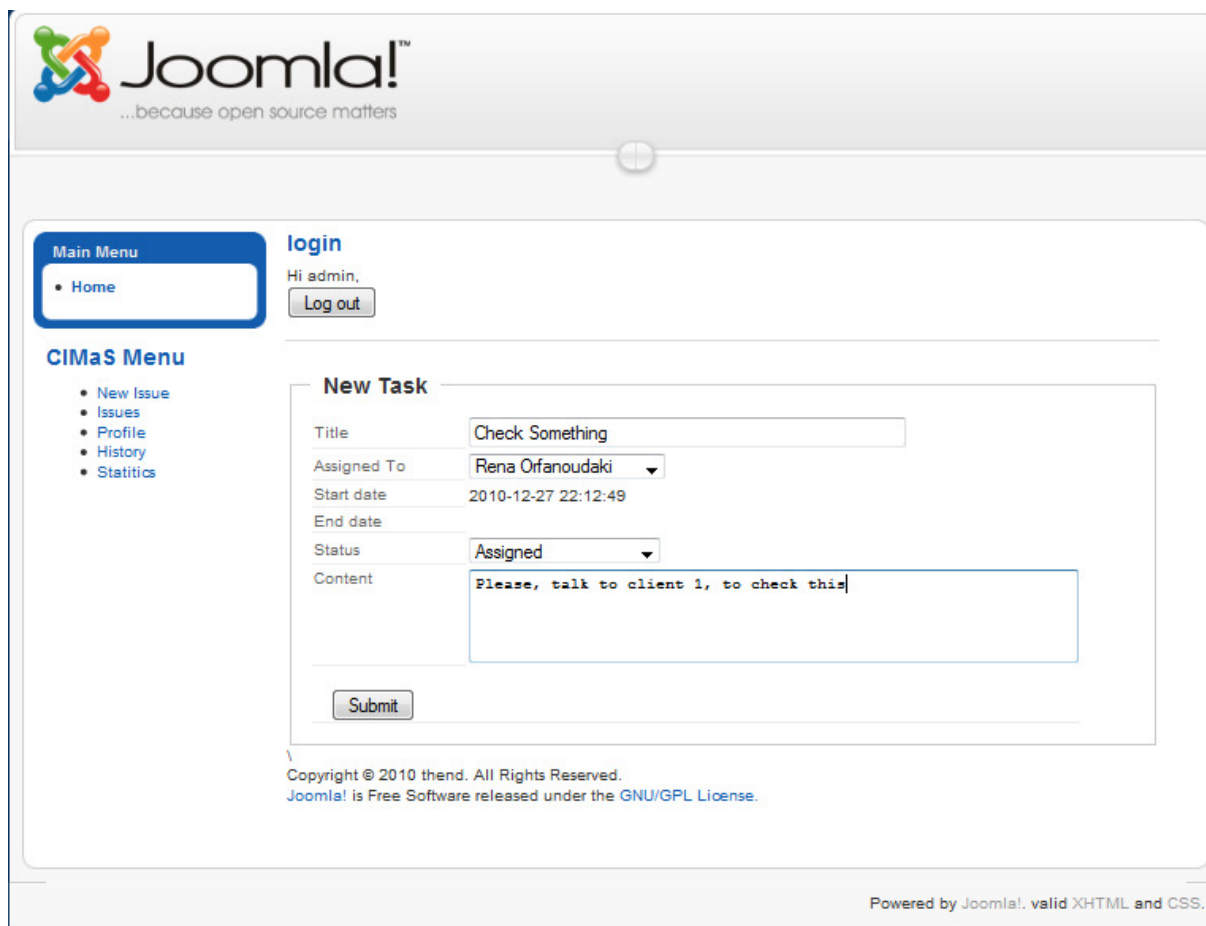
6.4.1.4 Διάσπαση Issue σε Task

Στη συνέχεια, εισέρχεται στο σύστημα κάποιος τεχνικός ή ο διευθυντής, και επόμενο βήμα είναι να κατακερματίσει το issue που δημιουργήθηκε σε tasks. Για να γίνει αυτό, η ακολουθία βημάτων είναι η εξής: επιλέγει για προβολή τα Issues → κλικ στο σύνδεσμο των Tasks → στην Εικόνα 43 επιλέγεται New Task → συμπληρώνονται τα πεδία στη φόρμα New Task → submit. Με τον τρόπο αυτό είναι δυνατόν να δημιουργηθούν όσα tasks χρειάζονται.

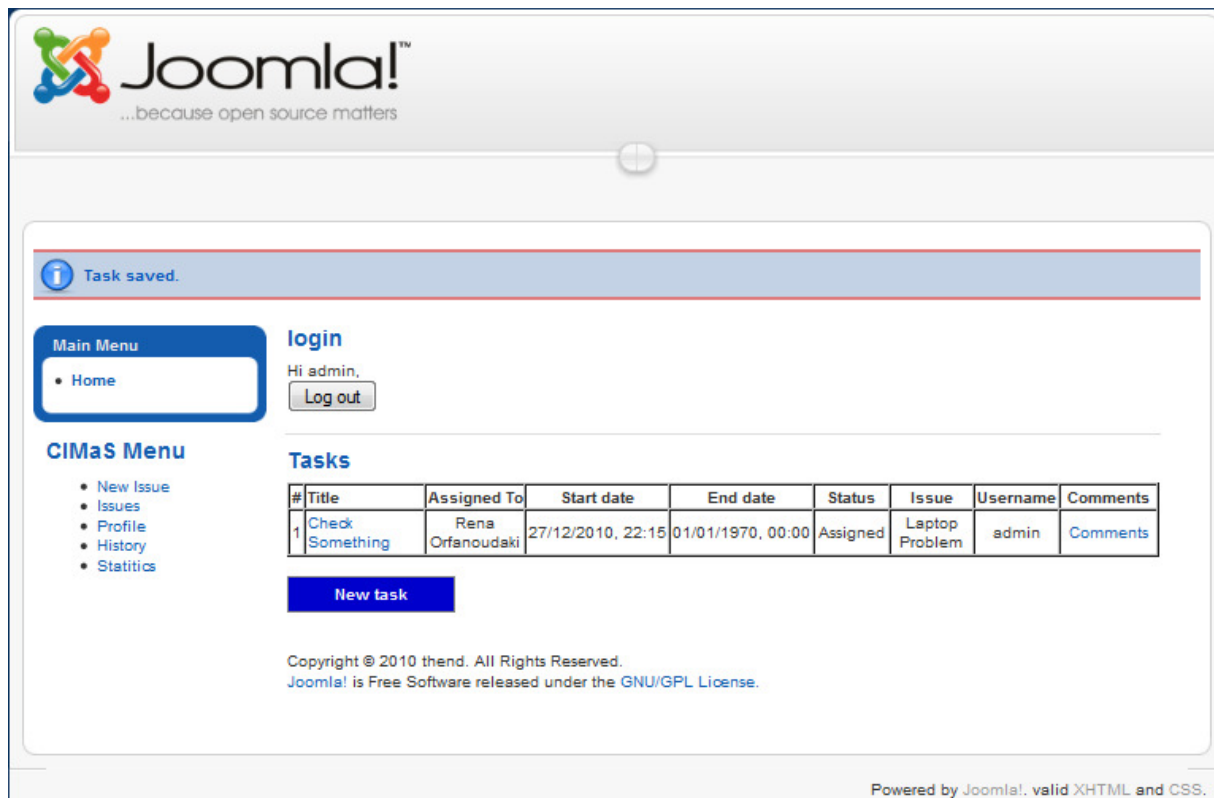
Στη συνέχεια εμφανίζεται η οθόνη της Εικόνα 45. Τα αποτελέσματα στη βάση δεδομένων εμφανίζονται στην Εικόνα 46.



Εικόνα 43: Δημιουργία Tasks



Εικόνα 44: Φόρμα Δημιουργίας Task



Εικόνα 45: Εμφάνιση Task μετά τη Δημιουργία του

	id	title	content	start_date	end_date	assigned_to	status	issue_id	user_id
<input checked="" type="checkbox"/>	16	Check Something	Please, talk to client 1 to check this	2010-12-27 22:15:32	0000-00-00 00:00:00	1	2	14	62

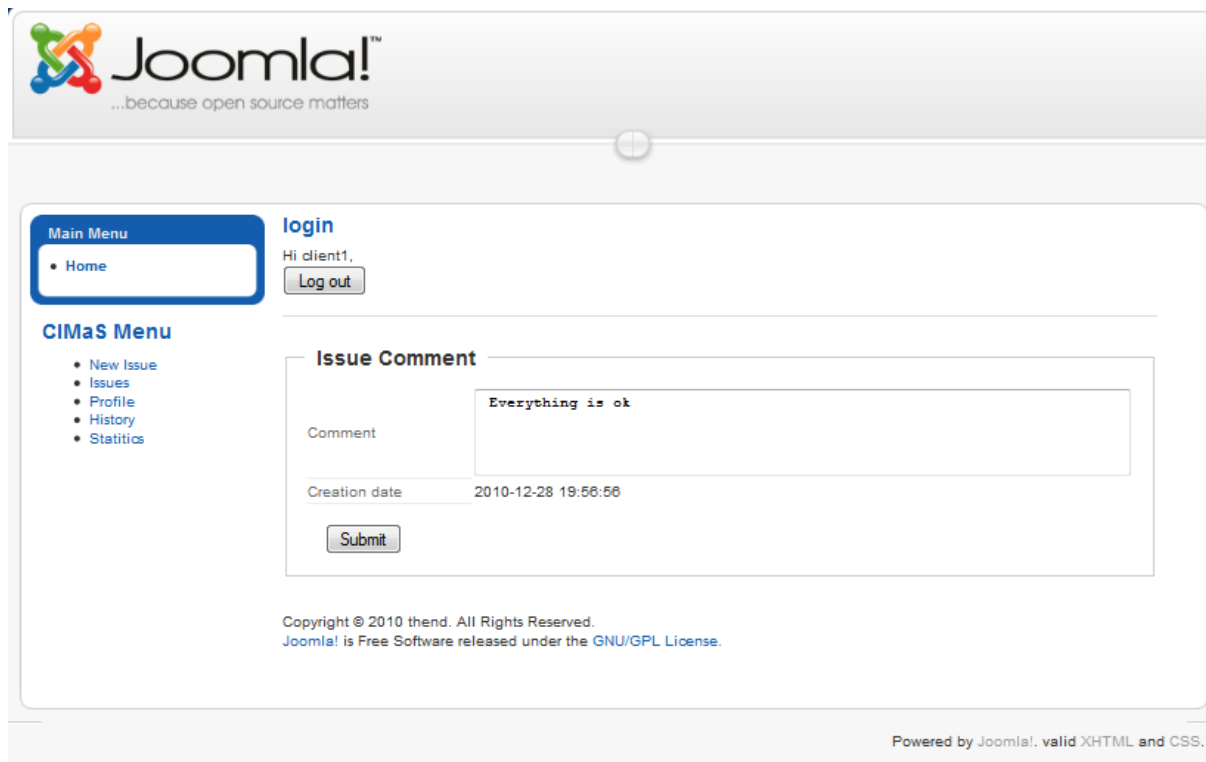
Εικόνα 46: Αποθήκευση στη Βάση Δεδομένων

6.4.1.5 Comments




Για τη δημιουργία comment υπάρχει διαφοροποίηση των λειτουργιών που παρέχονται μεταξύ των χρηστών.

Πελάτης

Για τη δημιουργία σχόλιου από τον πελάτη, ακολουθούνται τα ακόλουθα βήματα επιλογών: Issues→κλικ στο link Comment→ New Comment→Εισαγωγή δεδομένων στη φόρμα (Εικόνα 47)→submit.



Εικόνα 47: Δημιουργία Σχόλιου Πελάτη

	id	comment	creation_date	user_id	issue_id	public_level
  	10	Enverything is ok	2010-12-28 19:56:56	64	14	1

Εικόνα 48: Αποτελέσματα στη Βάση Δεδομένων

Υπάλληλος – Διευθυντής

Για τους χρήστες υπάλληλος και διευθυντής, σε περίπτωση δημιουργίας σχόλιου, εμφανίζεται η φόρμα της Εικόνα 49. Το διαφορετικό στοιχείο που εμφανίζεται για αυτούς, είναι το Public Level, όπου επιλέγουν σε ποιους χρήστες θα είναι ορατά τα σχόλια. Αναφορικά, οι διαβαθμίσεις είναι 1. All users, 2. Manager-Employers, 3. Manager.



Εικόνα 49: Δημιουργία Comment για Υπάλληλο – Διευθυντή

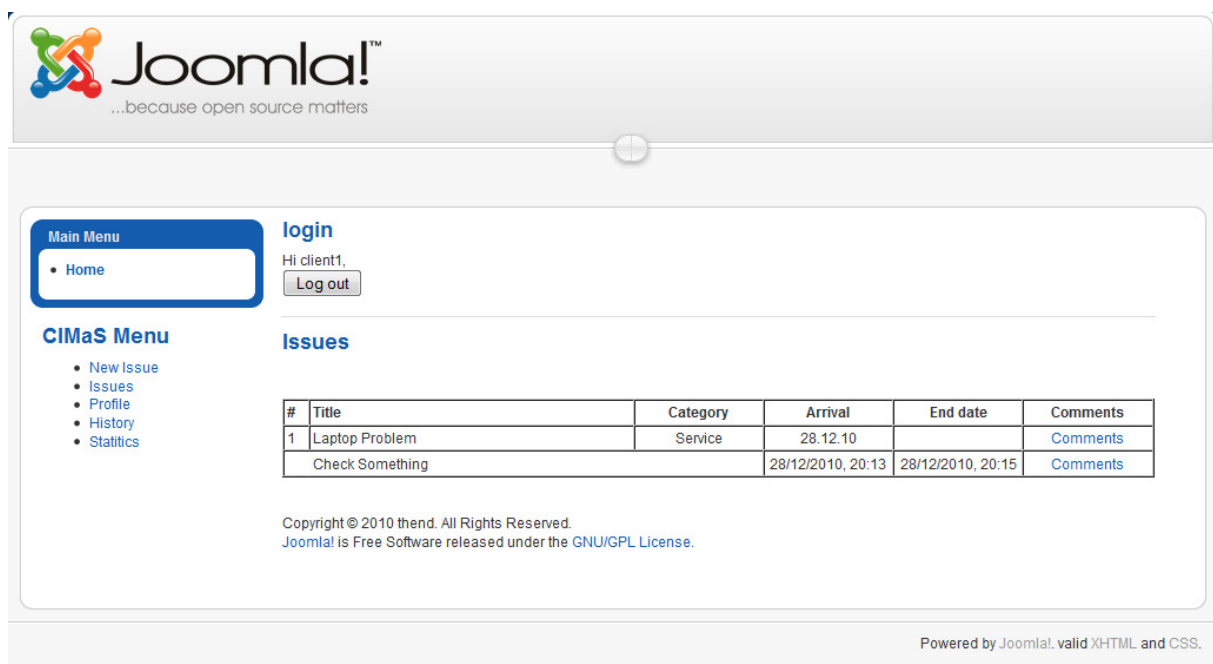
Με την ίδια διαδικασία δημιουργούνται και διαχειρίζονται τα σχόλια σχετικά με τα Tasks.

6.4.1.6 Ιστορικό

Το ιστορικό, επίσης έχει διαβαθμίσεις αναλόγως του χρήστη.

Πελάτης

Για τον πελάτη, εμφανίζεται μόνο το ιστορικό των εργασιών που του έχουν παρασχεθεί. Παράδειγμα φαίνεται στην Εικόνα 50.



Εικόνα 50: Προβολή Ιστορικού Πελάτη

Υπάλληλος – Διευθυντής

Η αντίστοιχη ενέργεια για τους χρήστες διευθυντή και υπάλληλο έχουν μια διαφοροποίηση, μπορούν να επιλέξουν ποιού χρήστη το ιστορικό θέλουν να δουν. Παράδειγμα φαίνεται παρακάτω.

Copyright © 2010 thend. All Rights Reserved.
Joomla! is Free Software released under the GNU/GPL License.

Powered by Joomla!, valid XHTML and CSS.

Εικόνα 51: Προβολή Ιστορικού για Υπάλληλο – Διευθυντή

6.4.2 Back end

Στο back end, παρέχεται πρόσβαση μόνο από τον διευθυντή. Η επιπλέον λειτουργικότητα που παρέχεται στο back end, είναι η δυνατότητα διαχείρισης χρηστών. Κατά τα άλλα, παρέχεται η ίδια λειτουργικότητα όπως στο front end. Αναφορικά, εμφανίζονται παρακάτω, οθόνες με τη λειτουργικότητα από το διαχειριστικό μέρος.

6.4.2.1 Εμφάνιση και επεξεργασία στοιχείων χρηστών

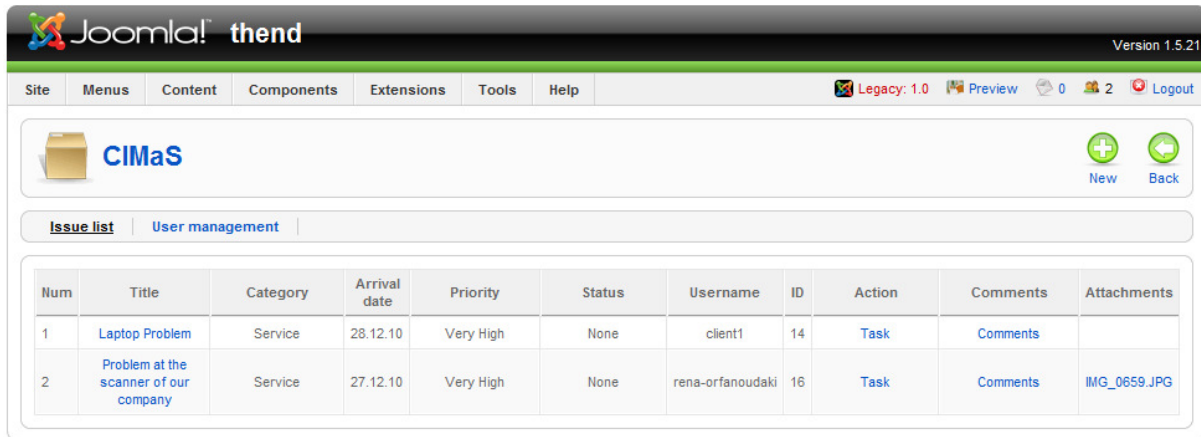
Μετά την επιλογή του χρήστη από μια λίστα, εμφανίζονται τα στοιχεία που έχει δηλώσει για το σύστημα CIMaS (Εικόνα 52). Για λόγους διαχείρισης χρηστών, έχει προστεθεί ένα πεδίο, το External Code, στο οποίο αποθηκεύεται ο κωδικός που έχει ο κάθε χρήστης στα συστήματα οικονομικής διαχείρισης της εταιρείας.

Copyright © 2010 thend. All Rights Reserved.
Joomla! is Free Software released under the GNU/GPL License.

Εικόνα 52: Διαχείριση Στοιχείων Χρήστη απο τον Διευθυντή

6.4.2.2 Εμφάνιση όλων των ανοιχτών Issues (Εικόνα 53)

Στην εικόνα αυτή, εμφανίζονται στο Διευθυντή όλα τα μηνύματα που είναι εκκρεμή.



The screenshot shows the Joomla! thend interface. At the top, there is a navigation menu with options: Site, Menus, Content, Components, Extensions, Tools, Help. On the right, there are icons for Legacy: 1.0, Preview, 0, 2, and Logout. Below the menu, there is a header for CIMaS with 'New' and 'Back' buttons. The main content area has two tabs: 'Issue list' (selected) and 'User management'. Below the tabs is a table with the following data:

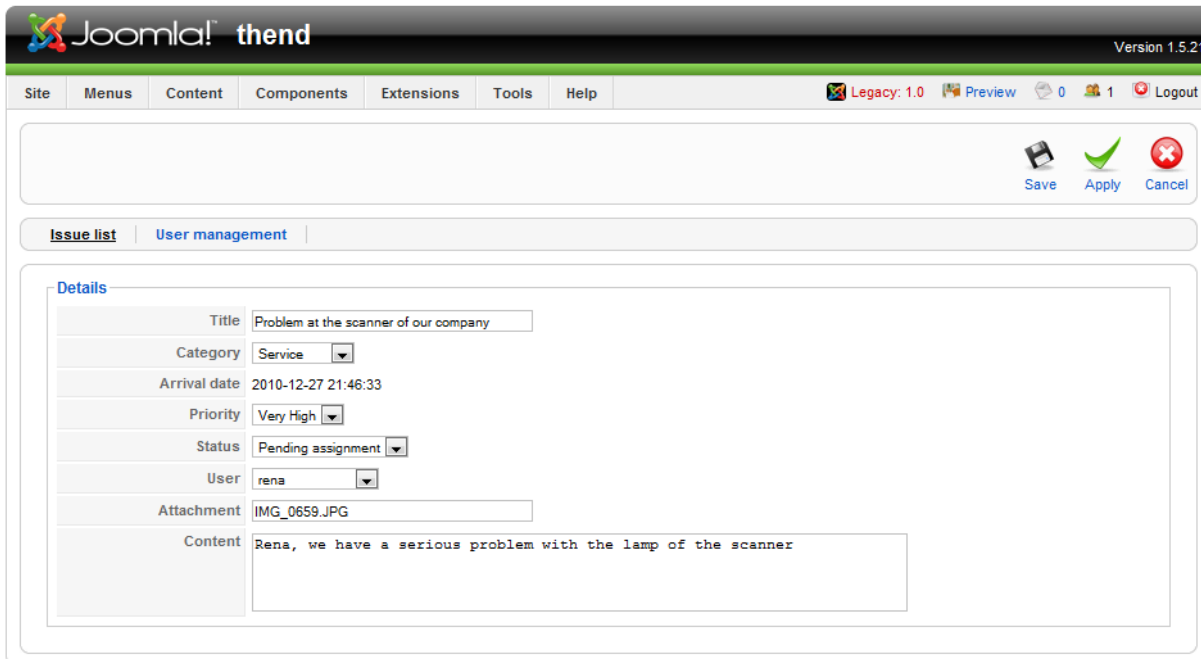
Num	Title	Category	Arrival date	Priority	Status	Username	ID	Action	Comments	Attachments
1	Laptop Problem	Service	28.12.10	Very High	None	client1	14	Task	Comments	
2	Problem at the scanner of our company	Service	27.12.10	Very High	None	rena-orfanoudaki	16	Task	Comments	IMG_0659.JPG

Joomla! is Free Software released under the GNU/GPL License.

Εικόνα 53: Εμφάνιση Μη Ολοκληρωμένων Issues στο Διευθυντή

6.4.2.3 Επεξεργασία Υπάρχοντος Issue

Για να επεξεργαστούν τα στοιχεία για κάποιο issue που έχει ήδη καταχωρηθεί, εμφανίζεται η παρακάτω φόρμα



The screenshot shows the Joomla! thend interface with the 'Issue list' tab selected. At the top right, there are 'Save', 'Apply', and 'Cancel' buttons. Below the tabs, there is a 'Details' section with the following form fields:

- Title: Problem at the scanner of our company
- Category: Service
- Arrival date: 2010-12-27 21:46:33
- Priority: Very High
- Status: Pending assignment
- User: rena
- Attachment: IMG_0659.JPG
- Content: Rena, we have a serious problem with the lamp of the scanner

Joomla! is Free Software released under the GNU/GPL License.

Εικόνα 54: Επεξεργασία υπάρχοντος Issue

6.4.2.4 Δημιουργία Issue

Στην Εικόνα 55 εμφανίζεται η φόρμα για τη δημιουργία νέου issue από το διευθυντή, μέσω back end.

The screenshot shows the Joomla! thend back end interface. At the top, there is a navigation menu with 'Site', 'Menus', 'Content', 'Components', 'Extensions', 'Tools', and 'Help'. The version is 1.5.21. Below the menu, there are icons for 'Legacy: 1.0', 'Preview', '0', '1', and 'Logout'. The main content area has tabs for 'Issue list' and 'User management'. The 'Issue list' tab is active, showing a 'Details' form with the following fields: Title (text input), Category (Service), Arrival date (2010-12-28 20:41:53), Priority (Low), Status (Pending assignment), User (- No User -), Attachment (text input), and Content (text area). At the bottom right of the form, there are 'Save', 'Apply', and 'Cancel' buttons.

Joomla! is Free Software released under the GNU/GPL License.

Εικόνα 55: Δημιουργία Issue απο το back end

6.4.2.5 Δημιουργία Task

Από την παρακάτω φόρμα, δημιουργούνται καινούρια tasks για κάποιο ήδη υπάρχον issue. Αν δεν υπάρχει κάποιο issue, δεν μπορεί να δημιουργηθεί καινούριο task.

The screenshot shows the Joomla! thend back end interface. At the top, there is a navigation menu with 'Site', 'Menus', 'Content', 'Components', 'Extensions', 'Tools', and 'Help'. The version is 1.5.21. Below the menu, there are icons for 'Legacy: 1.0', 'Preview', '0', '1', and 'Logout'. The main content area has tabs for 'Issue list' and 'User management'. The 'Issue list' tab is active, showing a 'Task' form with the following fields: Title (text input), Assigned To (Rena Orfanoudaki), Start date (2010-12-28 20:44:23), End date (text input), Status (Pending assignment), and Content (text area). At the bottom right of the form, there are 'Save', 'Apply', and 'Cancel' buttons.

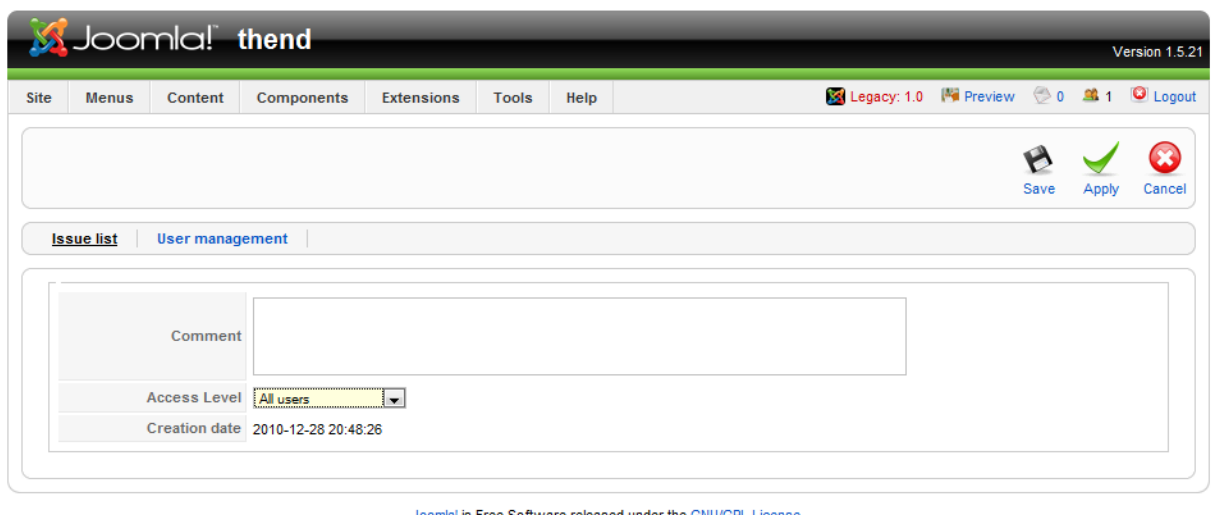
Joomla! is Free Software released under the GNU/GPL License.

Εικόνα 56: Δημιουργία Task απο το back end

Σε περίπτωση επεξεργασίας κάποιου task, εμφανίζεται η παραπάνω φόρμα έχοντας στα πεδία, τα στοιχεία που έχουν δηλωθεί.

6.4.2.6 Δημιουργία Comment

Για τη δημιουργία ενός comment από το back end, είτε είναι σχόλιο για Task είτε σχόλιο για Issue, εμφανίζεται η παρακάτω φόρμα (Εικόνα 57)



Εικόνα 57: Δημιουργία Comment από το back end

Σε περίπτωση επεξεργασίας κάποιου σχόλιου, εμφανίζεται η παραπάνω φόρμα έχοντας στα πεδία, τα στοιχεία που έχουν δηλωθεί.

6.5 Επεκτάσεις που προσφέρουν επιπλέον λειτουργικότητα

Οι παρακάτω επεκτάσεις, ολοκληρώνουν το σύστημα διαχείρισης ζητημάτων, παρέχοντας επιπλέον δυνατότητες. Κάποια από αυτά είναι λειτουργικής σημασίας (όπως η διαχείριση χρηστών του Joomla! και η φόρμα login), κάποιες αποσκοπούν σε προωθητικές ενέργειες (όπως το DOCman και το AcyMailing) και κάποιες αποσκοπούν στη διεξαγωγή ερευνών για την αναβάθμιση του συστήματος, τόσο σε υπολογιστικό (CIMaS), όσο και σε φυσικό (προσωπικού, υπηρεσιών, προϊόντων κ.α.) επίπεδο.

6.5.1 Δημιουργία Λογαριασμού Χρήστη

Για τη δημιουργία λογαριασμού χρήστη στο σύστημα, παρέχεται έτοιμη λειτουργικότητα από το Joomla! μέσω του module login (που φαίνεται στην εικόνα Εικόνα 58).

Εικόνα 58: Λογαριασμός Χρήστη

Λειτουργικότητα που παρέχει αυτό το module:

- ✓ Δημιουργία νέου λογαριασμού
- ✓ Αποστολή password σε περίπτωση που έχει χαθεί ή ξεχαστεί
- ✓ Αποστολή όνομα χρήστη σε περίπτωση που έχει χαθεί ή ξεχαστεί

- ✓ Είσοδος στο σύστημα συμπληρώνοντας το όνομα χρήστη (username) και τον κωδικό πρόσβασης (password)
- ✓ Δυνατότητα διατήρησης σε ετοιμότητα, των στοιχείων αυτών με την ενεργοποίηση της επιλογής να με θυμάσαι (remember me).

6.5.2 Απόδοση δικαιωμάτων ανάλογα με το χρήστη

Η λειτουργικότητα αυτή παρέχεται επίσης έτοιμη από το Joomla!, από το διαχειριστικό μέρος της ιστοσελίδας (όπως φαίνεται και στην Εικόνα 4: Joomla! users Διαχείριση Χρηστών - Απόδοση Δικαιωμάτων). Βάσει αυτής της λειτουργικότητας, ο διαχειριστής μπορεί να κατατάξει το χρήστη σε όποιο γκρουπ επιθυμεί.

6.5.3 Αποστολή και λήψη αρχείων ενδιαφέροντος

6.5.3.1 DOCman (extensions.joomla.org/extensions/directory-a-documentation/downloads/82)

Για την εφαρμογή λειτουργιών upload και download αρχείων ενδιαφέροντος από τους τελικούς χρήστες, θα χρησιμοποιηθεί η εφαρμογή DOCman. Παρέχει την επέκταση του Joomla! για τη χρήση της λειτουργικότητας αυτής, με component, module και plugin. Το DOCman είναι μια λύση διαχείρισης εγγράφων για το Joomla!. Διαχειρίζεται τα έγγραφα σε πολλαπλές κατηγορίες και υποκατηγορίες, δίνει στους χρήστες την άδεια να ανεβάσουν, να κατεβάσουν ή να επεξεργαστούν έγγραφα. Οι δυνατότητες που παρέχει επιγραμματικά είναι οι εξής:

- ✓ Τα έγγραφα μπορούν να οργανωθούν σε όσες κατηγορίες και υποκατηγορίες είναι αναγκαίο.
- ✓ Τα έγγραφα μπορούν να ανήκουν σε ένα συγκεκριμένο εγγεγραμμένο χρήστη, σε ομάδες εγγεγραμμένων χρηστών, σε όλους τους εγγεγραμμένους χρήστες ή σε όλους τους χρήστες.
- ✓ Τα δικαιώματα του συστήματος DOCman παρέχουν πλήρη έλεγχο σε σχέση με το ποιος μπορεί να έχει πρόσβαση, να επεξεργαστεί, να διαχειριστεί, να ανεβάσει και να κατεβάσει έγγραφα.
- ✓ Απομακρυσμένη ή τοπική αποθήκευση. Τα αρχεία μπορούν είτε να φιλοξενοούνται τοπικά ή μπορεί με εύκολο τρόπο να γίνεται σύνδεση με αρχεία που είναι αποθηκευμένα σε έναν απομακρυσμένο server.
- ✓ Παρέχεται η δυνατότητα εφαρμογής ενός μετρητή, που θα καταμετράει τις φορές που έχει μεταφορτωθεί ένα αρχείο έτσι ώστε να παρουσιάζεται πλήρης εικόνα για τη διακίνηση των εγγράφων. Για όλες τις μεταφορτώσεις αποθηκεύεται η πληροφορία της IP του χρήστη, ο browser μέσω του οποίου συνδέθηκε, και της ημερομηνίας – ώρας).
- ✓ Ολοκληρωμένο σύστημα αναζήτησης εγγράφων. Τα έγγραφα μπορούν να αναζητηθούν βάσει ονόματος ή και περιγραφής. Το σύστημα αναζήτησης ενσωματώνεται στο Joomla! προαιρετικά, με τη χρήση ενός plugin.
- ✓ Τα πραγματικά μονοπάτια αποθήκευσης των αρχείων δεν εμφανίζονται στους τελικούς χρήστες.
- ✓ Παρέχει ευκολία στη διαχείριση της όψης και της αισθητικής του χώρου αποθήκευσης εγγράφων.

Για να γίνει η εγκατάσταση του DOCman πρέπει να ακολουθηθούν τα εξής βήματα:

Βήμα 1: Πρέπει να ενεργοποιηθεί το legacy mode (Extensions→Plugin Manager, δημοσίευση του στοιχείου “System - Legacy”). Το legacy mode είναι ένα plugin που χρησιμοποιείται στην εγκατάσταση κάποιων επεκτάσεων στις εκδόσεις Joomla! 1.5.x. Το plugin αυτό, επιτρέπει στις εκδόσεις Joomla! 1.5.x να μιμούνται τις εκδόσεις 1.0.x , ούτως ώστε να τρέχουν στις εφαρμογές, σε περίπτωση που δεν έχουν γίνει οι απαραίτητες αλλαγές που απαιτούνται από τις εκδόσεις 1.5.x και να τρέξουν κάποια επέκταση. Το μόνο μειονέκτημα της τεχνικής αυτής, είναι ότι υπάρχει ελάχιστη μείωση της απόδοσης, αλλά στις περισσότερες περιπτώσεις, αυτή δεν είναι σημαντική.

Βήμα 2: Εγκατάσταση του DOCman μέσω του Joomla! συστήματος extensions→install/uninstall→upload file & install.

6.5.3.2 *AcyMailing* (extensions.joomla.org/extensions/content-sharing/newsletter/10341?qh=YT0xOntpOjA7czoXMDoiYWN5bWFpbGluZy17fQ%3D%3D)

Το AcyMailing, είναι μια επέκταση για το Joomla! κατάλληλο για προώθηση προϊόντων μέσω e-mails και newsletters. Η εφαρμογή αυτή είναι σημαντική, καθώς θα παρέχεται στο σύστημα η δυνατότητα να ενημερώνει μαζικά, ομάδες εγγεγραμμένων χρηστών, χρησιμοποιώντας την αποστολή emails για την ενημέρωση νέων προϊόντων και υπηρεσιών.

Επιπλέον χαρακτηριστικά που παρέχει η επέκταση αυτή είναι:

- ✓ Ενημέρωση διαδικασίας αποστολής, σε πραγματικό χρόνο, χρησιμοποιώντας ένα σύστημα ουράς, δίνοντας έτσι τη δυνατότητα να ξεπεραστεί οποιοσδήποτε περιορισμός από το server.
- ✓ Εύκολη ενσωμάτωση σε άρθρα και ενημερωτικά δελτία του Joomla!
- ✓ Ευκολία στο να περιληφθούν όλες οι πληροφορίες των χρηστών στα ενημερωτικά δελτία.
- ✓ Διαχείριση των εγγεγραμμένων χρηστών και των απλών επισκεπτών.
- ✓ Μαζική εγγραφή με τη χρήση φίλτρων
- ✓ Διαχείριση προτύπων ενημερωτικών δελτίων
- ✓ Δυνατότητα επισύναψης αρχείων και ενσωμάτωσης εικόνων
- ✓ Προεπισκόπηση και αποστολή δοκιμαστικών ενημερωτικών δελτίων σε μια ολόκληρη ομάδα Joomla ή με μια συγκεκριμένη διεύθυνση e-mail
- ✓ Ενημερωτικά στατιστικά διαχείρισης (πόσα μηνύματα στάλθηκαν, πόσα διαβάστηκαν κτλ.)
- ✓ Αυτόματη εγκατάσταση των plugins, modules και των templates
- ✓ Χρήση ασφαλούς σύνδεσης SMTP (δυνατότητα χρησιμοποίησης Gmail για την αποστολή ενημερωτικών δελτίων).
- ✓ Αυτόματη διασύνδεση με τους χρήστες που υπάρχουν στο Joomla!
- ✓ Καταγραφή της IP διεύθυνσης του χρήστη, για να είναι συμβατό με τη νομοθεσία του κάθε κράτους (να προσαρμόζονται δηλαδή τα περιεχόμενα των newsletters και των emails).
- ✓ Επικύρωση και επιβεβαίωση της διεύθυνσης ηλεκτρονικού ταχυδρομείου.
- ✓ Δυνατότητα μετάφρασης σε πάνω από 30 γλώσσες.

Στα θετικά της επέκτασης αυτής, είναι ότι έχει σχεδιαστεί και υλοποιηθεί με το pattern Model View Controller (θα αναλυθεί παρακάτω στο κεφάλαιο αυτό), που επιτρέπει την εύκολη παραμετροποίηση του με την μετατροπή κομματιών κώδικα.

Για να γίνει η εγκατάσταση του AcyMailing, πρέπει να ακολουθηθεί η εξής σειρά: extensions→install/uninstall→upload file & install.

6.5.3.3 *Pools*

Για τη δημιουργία ερωτηματολογίων και για την εξαγωγή στατιστικών στη συνέχεια, θα χρησιμοποιηθεί η προεγκατεστημένη λειτουργικότητα του Joomla! για δημοσκοπήσεις.

Τα αποτελέσματα αυτά, θα συγκεντρώνονται και θα αναλύονται από το διευθυντή, με σκοπό τη βελτιστοποίηση των παρεχόμενων προϊόντων και κατ' επέκταση την αύξηση του κέρδους.

6.6 **Περίληψη Κεφαλαίου**

Στο κεφάλαιο αυτό, παρουσιάστηκε το πρότυπο Model View Controller που χρησιμοποιείται για την υλοποίηση εφαρμογών από πλειάδα προγραμματιστών. Επίσης απεικονίστηκε η εφαρμογή που υλοποιήθηκε με τη βοήθεια του κώδικα, ο οποίος αναφέρεται στο παράρτημα Α. Τέλος αναφέρθηκαν κάποιες άλλες έτοιμες επεκτάσεις του Joomla!, όπου με τη βοήθειά τους, θα προσφέρουμε επιπλέον λειτουργικότητα για ένα ολοκληρωμένο σύστημα διαχείρισης ζητημάτων.

ΚΕΦΑΛΑΙΟ 7

Αποτελέσματα

Τα αποτέλεσμα της παρούσας πτυχιακής, ικανοποιούν ως κάποιο βαθμό του στόχους που είχαν τεθεί και καταγραφεί στο 1^ο κεφάλαιο. Το συμπέρασμα αυτό, αιτιολογείται καθώς τελικά δημιουργήθηκε μια εφαρμογή (ένα component) με τη βοήθεια PHP και μοντέλο Model View Controller, που μπορεί να λειτουργεί με τη βοήθεια του συστήματος διαχείρισης περιεχομένου Joomla!. Επίσης το σύστημα διαχείρισης ζητημάτων που δημιουργήθηκε τελικά, ικανοποιεί επιμέρους στόχους που χτίζονταν κατά τη διάρκεια της ανάλυσης των απαιτήσεων και της μελέτης που έγινε για αυτό.

Το σύστημα που δημιουργήθηκε, είναι πλήρως λειτουργικό και θα μπορούσε να χρησιμοποιηθεί από οποιαδήποτε εταιρεία ή οργανισμό, που ασχολείται με διαχείριση ζητημάτων κατόπιν αιτήσεως τους. Θα χρειαστεί μόνο η παραμετροποίησή του, ώστε να καλύπτονται οι απαιτήσεις που υπάρχουν ανά περίπτωση χρήσης, καθώς όπως έχει αναφερθεί και σε προηγούμενο κεφάλαιο, κάθε οργανισμός – εταιρεία έχει ρουτίνες και ροή εργασιών, βάσει του αντικειμένου ενασχόλησης, του απευθυνόμενου κοινού και του ύψους του κύκλου εργασιών του.

7.1 Συμπεράσματα

Στην εργασία αυτή, μελετήθηκαν τα δημοφιλέστερα CMS και κατόπιν σύγκρισης μεταξύ τους, επιλέχθηκε το Joomla! για τη δημιουργία μιας καινούριας εφαρμογής, ενός συστήματος διαχείρισης ζητημάτων. Με τη βοήθεια του Wampserver, στήθηκε ο server που παρείχε έτοιμη λειτουργικότητα για τη διαχείριση της βάσης δεδομένων, ο διερμηνέας της PHP και ο HTTP server.

Μελετήθηκαν συστήματα και εφαρμογές, παρόμοιες με το επιθυμητό προς υλοποίηση, για τη διευκόλυνση και λήψη ιδεών.

Στη συνέχεια, με τη βοήθεια της σφαιρικής αυτής ενημέρωσης, άρχισε η διαδικασία υλοποίησης της επιθυμητής εφαρμογής.

Αρχικά, έγινε ανάλυση και καταγραφή απαιτήσεων, διαδικασία πάρα πολύ σημαντική για τις αποφάσεις που θα παρθούν κατά τη διάρκεια της υλοποίησης. Μέσα από το στάδιο αυτό, εντοπίζονται οι ανάγκες που πρέπει να ικανοποιούνται από την εφαρμογή, καθώς επίσης και οι εμπλεκόμενοι χρήστες. Η εξειδικευμένη αναζήτηση απαιτήσεων από τη σκοπιά του κάθε χρήστη για την ικανοποίησή τους, είναι ζωτικής σημασίας, καθώς στη περίπτωση που δεν ικανοποιηθούν αυτές, η εφαρμογή θα απαξιωθεί και δεν θα χρησιμοποιείται, άρα θα υπάρχει αποτυχία.

Στη συνέχεια, καταγράφηκαν τα σενάρια χρήσης, πάρα πολύ σημαντική επίσης διαδικασία, καθώς αποτυπώνεται η εφαρμογή σε ένα πρώιμο στάδιο, ικανό να παρουσιαστεί και σε εξωτερικούς χρήστες για την ανταλλαγή απόψεων με στόχο τη βέλτιστη απόδοση της επιθυμητής λειτουργικότητας, κατά το στάδιο της υλοποίησης.

Επίσης πολύ σημαντικό στάδιο είναι η διαγραμματική απεικόνιση της business logic του υπό υλοποίηση συστήματος. Στο στάδιο αυτό, δημιουργήθηκαν workflows ανά εμπλεκόμενο χρήστη, και τα data flow diagrams, επίσης αναλόγως του χρήστη. Με τον τρόπο αυτό, καθορίζεται η «λογική» συνοχή που θα υπάρχει στο σύστημα, βασισμένη στη διακίνηση των δεδομένων και στον τρόπο επεξεργασίας τους.

Αναλύθηκε και δημιουργήθηκε το πρότυπο της βάσης δεδομένων, για την αποθήκευση και διαχείριση των απαραίτητων δεδομένων. Ζωτικής σημασίας το στάδιο αυτό, καθώς με μία κακοσχεδιασμένη βάση, δημιουργούνται προβλήματα στην επικοινωνία μέσω ερωτημάτων (SQL) με τη βάση δεδομένων, και τα αποτελέσματα είναι αποκαρδιωτικά.

Τέλος μελετήθηκε το μοντέλο αρχιτεκτονικής σχεδίασης εφαρμογών Model View Controller, στο οποίο βασίστηκε η υλοποίηση του CIMaS. Όλο το σύστημα στήθηκε με τη βοήθεια PHP για τη διαχείριση και HTML για την παρουσίαση.

Συμπερασματικά, από την πτυχιακή αυτή, επεκτάθηκαν σε ικανοποιητικό βαθμό οι γνώσεις μου. Επίσης εμφανίστηκε ένα πάρα πολύ ενδιαφέρον για εμένα κομμάτι της επιστήμης της πληροφορικής. Η υλοποίηση εφαρμογών προς χρήση, μέσω της ανάπτυξης κώδικα.

Τελειώνοντας αυτή την πτυχιακή, υπάρχει η κούραση (λόγο του όγκου της εργασίας και της πρωτότερης απειρίας στο συγκεκριμένο αντικείμενο), αλλά σε μεγαλύτερο βαθμό υπάρχει η όρεξη επέκτασης των γνώσεων και η μετέπειτα ενασχόληση με ανάπτυξη εφαρμογών.

7.2 Μελλοντική Εργασία και Επεκτάσεις

Μελλοντικά, θα επεξεργαστεί το κομμάτι της παρουσίασης, με προσθήκη γραφικών.

Πολύ σημαντικό μειονέκτημα εδώ, είναι η έλλειψη εσωτερικής αναζήτησης, η οποία υπολογίζεται να προστεθεί σε κάποια από τις αναβαθμίσεις, καθώς επίσης θα αναπτυχθεί περαιτέρω το user management για λόγους ευχρηστίας.

Επίσης στόχος είναι, να πραγματοποιηθεί έρευνα για την περαιτέρω λειτουργικότητα που χρειάζεται για την κάλυψη περισσότερων αναγκών.

Ο τελικός στόχος που τίθεται για το component αυτό, αφορά στη μελλοντική ανάρτησή του (μετά από αλλαγές που θα πραγματοποιηθούν) στο Joomla!.org, καθώς μπορεί να μην είναι το καλύτερο που υπάρχει, αλλά λόγω της δομής και του προτύπου (MVC) που έχει ακολουθηθεί για την υλοποίησή του, μπορεί να παραμετροποιηθεί εύκολα από κάποιον με γνώσεις PHP.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- A/A Πηγή**
- [1] Apress, Foundation Joomla! 2009
 - [2] Prentice Hall, Joomla! Guide 1.5 Joomla!book, 2008
 - [3] Wiley, Joomla! Bible, 2010
 - [4] Pact, Pro Drupal Development
 - [5] Apress, Building Online Communities with Drupal, phpBB and Wordpress, 2006
 - [6] Wrox.WordPress, 24 Hour Trainer, Nov. 2009
 - [7] Packt, Mastering Joomla! 1.5 Extension and Framework Development, 2010

ΠΗΓΕΣ

A/A Πηγή

- [1] [el/Wikipedia.org/wiki/html](http://el.wikipedia.org/wiki/html)
- [2] en.wikipedia.org/wiki/Drupal
- [3] www.siteground.com/tutorials/drupal-tutorial
- [4] en.wikipedia.org/wiki/Wordpress
- [5] www.blogsdna.com/1601/wordpress-27-release-candidate-1-rc-1-released-with-new-icon-set-download-now.htm
- [6] webdesign.oceanartwork.com/wp-content/uploads/2009/07/manage_users.jpg
www.cmsmatrix.org
- [7] en.wikipedia.org/wiki/Issue_tracking_system
- [8] freestyle-Joomla!.com/products/freestyle-support-porta
- [9] www.fast-forward.es/joomla
- [10] www.huruhelpdesk.net
- [11] www.ilient.com
- [12] [en.wikipedia.org/wiki/Data_\(computing\)](http://en.wikipedia.org/wiki/Data_(computing))
- [13] el.wikipedia.org/wiki/Πληροφορία
- [14] en.wikipedia.org/wiki/Php
- [15] en.wikipedia.org/wiki/MySQL
- [16] en.wikipedia.org/wiki/Apache_HTTP_Server

Παράρτημα Α: Πηγαίος Κώδικας CIMaS component

Front End

site/cimas.php

```
<?php
// no direct access
defined('_JEXEC') or die('Restricted access');

// Require the base controller
require_once (JPATH_COMPONENT.DS.'controller.php');
JTable::addIncludePath(JPATH_COMPONENT.DS.'tables');

// Require specific controller if requested
$controller = JRequest::getVar( 'controller', 'Issues' );

require_once(JPATH_COMPONENT.DS.'controllers'.DS.$controller.'.php');

// Create the controller
$controller = 'cimasController'.$controller;
$controller = new $controller();

$controller->execute(JRequest::getCmd('task', 'issues' ));
// redirect
$controller->redirect();
?>
```

Controllers

site/controllers/history

```
<?php
// no direct access
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.controller');

class cimasControllerHistory extends JController {
    // Όνομα model
    // Όνομα του φακέλου που είναι το view
    // Όνομα του template == όνομα του task
    function display() {

        $view = & $this->getView("History");

        $model = & $this->getModel('history');
        $view->setModel($model, true);

        // Display the issues
        $view->setLayout('default'); // Φόρτωση το tmpl/default.php
        $view->display();
    }
}
?>
```

site/controllers/issue_comments

```
<?php
defined('_JEXEC') or die('Restricted access');
```

```

jimport('joomla.application.component.controller');

class cimasControllerIssue_comments extends JController {
    // Όνομα model
    // Όνομα του φακέλου που είναι το view
    // Όνομα του template == όνομα του task
    function display() {
        $modelname = JRequest :: getVar('model', 'Issue_comments');
        $task = JRequest :: getVar('task', 'default');

        $view = & $this->getView("Issue_comments"); // Ψάξε στον φάκελο
views/issue_comments

        $model = & $this->getModel($modelname);
        $view->setModel($model, true);
        $issueId = JRequest :: getVar('issue_id', 0);

        switch ($task) {
            case 'add' :
            case 'edit' :
                $view->setLayout('edit'); // Φόρτωσε το
                $id = JRequest :: getVar('id', 0);
                $view->edit($id, $issueId);
                break;
            case 'save':
                $this->do_save();
                $this->
>setRedirect('index.php?option=com_cimas&controller=issue_comments&model=issue_comments&issue_id='.$issueId, 'Comment for issue: saved.');
```

tpl/edit.php

```

                break;
            case 'apply':
                $id = $this->do_save();
                $this->
>setRedirect('index.php?option=com_cimas&controller=issue_comments&model=issue_comment&task=edit&id='.$id, 'Comment for issue: saved.');
```

tpl/default.php

```

                break;

            case 'cancel' :
                $model = & $this->getModel('Issue_comments');
                $view->setModel($model, true);
            default :
                $view->setLayout('default'); // Φόρτωσε το
                $view->display();
        }
    }

    function do_save() {
        $row = & JTable :: getInstance('issue_comment', 'Table');
        if (!$row->bind(JRequest :: get('post'))) {
            echo "<script> alert('" . $row->getError() . "');
            window.history.go(-1); </script>\n";
            exit ();
        }

        if($row->user_id == 0){
            $user = & JFactory :: getUser();
            $row->user_id = $user->id;
        }
    }
}

```

```

        if (!$row->store()) {
            echo "<script> alert('" . $row->getError() . "');
            window.history.go(-1); </script>\n";
            exit ();
        }

        return $row->id;
    }
}
?>

site/controllers/issues

<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.controller');
jimport('joomla.filesystem.file');

class cimasControllerIssues extends JController {
    // Όνομα model
    // Όνομα του φακέλου που είναι το view
    // Όνομα του template == όνομα του task
    function display() {

        $modelName = JRequest :: getVar('model', 'Issues');
        $task = JRequest :: getVar('task', 'default');

        $view = & $this->getView("Issues"); // Ψάξε στον φάκελο
views/issues

        $model = & $this->getModel($modelName);
        $view->setModel($model, true);

        // Display the issues

        switch ($task) {
            case 'add' :
            case 'edit' :
                $view->setLayout('edit'); // Φόρτωσε το
tmpl/default.php

                $id = JRequest :: getVar('id', 0);
                $view->edit($id);
                break;
            case 'save':
                $this->do_save();
                $this->setRedirect('index.php?option=com_cimas',
'Issue saved.');
```

```

                default :
                    $view->setLayout('default'); // Φόρτωσε το
tmpl/default.php
                    $view->display();
                }
            }

function do_save() {
    $row = & JTable :: getInstance('issue', 'Table');

    if (!$row->bind(JRequest :: get('post'))) {
        echo "<script> alert('" . $row->getError() . "')";
        window.history.go(-1); </script>\n";
        exit ();
    }
    $uploadFile = JRequest::getVar('attach', null, 'FILES',
'ARRAY');
    if (is_array($uploadFile) && !$uploadFile['error'] &&
$uploadFile['size'] >= 1 && is_uploaded_file($uploadFile['tmp_name'])) {
        $config = & JFactory::getConfig();
        $uploaded = JFile::upload($uploadFile['tmp_name'],
JPATH_ROOT.'/components/com_cimas/files/'.JFile::getName($uploadFile['name']
));
        $row->attach = JFile::getName($uploadFile['name']);
    }else{
        $row->attach = '';
    }

    if($row->user_id == 0){
        $user = & JFactory :: getUser();
        $row->user_id = $user->id;
    }

    if (!$row->store()) {
        echo "<script> alert('" . $row->getError() . "')";
        window.history.go(-1); </script>\n";
        exit ();
    }
    echo '<pre>';
    print_r( $row );
    echo '</pre>';

    return $row->id;
}
}
?>

```

site/controllers/task_comments

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.controller');

class cimasControllerTask_comments extends JController {
    // Όνομα model
    // Όνομα του φακέλου που είναι το view
    // Όνομα του template == όνομα του task
    function display() {

```

```

$modelname = JRequest :: getVar('model', 'Task_comments');
$task = JRequest :: getVar('task', 'default');

$view = & $this->getView("Task_comments"); // Ψάξε στον φάκελο
views/task_comments

$model = & $this->getModel($modelname);
$view->setModel($model, true);
$taskId = JRequest :: getVar('task_id', 0);

// Display the task's comments

switch ($task) {
    case 'add' :
    case 'edit' :
        $view->setLayout('edit'); // Φόρτωσε το
        $id = JRequest :: getVar('id', 0);
        $view->edit($id, $taskId);
        break;
    case 'save':
        $this->do_save();
        $this->
>setRedirect('index.php?option=com_cimas&controller=task_comments&model=task_comments&task_id='.$taskId, 'Tasks comment saved.');
```

tpl/edit.php

```

        break;
    case 'apply':
        $id = $this->do_save();
        $this->
>setRedirect('index.php?option=com_cimas&controller=task_comments&model=task_comments&task_id='.$id, 'Tasks comment saved.');
```

tpl/default.php

```

        break;

    case 'cancel' :
        $model = & $this->getModel('Task_comments');
        $view->setModel($model, true);
    default :
        $view->setLayout('default'); // Φόρτωσε το
        $view->display();
}

}

function do_save() {
    $row = & JTable :: getInstance('task_comment', 'Table');
    if (!$row->bind(JRequest :: get('post'))) {
        echo "<script> alert('" . $row->getError() . "')";
        window.history.go(-1); </script>\n";
        exit ();
    }

    if($row->user_id == 0){ // TODO αλλαγή όταν βάλουμε drop down
    με τους χρηστές
        $user = & JFactory :: getUser();
        $row->user_id = $user->id;
    }

    if (!$row->store()) {
        echo "<script> alert('" . $row->getError() . "')";
        window.history.go(-1); </script>\n";
    }
}

```



```

        exit ();
    }

    return $row->id;
}
}
?>

site/controllers/tasks

<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.controller');

class cimasControllerTasks extends JController {
    // Όνομα model
    // Όνομα του φακέλου που είναι το view
    // Όνομα του template == όνομα του task
    function display() {
        $modelname = JRequest :: getVar('model', 'Tasks');
        $task = JRequest :: getVar('task', 'default');

        $view = & $this->getView("Tasks"); // Ψάξε στον φάκελο
views/issues

        $model = & $this->getModel($modelname);
        $view->setModel($model, true);
        $issueId = JRequest :: getVar('issue_id', 0);

        // Display the issues

        switch ($task) {
            case 'add' :
            case 'edit' :
                $view->setLayout('edit'); // Φόρτωσε το
tmpl/default.php

                $id = JRequest :: getVar('id', 0);
                $view->edit($id, $issueId);
                break;
            case 'save':
                $this->do_save();
                $this->
>setRedirect('index.php?option=com_cimas&controller=tasks&model=tasks&issue
_id='.$issueId, 'Task saved.');
```

```

}

function do_save() {
    $row = & JTable :: getInstance('task', 'Table');
    if (!$row->bind(JRequest :: get('post'))) {
        echo "<script> alert('" . $row->getError() . "');
        window.history.go(-1); </script>\n";
        exit ();
    }

    if($row->user_id == 0
        $user = & JFactory :: getUser();
        $row->user_id = $user->id;
    }

    if (!$row->store()) {
        echo "<script> alert('" . $row->getError() . "');
        window.history.go(-1); </script>\n";
        exit ();
    }

    return $row->id;
}
}
?>

site/controllers/users

<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.controller');

class cimasControllerUsers extends JController {
    // Όνομα model
    // Όνομα του φακέλου που είναι το view
    // Όνομα του template == όνομα του task
    function display() {
        $modelname = JRequest :: getVar('model', 'User');
        $task = JRequest :: getVar('task', 'edit');

        $view = & $this->getView("Users"); // Ψάξε στον φάκελο
views/users

        $model = & $this->getModel($modelname);
        $view->setModel($model, true);

        // Display the issues

        switch ($task) {

            case 'edit' :
                $view->setLayout('edit'); // Φόρτωσε το
tmpl/default.php

                $userId = & JFactory :: getUser()->id;
                $view->edit($userId);
                break;
            case 'save':
                $this->do_save();
                $this->
>setRedirect('index.php?option=com_cimas&controller=users&model=user',
'User saved.');
```

```

                break;

                default :
                    $view->setLayout('edit'); // Φόρτωσε το
templ/default.php
                    $view->display();
                }
            }

    function do_save() {
        $row = & JTable :: getInstance('user', 'Table');
        if (!$row->bind(JRequest :: get('post'))) {
            echo "<script> alert('" . $row->getError() . "')";
            window.history.go(-1); </script>\n";
            exit ();
        }

        if (!$row->store()) {
            echo "<script> alert('" . $row->getError() . "')";
            window.history.go(-1); </script>\n";
            exit ();
        }

        return $row->user_id;
    }
}
?>

```

Models

site/models/history

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport( 'joomla.application.component.model' );

class cimasModelHistory extends Jmodel
{
    var $issues = null;
    var $_issue_query = null;
    var $_task_query = null;

    function __construct(){
        parent::__construct();
    }
    function getList($userId=0)
    {
        global $mainframe;

        if(empty($this->issues))
        {
            $_issue_query = $this->_buildIssueQuery($userId);
            $this->issues = $this->_getList($_issue_query);

            foreach($this->issues as $current){
                $this->_task_query = $this->
>_buildTaskQuery($current->id);
                $current->tasks = $this->_getList($this->
>_task_query);
            }
        }
    }
}

```

```

    }
}

return $this->issues;
}

function _buildIssueQuery($userId=0)
{
    if(!$this->_issue_query)
    {
        $this->_issue_query = "SELECT #__cimas_issue.id, title,
category, arrival_date, priority, `status`, attach, username";
        $this->_issue_query .= " FROM #__cimas_issue, #__users";
        $this->_issue_query .= " WHERE #__users.id =
#__cimas_issue.user_id ";

        if($userId != 0){
            $this->_issue_query .= " AND
#__cimas_issue.user_id=".$userId;
        }

        $this->_issue_query .= " ORDER BY id ";
    }

    return $this->_issue_query;
}

function _buildTaskQuery($issueId = 0)
{
    $this->_task_query = "SELECT #__cimas_task.id,
#__cimas_issue.title AS issue_title, #__cimas_task.title,
#__cimas_task.content, start_date, end_date, #__cimas_task.status,
issue_id, username";
    $this->_task_query .= " FROM #__cimas_task, #__cimas_issue,
#__users";
    $this->_task_query .= " WHERE #__cimas_task.issue_id =
#__cimas_issue.id AND #__users.id = #__cimas_task.user_id";
    if($issueId != 0){
        $this->_task_query .= " AND #__cimas_task.issue_id =
$issueId";
    }

    $this->_task_query .= " ORDER BY id ";

    return $this->_task_query;
}
}
?>

```

site/models/issue_comment

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport( 'joomla.application.component.model' );

class cimasModelIssue_comment extends Jmodel
{
    var $issue_comment = null;
    var $username = null;
    var $issue = null;

```

```

var $_query = null;

function __construct(){
    parent::__construct();
}
function getIssue_comment($id)
{
    global $mainframe;

    if($id == 0){
        $this->issue_comment =&
JTable::getInstance('issue_comment', 'Table');
    }
    if(empty($this->issue_comment))
    {
        $query = $this->_buildQuery($id);

        $this->_db->setQuery( $query );
        $this->issue_comment = $this->_db->loadObject();

        $issue_id = $this->issue_comment->issue_id;
        $user_id = $this->issue_comment->user_id;

        $user_query = "SELECT username FROM #__users WHERE
`id`=".$user_id;
        $this->_db->setQuery( $query );
        $this->username = $this->_db->loadObject();

        // TODO Comments
    }

    return $this->issue_comment;
}

function _buildQuery($id)
{
    if(!$this->_query)
    {
        $this->_query = "SELECT * FROM #__cimas_issue_comment
WHERE id=".$id;
    }

    return $this->_query;
}
}
?>

```

site/models/issue_comments

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport( 'joomla.application.component.model' );

class cimasModelIssue_comments extends JModel
{
    var $issue_comments = null;
    var $_query = null;

    function __construct(){

```

```

        parent::__construct();
    }
    function getList($issueId = 0, $gid = 0)
    {
        global $mainframe;

        if(empty($this->issue_comments))
        {
            $query = $this->_buildQuery($issueId, $gid);

            $this->issue_comments = $this->_getList($query);
        }

        return $this->issue_comments;
    }

    function _buildQuery($issueId = 0, $gid = 0)
    {
        if(!$this->_query)
        {
            $accessLevel = 0;
            if($gid < 19){
                $accessLevel = 1;
            }else if($gid<25){
                $accessLevel = 2;
            }else{
                $accessLevel = 3;
            }

            $this->_query = "SELECT #__cimas_issue_comment.id,
#__cimas_issue.title, #__cimas_issue_comment.comment,
#__cimas_issue_comment.public_level,creation_date, issue_id, username";
            $this->_query .= " FROM #__cimas_issue_comment,
#__cimas_issue, #__users";
            $this->_query .= " WHERE #__cimas_issue_comment.issue_id
= #__cimas_issue.id AND #__users.id = #__cimas_issue_comment.user_id";
            $this->_query .= " AND
#__cimas_issue_comment.public_level <= $accessLevel";
            if($issueId != 0){
                $this->_query .= " AND
#__cimas_issue_comment.issue_id = $issueId";
            }

            $this->_query .= " ORDER BY id ";
        }

        return $this->_query;
    }
}
?>

```

site/models/issue

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport( 'joomla.application.component.model' );

class cimasModelIssue extends Jmodel

```

```

{
    var $issue = null;
    var $username = null;

    var $_query = null;

    function __construct(){
        parent::__construct();
    }

    function getIssue($id)
    {
        global $mainframe;

        if($id == 0){
            $this->issue =& JTable::getInstance('issue', 'Table');
        }
        if(empty($this->issue))
        {
            $query = $this->_buildQuery($id);

            $this->_db->setQuery( $query );
            $this->issue = $this->_db->loadObject();

            $user_id = $this->issue->user_id;

            $user_query = "SELECT username FROM #__users WHERE
`id`=" . $user_id;

            $this->_db->setQuery( $query );
            $this->username = $this->_db->loadObject();

            // TODO Comments
        }

        return $this->issue;
    }

    function _buildQuery($id)
    {
        if(!$this->_query)
        {
            $this->_query = "SELECT * FROM #__cimas_issue WHERE
id=" . $id;
        }

        return $this->_query;
    }
}
?>

```

site/models/issues

```

<?php
defined( '_JEXEC' ) or die('Restricted access');

jimport( 'joomla.application.component.model' );

class cimasModelIssues extends Jmodel

```



```

{
    var $issues = null;
    var $_query = null;

    function __construct(){
        parent::__construct();
    }

    function getList($userId=0)
    {
        global $mainframe;

        if(empty($this->issues))
        {
            $query = $this->_buildQuery($userId);

            $this->issues = $this->_getList($query);
        }

        return $this->issues;
    }
    function _buildQuery($userId=0)
    {
        if(!$this->_query)
        {
            $this->_query = "SELECT #__cimas_issue.id, title,
category, arrival_date, priority, `status`, attach, username";
            $this->_query .= " FROM #__cimas_issue, #__users";
            $this->_query .= " WHERE #__users.id =
#__cimas_issue.user_id ";

            if($userId != 0){
                $this->_query .= " AND
#__cimas_issue.user_id=".$userId;
            }

            $this->_query .= " ORDER BY id ";
        }
        return $this->_query;
    }
}
?>

```

site/models/task_comment

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport( 'joomla.application.component.model' );

class cimasModelTask_comment extends Jmodel
{
    var $task_comment = null;
    var $username = null;
    var $task = null;
    var $_query = null;

    function __construct(){
        parent::__construct();
    }
}

```

```

function getTask_comment($id)
{
    global $mainframe;

    if($id == 0){
        $this->task_comment =&
JTable::getInstance('task_comment', 'Table');
    }
    if(empty($this->task_comment))
    {
        $query = $this->_buildQuery($id);

        $this->_db->setQuery( $query );
        $this->task_comment = $this->_db->loadObject();

        $task_id = $this->task_comment->task_id;
        $user_id = $this->task_comment->user_id;

        $user_query = "SELECT username FROM #__users WHERE
`id`=" . $user_id;
        $this->_db->setQuery( $query );
        $this->username = $this->_db->loadObject();

        // TODO Comments
    }

    return $this->task_comment;
}

function _buildQuery($id)
{
    if(!$this->_query)
    {
        $this->_query = "SELECT * FROM #__cimas_task_comment
WHERE id=" . $id;
    }

    return $this->_query;
}
}
?>

```

site/models/task_comments

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport( 'joomla.application.component.model' );

class cimasModelTask_comments extends JModel
{
    var $task_comments = null;
    var $_query = null;

    function __construct(){
        parent::__construct();
    }
}

```

```

function getList($taskId = 0)
{
    global $mainframe;

    if(empty($this->task_comments))
    {
        $query = $this->_buildQuery($taskId);

        $this->task_comments = $this->_getList($query);
    }

    return $this->task_comments;
}

function _buildQuery($taskId = 0)
{
    if(!$this->_query)
    {
        $this->_query = "SELECT #__cimas_task_comment.id,
#__cimas_task.title, #__cimas_task_comment.comment,
#__cimas_task_comment.public_level,creation_date, task_id, username";
        $this->_query .= " FROM #__cimas_task_comment,
#__cimas_task, #__users";
        $this->_query .= " WHERE #__cimas_task_comment.task_id =
#__cimas_task.id AND #__users.id = #__cimas_task_comment.user_id";
        if($taskId != 0){
            $this->_query .= " AND
#__cimas_task_comment.task_id = $taskId";
        }

        $this->_query .= " ORDER BY id ";
    }

    return $this->_query;
}
}
?>

```

site/models/task

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport( 'joomla.application.component.model' );

class cimasModelTask extends Jmodel
{
    var $task = null;
    var $username = null;
    var $issue = null;
    var $_query = null;

    function __construct(){
        parent::__construct();
    }
}

```

```

function getTask($id)
{
    global $mainframe;

    if($id == 0){
        $this->task =& JTable::getInstance('task', 'Table');
    }
    if(empty($this->task))
    {
        $query = $this->_buildQuery($id);

        $this->_db->setQuery( $query );
        $this->task = $this->_db->loadObject();

        $issue_id = $this->task->issue_id;
        $user_id = $this->task->user_id;

        $user_query = "SELECT username FROM #__users WHERE
`id`=" . $user_id;
        $this->_db->setQuery( $query );
        $this->username = $this->_db->loadObject();

        // TODO Comments
    }

    return $this->task;
}

function _buildQuery($id)
{
    if(!$this->_query)
    {
        $this->_query = "SELECT * FROM #__cimas_task WHERE
id=" . $id;
    }

    return $this->_query;
}
}
?>

```

site/models/tasks

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport( 'joomla.application.component.model' );

class cimasModelTasks extends JModel
{
    var $tasks = null;
    var $_query = null;

    function __construct(){
        parent::__construct();
    }

    function getList($issueId = 0)

```

```

{
    global $mainframe;

    if(empty($this->tasks))
    {
        $query = $this->_buildQuery($issueId);

        $this->tasks = $this->_getList($query);
    }

    return $this->tasks;
}

function _buildQuery($issueId = 0)
{
    if(!$this->_query)
    {
        $this->_query = "SELECT #__cimas_task.id,
#__cimas_issue.title AS issue_title, #__cimas_task.title,
#__cimas_task.content, start_date, end_date, assigned_to,
#__cimas_task.status, issue_id, username";
        $this->_query .= " FROM #__cimas_task, #__cimas_issue,
#__users";
        $this->_query .= " WHERE #__cimas_task.issue_id =
#__cimas_issue.id AND #__users.id = #__cimas_task.user_id";
        if($issueId != 0){
            $this->_query .= " AND #__cimas_task.issue_id =
$issueId";
        }

        $this->_query .= " ORDER BY id ";
    }

    return $this->_query;
}
}
?>

```

site/models/user

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport( 'joomla.application.component.model' );

class cimasModelUser extends Jmodel
{
    var $user = null;
    var $username = null;

    var $_query = null;

    function __construct(){
        parent::__construct();
    }

    function getUser($userId)

```

```

{
    global $mainframe;

    if(empty($this->user))
    {
        $query = $this->_buildQuery($userId);

        $this->_db->setQuery( $query );
        $this->user = $this->_db->loadObject();

        if($this->user == null){ //
            $this->user =& JTable::getInstance('user',
'Table');
            $this->user->user_id = $userId;
        }

        $user_query = "SELECT username FROM #__users WHERE
`id`=" . $userId;
        $this->_db->setQuery( $user_query );
        $this->user->username = $this->_db->loadObject()-
>username;
    }

    return $this->user;
}

function _buildQuery($userId)
{
    if(!$this->_query)
    {
        $this->_query = "SELECT * FROM #__cimas_user WHERE
user_id=" . $userId;
    }

    return $this->_query;
}
}
?>

```

Views

site/views/history/view

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.view');

class cimasViewHistory extends Jview {
    function display($tpl = null) {
        global $option;

        $model = &$this->getModel();

        $user = & JFactory :: getUser();
        $userId = JRequest::getVar( 'user_id', $user->id );
        $list = $model->getList($userId);

        for ($i = 0; $i < count($list); $i++) {
            $row = & $list[$i];

```

```

        $row->issue_comments = JRoute :: _('index.php?option=' .
$options . '&issue_id=' . $row->id .
'&controller=issue_comments&model=issue_comments');
        for ($j = 0; $j < count($row->tasks); $j++) {
            $task = & $row->tasks[$j];
            $task->task_comments = JRoute ::
_('index.php?option=' . $options . '&task_id=' . $task->id .
'&controller=task_comments&model=task_comments');
        }

        $this->assignRef('list', $list);
        $this->assignRef('user', $user);
        parent :: display($tpl);
    }
}
?>

```

site/views/history/tmpl/default

```

<?php
defined('_JEXEC') or die('Restricted access');
require_once (JPATH_COMPONENT_ADMINISTRATOR.DS.'helpers'.DS.'helper.php');
// Your custom code here

$user = $this->user;

?>

<h3>Issues</h3>

<?php if($user->gid > 18){ ?>
<form name="historyForm" id="historyForm" method="post"
action="index.php?option=com_cimas&controller=history">
    <label for="chooseUser" style="padding-right:20px;">User:</label>
    <?php echo JHTML::_('list.users', 'user_id', 0, 1, NULL, 'name', 0 );
?>
    <input type="submit" value="Go"/>
</form>
<?php } ?>

    <br />
    <br />

<table class="adminlist" cellspacing="0" cellpadding="2" border="1">
<thead>
    <tr>
        <th width="5">
            <?php echo JText::_('Num'); ?>
        </th>
        <th class="title" width="25%" >
            <?php echo 'Title'; ?>
        </th>
        <th class="title" width="10%" align="center" nowrap="nowrap">
            <?php echo 'Category'; ?>
        </th>
        <th align="center" width="10">
            <?php echo 'Arrival'; ?>
        </th>
        <th class="title" width="10%" align="center" nowrap="nowrap">

```

```

                <?php echo 'End date'; ?>
            </th>
            <th class="title" width="10%" align="center" nowrap="nowrap">
                <?php echo 'Comments'; ?>
            </th>
        </tr>
    </thead>

    <tbody>
    <?php
    $k = 0;
    for ($i=0, $n=count( $this->list ); $i < $n; $i++)
    {
        $row = $this->list[$i]; // XXX &

        $link      = '';
        $date = JHTML::_('date', $row->arrival_date,
JText::_('DATE_FORMAT_LC4') );
        ?>
        <tr class="<?php echo "row$k"; ?>">
            <td width="2%">
                <?php echo ($i+1); ?>
            </td>
            <td>
                <?php echo $row->title; ?>
            </td>
            <td align="center">
                <?php echo getCategoryName($row->category); ?>
            </td>
            <td align="center" width="10%">
                <?php echo $date; ?>
            </td>
            <td align="center" width="10%">
                &nbsp;
            </td>
            <td align="center">
                <a href="<?php echo $row->issue_comments; ?>">Comments</a>
            </td>
        </tr>
        <?php
        $k = 1 - $k;

        foreach($row->tasks as $task){
            ?>
            <tr class="<?php echo "row$k"; ?>">
                <td colspan="3" style="padding-left:2em;">
                    <?php echo $task->title; ?>
                </td>
                <td align="center" width="10%">
                    <?php echo fdate($task->start_date); ?>
                </td>
                <td align="center" width="10%">
                    <?php echo fdate($task->end_date); ?>
                </td>
                <td align="center">
                    <a href="<?php echo $task->task_comments; ?>">Comments</a>
                </td>
            </tr>

            <?php

```



```

    }
}
?>
</tbody>
</table>

<br/><br/>
site/views/issue_comments/view

<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.view');

class cimasViewIssue_comments extends Jview {
    function display($tpl = null) {
        global $option;
        $issueId = JRequest::getVar( 'issue_id', '0' );

        $model = &$this->getModel();
        $user = & JFactory :: getUser();
        $list = $model->getList( $issueId, $user->gid );

        // global $mainframe;
        for ($i = 0; $i < count($list); $i++) {
            $row = & $list[$i];
            $row->link = JRoute :: _('index.php?option=' . $option .
                '&id=' . $row->id .
                '&controller=issue_comments&model=issue_comment&task=edit
                ');
        }
        $this->assignRef('list', $list);
        $this->assignRef('user', $user);
        $this->assignRef('issueId', $issueId);
        parent :: display($tpl);
    }

    function edit($id = 0, $issueId = 0) {

        global $option;

        $model = &$this->getModel();

        $issue_comment = $model->getIssue_comment($id);
        if($id == 0){
            $issue_comment->issue_id = $issueId;
        }

        $user = & JFactory :: getUser();

        $this->assignRef('user', $user);
        $this->assignRef('issue_comment', $issue_comment);
        parent :: display();
    }
}
?>

site/views/issue_comments/tmpl/default

<?php

```

```

defined('_JEXEC') or die('Restricted access');

// Your custom code here
require_once (JPATH_COMPONENT_ADMINISTRATOR.DS.'helpers'.DS.'helper.php');

?>

<form
action="index.php?option=com_cimas&controller=issue_comments&model=
issue_comment" method="post" name="adminForm">

<table class="adminlist" cellspacing="0" cellpadding="2" border="1">
<thead>
  <tr>
    <th width="2%">
      <?php echo JText::_('Num'); ?>
    </th>
    <th class="title" width="20%" nowrap="nowrap">
      <?php echo 'Comment'; ?>
    </th>
    <th align="center" width="10%">
      <?php echo 'Creation date'; ?>
    </th>
    <th class="title" align="center" width="20%" nowrap="nowrap">
      <?php echo 'Issue'; ?>
    </th>
    <th width="10%" align="center" class="title">
      <?php echo 'Username'; ?>
    </th>
    <?php if($this->user->gid > 18){ ?>
    <th class="title" align="center" width="15%" nowrap="nowrap">
      <?php echo 'Public Level'; ?>
    </th>
    <?php } ?>
  </tr>
</thead>

<tbody>
<?php
$k = 0;
for ($i=0, $n=count( $this->list ); $i < $n; $i++)
{
  $row = $this->list[$i]; // XXX &

  $link      = '';
  ?>
  <tr class="<?php echo "row$k"; ?>">
    <td>
      <?php echo ($i+1); ?>
    </td>
    <td>
      <a href="<?php echo $row->link; ?>"><?php echo $row->comment;
?></a>
    </td>
    <td nowrap="nowrap" align="center">
      <?php echo fdate($row->creation_date); ?>
    </td>
    <td align="center">
      <?php echo $row->title;?>
    </td>
    <td align="center">

```

```

                <?php echo $row->username; ?>
            </td>
            <?php if($this->user->gid > 18){ ?>
            <td align="center">
                <?php echo getPublic_levelName($row->public_level);?>
            </td>
            <?php } ?>
        </tr>
        <?php
        $k = 1 - $k;
    }
?>
</tbody>
</table>

<input type="hidden" name="issue_id" value="<?php echo $this->issueId; ?>"
/>
<input type="hidden" name="task" value="" />
</form>
<br />
<a style="display:block; border:1px solid gray; text-align:center; width:
10em; padding: 5px; background-color: #0000CC; color:#FFFFFF; font-
weight:bold;"
    href="index.php?option=com_cimas&issue_id=<?php echo $this->issueId;
?>&controller=issue_comments&model=issue_comment&task=edit">New Comment</a>

<br/><br/>
site/views/issue_comments/tmpl/edit

<?php
defined('_JEXEC') or die('Restricted access');
require_once (JPATH_COMPONENT_ADMINISTRATOR.DS.'helpers'.DS.'helper.php');
    $issue_comment = $this->issue_comment;
?>

<form method="post"
action="index.php?option=com_cimas&controller=issue_comments&model=issue_co
mment&task=save" name="adminForm">
    <input type="hidden" name="id" value="<?php echo $issue_comment->id;
?>" />

    <div class="col100">
        <fieldset class="adminform">
            <legend><?php echo JText::_(' Issue Comment ' );
?></legend>

            <table class="admintable">
                <tbody>
                    <tr>
                        <td width="20%" class="key">
                            <label for="comment">
                                Comment
                            </label>
                        </td>
                        <td width="80%">
                            textarea class="inputbox" cols="70" rows="3"
                            name="comment" id="comment"
                            <?php
                            if($this->user->gid <= 18 && $issue_comment->comment!=null) {
                                echo 'disabled="true"';

```

```

        }?>
        <?php echo $issue_comment->comment;?> </textarea>
    </td>
</tr>

    <?php if($this->user->gid > 18){ ?>
<tr>
    <td width="20%" class="key">
        <label for="public_level">
            Public Level
        </label>
    </td>
    <td width="80%">
        <?php echo getPublic_levelMenu($issue_comment->public_level);
?>
    </td>
</tr>
    <?php } ?>

<tr>
    <td width="20%" class="key">
        <label>
            Creation date
        </label>
    </td>
    <td width="80%">
        <?php echo $issue_comment->creation_date;?>
    </td>
</tr>
<tr>
    <td valign="middle" colspan="2" style="padding: 15px 0 0
15px;">
        <input type="submit" value="Submit" />
    </td>
</tr>

</tbody>
</table>
</fieldset>
</div>
<div class="clr"></div>
<input type="hidden" name="id" value="<?php echo $issue_comment->id;?>" />
<input type="hidden" name="issue_comment" value="" />
<input type="hidden" name="issue_id" value="<?php echo $issue_comment->issue_id;?>"
/>
<input type="hidden" name="user_id" value="<?php echo $issue_comment->user_id;?>"
/>
<input type="hidden" name="task" value="" />
</form>
<br/><br/>

```

site/views/issues/view

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.view');

class cimasViewIssues extends Jview {
    function display($tpl = null) {
        global $option;
    }
}

```

```

$model = &$this->getModel();

$user = & JFactory :: getUser();
if($user->gid > 18){
    $list = $model->getList();
}else{
    $list = $model->getList($user->id);
}

// global $mainframe;
for ($i = 0; $i < count($list); $i++) {
    $row = & $list[$i];
    $row->link = JRoute :: _('index.php?option=' . $option .
'&id=' . $row->id . '&controller=issues&model=issue&task=edit');
    $row->task = JRoute :: _('index.php?option=' . $option .
'&issue_id=' . $row->id . '&controller=tasks&model=tasks');
    $row->issue_comments = JRoute :: _('index.php?option=' .
$option . '&issue_id=' . $row->id .
'&controller=issue_comments&model=issue_comments');
    $row->attachment =
'/Joomla/components/com_cimas/files/' . $row->attach;
}

$this->assignRef('list', $list);
parent :: display($tpl);
}

function edit($id = 0) {

    global $option;

    $model = &$this->getModel();
    $issue = $model->getIssue($id);

    $this->assignRef('issue', $issue);
    $user = & JFactory :: getUser();
    $this->assignRef('user', $user);

    parent :: display();
}
}
?>

```

site/views/issues/tmpl/default

```

<?php
defined('_JEXEC') or die('Restricted access');
require_once (JPATH_COMPONENT_ADMINISTRATOR.DS.'helpers'.DS.'helper.php');
?>

<h3>Issues</h3>
<form
action="index.php?option=com_cimas&controller=issues&model=issue"
method="post" name="adminForm">

<table class="adminlist" cellspacing="0" cellpadding="2" border="1">
<thead>
<tr>
<th width="5">
<?php echo JText::_('Num'); ?>

```

```

        </th>
        <th class="title" width="25%" >
            <?php echo 'Title'; ?>
        </th>
        <th class="title" width="10%" align="center" nowrap="nowrap">
            <?php echo 'Category'; ?>
        </th>
        <th align="center" width="10">
            <?php echo 'Arrival'; ?>
        </th>
        <th class="title" width="10%" align="center" nowrap="nowrap">
            <?php echo 'Priority'; ?>
        </th>
        <th class="title" width="10%" align="center" nowrap="nowrap">
            <?php echo 'Status'; ?>
        </th>
        <th width="2%" class="title" align="center">
            <?php echo 'Username'; ?>
        </th>
        <th width="2%" class="title" align="center" >
            <?php echo 'ID'; ?>
        </th>
        <th class="title" width="15%" align="center" nowrap="nowrap">
            <?php echo 'Action' ; ?>
        </th>
        <th class="title" width="15%" align="center" nowrap="nowrap">
            <?php echo 'Comments' ; ?>
        </th>
        <th class="title" width="15%" align="center" nowrap="nowrap">
            <?php echo 'Attachments'; ?>
        </th>
    </tr>
</thead>

<tbody>
<?php
$k = 0;
for ($i=0, $n=count( $this->list ); $i < $n; $i++)
{
    $row = $this->list[$i]; // XXX &

    $link      = '';
    $date = JHTML::_('date', $row->arrival_date,
JText::_('DATE_FORMAT_LC4') );
    ?>
    <tr class="<?php echo "row$k"; ?>">
        <td>
            <?php echo ($i+1); ?>
        </td>
        <td>
            <a href="<?php echo $row->link; ?>"><?php echo $row-
>title; ?></a>
        </td>
        <td align="center">
            <?php echo getCategoryName($row->category); ?>
        </td>
        <td align="center">
            <?php echo $date; ?>
        </td>
        <td align="center">
            <?php echo getPriorityName($row->priority); ?>

```

```

        </td>
        <td align="center">
            <?php echo getStatusName($row->status); ?>
        </td>
        <td align="center">
            <?php echo $row->username; ?>
        </td>
        <td align="center" >
            <?php echo $row->id; ?>
        </td>
        <td align="center">
            <a href="<?php echo $row->task; ?>">Task</a>
        </td>
        <td align="center">
            <a href="<?php echo $row->issue_comments; ?>">Comments</a>
        </td>
        <td align="center">
            <a href="<?php echo $row->attachment; ?>"><?php echo $row-
>attach; ?></a>
        </td>
    </tr>
    <?php
    $k = 1 - $k;
}
?>
</tbody>
</table>

<input type="hidden" name="task" value="" />
<input type="hidden" name="issue_comments" value="" />
</form>
<br/><br/>

```

site/views/issues/tmpl/edit

```

<?php
defined('_JEXEC') or die('Restricted access');
require_once (JPATH_COMPONENT_ADMINISTRATOR.DS.'helpers'.DS.'helper.php');

// Your custom code here
    $issue = $this->issue;
?>

<form
action="index.php?option=com_cimas&controller=issues&model=issue&task=save"
method="post" name="adminForm" enctype="multipart/form-data">
    <input type="hidden" name="id" value="<?php echo $issue->id; ?>" />

    <div class="col100">
    <fieldset class="adminform">
    <legend><?php echo JText::_( 'New Issue' ); ?></legend>
        <table class="admintable">
        <tbody>
            <tr>
                <td width="20%" class="key">
                    label for="title">
                        Title
                    </label>
                </td>
                <td width="80%">

```

```

        <input class="inputbox" type="text" name="title"
id="title" size="50" value="<?php echo $issue->title;?>" />
    </td>
</tr>
<tr>
    <td width="20%" class="key">
        <label for="category">
            Category
        </label>
    </td>
    <td width="80%">
        <?php echo getCategoryMenu($issue->category); ?>
    </td>
</tr>
<tr>
    <td width="20%" class="key">
        <label>
            Arrival date
        </label>
    </td>
    <td width="80%">
        <?php echo $issue->arrival_date;?>
    </td>
</tr>
<tr>
    <td width="20%" class="key">
        <label for="priority">
            Priority
        </label>
    </td>
    <td width="80%">
        <?php echo getPriorityMenu($issue->priority);?>
    </td>
</tr>
<tr>
    <td width="20%" class="key">
        <label for="status">
            Status
        </label>
    </td>
    <td width="80%">
        <?php
            if($this->user->gid > 18){
                echo getStatusMenu($issue->status);
            }else{
                echo 'Pending assignment';
                echo '<input type="hidden"
name="status" value="1" />';
            }
        ?>
    </td>
</tr>
<tr>
    <td width="20%" class="key">

```



```

        <label for="user_id">
            User
        </label>
    </td>
    <td width="80%">
        <?php
            if($this->user->gid > 18){
                echo JHTML::_('list.users', 'user_id',
$issue->user_id, 1, NULL, 'name', 0 );
            }else{
                echo $this->user->username;
                echo '<input type="hidden" name="username"
value="'. $this->user->username. '" />';
            }
        ?>
    </td>
</tr>

<tr>
    <td valign="top" class="key">
        <label for="content">
            Content
        </label>
    </td>
    <td>
        <textarea class="inputbox" cols="36" rows="3"
name="content" id="tags" value="<?php echo $issue->content;?>"></textarea>
    </td>
</tr>

<tr>
    <td width="20%" class="key">
        <label for="attach">
            Attachment
        </label>
    </td>
    <td width="80%">
        <input class="inputbox" type="file" name="attach"
id="attach" size="50" />
    </td>
</tr>

<tr>
    <td colspan="2" style="text-align: center; padding: 15px 0 0 15px;">
        <input type="submit" value="Submit" />
    </td>
</tr>
</tbody>
</table>
</fieldset>
</div>
<div class="clr"></div>
<input type="hidden" name="id" value="<?php echo $issue->id;?>" />
</form>
<br/>
<br/>

```

site/views/task_comments/view

<?php

```

defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.view');

class cimasViewTask_comments extends Jview {
    function display($tpl = null) {
        global $option;
        $taskId = JRequest::getVar( 'task_id', '0' );

        $model = &$this->getModel();
        $list = $model->getList( $taskId );

        // global $mainframe;
        for ($i = 0; $i < count($list); $i++) {
            $row = & $list[$i];
            $row->link = JRoute :: _('index.php?option=' . $option .
                '&id=' . $row->id .
                '&controller=task_comments&model=task_comment&task=edit');
        }

        $this->assignRef('list', $list);
        $this->assignRef('taskId', $taskId);
        parent :: display($tpl);
    }

    function edit($id = 0, $taskId = 0) {
        global $option;

        $model = &$this->getModel();

        $task_comment = $model->getTask_comment($id);
        if($id == 0){
            $task_comment->task_id = $taskId;
        }

        $this->assignRef('task_comment', $task_comment);
        parent :: display();
    }
}
?>

```

site/views/task_comments/tmpl/default

```

<?php
defined('_JEXEC') or die('Restricted access');

// Your custom code here
require_once (JPATH_COMPONENT_ADMINISTRATOR.DS.'helpers'.DS.'helper.php');
?>

<form
action="index.php?option=com_cimas&controller=task_comments&model=t
ask_comment" method="post" name="adminForm">

<table class="adminlist" cellspacing="0" cellpadding="2" border="1">
<thead>
    <tr>
        <th width="2%">

```

```

        <?php echo JText::_ ( 'Num' ); ?>
    </th>
    <th class="title" width="20%" nowrap="nowrap">
        <?php echo 'Comment'; ?>
    </th>
    <th align="center" width="10%">
        <?php echo 'Creation date'; ?>
    </th>
    <th class="title" align="center" width="20%" nowrap="nowrap">
        <?php echo 'Task'; ?>
    </th>
    <th width="10%" align="center" class="title">
        <?php echo 'Username'; ?>
    </th>
    <th class="title" align="center" width="15%" nowrap="nowrap">
        <?php echo 'Public Level'; ?>
    </th>
</tr>
</thead>

<tbody>
<?php
$k = 0;
for ($i=0, $n=count( $this->list ); $i < $n; $i++)
{
    $row = $this->list[$i]; // XXX &

    $link      = '';
    ?>
    <tr class="<?php echo "row$k"; ?>">
        <td>
            <?php echo ($i+1); ?>
        </td>
        <td>
            <a href="<?php echo $row->link; ?>"><?php echo $row-
>comment; ?></a>
        </td>
        <td nowrap="nowrap">
            <?php echo fdate($row->creation_date); ?>
        </td>
        <td align="center">
            <?php echo $row->title;?>
        </td>
        <td align="center">
            <?php echo $row->username; ?>
        </td>
        <td align="center">
            <?php echo getPublic_levelName($row->public_level);?>
        </td>

    </tr>
    <?php
    $k = 1 - $k;
}
?>
</tbody>
</table>

<input type="hidden" name="task_id" value="<?php echo $this->taskId; ?>" />
<input type="hidden" name="task" value="" />

```

```

</form>
<br />
<a style="display:block; border:1px solid gray; text-align:center; width:
10em; padding: 5px; background-color: #E0E0E0 ; font-weight:bold;"
href="index.php?option=com_cimas&task_id=<?php echo $this->taskId;
?>&controller=task_comments&model=task_comment&task=edit">New Comment</a>

<br/><br/>
site/views/task_comments/tmpl/edit

<?php
defined('_JEXEC') or die('Restricted access');

// Your custom code here
    $task_comment = $this->task_comment;
?>

<form
action="index.php?option=com_cimas&controller=task_comments&model=task_comm
ent&task=save" method="post" name="adminForm">
<input type="hidden" name="id" value="<?php echo $task_comment->id; ?>" />

    <div class="col100">
    <fieldset class="adminform">
    <legend><?php echo JText::_('Details' ); ?></legend>

    <table class="admintable">
    <tbody>
    <tr>
        <td width="20%" class="key">
            <label for="title">
                Comment
            </label>
        </td>
        <td width="80%">
            <textarea class="inputbox" cols="70" rows="3"
name="comment" id="comment"><?php echo $task_comment->comment; ?>
</textarea>
        </td>
    </tr>
    <tr>
        <td width="20%" class="key">
            <label for="public_level">
                Public Level
            </label>
        </td>
        <td width="80%">
            <select name="public_level" id="public_level"
class="inputbox" size="1">
            <option value="1" selected="selected">All users</option>
            <option value="2">Manager-Employers</option>
            <option value="3">Manager</option>
            </select>
        </td>
    </tr>
    <tr>
        <td width="20%" class="key">
            <label>

```

```

                Creation date
            </label>
        </td>
        <td width="80%">
            <?php echo $task_comment->creation_date;?>
        </td>
    </tr>
    <tr>
        <td valign="middle" class="key" colspan="2"
        style="padding: 15px 0 0 15px;">
            <input type="submit" value="Submit" />
        </td>
    </tr>
</tbody>
</table>
</fieldset>
</div>
<div class="clr"></div>
<input type="hidden" name="id" value="<?php echo $task_comment->id;?>" />
<input type="hidden" name="task" value="" />
<input type="hidden" name="task_id" value="<?php echo $task_comment-
>task_id;?>" />
<input type="hidden" name="user_id" value="<?php echo $task_comment-
>user_id;?>" />
</form>

```

site/views/tasks/view

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.view');

class cimasViewTasks extends Jview {
    function display($tpl = null) {
        global $option;
        $issueId = JRequest::getVar( 'issue_id', '0' );

        $model = &$this->getModel();
        $list = $model->getList( $issueId );

        // global $mainframe;
        for ($i = 0; $i < count($list); $i++) {
            $row = & $list[$i];
            $row->link = JRoute :: _('index.php?option=' . $option .
            '&id=' . $row->id .
            '&controller=tasks&model=task&task=edit');
            $row->task_comments = JRoute :: _('index.php?option=' .
            $option . '&task_id=' . $row->id .
            '&controller=task_comments&model=task_comments');
        }

        $user = & JFactory :: getUser();

        $this->assignRef('user', $user);
        $this->assignRef('list', $list);
        $this->assignRef('issueId', $issueId);
        parent :: display($tpl);
    }
}

```

```

function edit($id = 0, $issueId = 0) {

    global $option;

    $model = &$this->getModel();

    $task = $model->getTask($id);
    if($id == 0){
        $task->issue_id = $issueId;
    }

    $this->assignRef('task', $task);

    parent :: display();

}

}
?>

site/views/tasks/tmpl/default

<?php
defined('_JEXEC') or die('Restricted access');

require_once (JPATH_COMPONENT_ADMINISTRATOR.DS.'helpers'.DS.'helper.php');

?>

<h3>Tasks</h3>
<form
action="index.php?option=com_cimas&controller=tasks&model=task"
method="post" name="adminForm">

<table class="adminlist" cellspacing="0" cellpadding="2" border="1">
<thead>
    <tr>
        <th width="5">
            <?php echo JText::_('Num'); ?>
        </th>
        <th class="title" width="25">
            <?php echo 'Title'; ?>
        </th>
        <th class="title" width="10%" align="center" nowrap="nowrap">
            <?php echo 'Assigned To'; ?>
        </th>
        <th align="center" width="10">
            <?php echo 'Start date'; ?>
        </th>
        <th align="center" width="10">
            <?php echo 'End date'; ?>
        </th>
        <th class="title" width="10%" align="center" nowrap="nowrap">
            <?php echo 'Status'; ?>
        </th>
        <th class="title" align="center" width="15%" nowrap="nowrap">
            <?php echo 'Issue'; ?>
        </th>
        <th class="title" width="10%" align="center" nowrap="nowrap">
            <?php echo 'Username'; ?>

```

```

        </th>
        <th class="title" width="15%" align="center" nowrap="nowrap">
            <?php echo 'Comments' ; ?>
        </th>
    </tr>
</thead>

<tbody>
<?php
$k = 0;
for ($i=0, $n=count( $this->list ); $i < $n; $i++)
{
    $row = $this->list[$i]; // XXX &

    $link      = '';
    ?>
    <tr class="<?php echo "row$k"; ?>">
        <td>
            <?php echo ($i+1); ?>
        </td>
        <td>
            <a href="<?php echo $row->link; ?>"><?php echo $row->title;
?></a>
        </td>
        <td align="center">
            <?php echo getAssigned_toName($row->assigned_to); ?>
        </td>
        <td nowrap="nowrap">
            <?php echo fdate($row->start_date); ?>
        </td>
        <td nowrap="nowrap">
            <?php echo fdate($row->end_date); ?>
        </td>
        <td align="center">
            <?php echo getStatusName($row->status); ?>
        </td>
        <td align="center">
            <?php echo $row->issue_title; ?>
        </td>
        <td align="center">
            <?php echo $row->username; ?>
        </td>
        <td align="center">
            <a href="<?php echo $row->task_comments; ?>">Comments</a>
        </td>
    </tr>
    <?php
    $k = 1 - $k;
}
?>
</tbody>
</table>

<input type="hidden" name="issue_id" value="<?php echo $this->issueId; ?>"
/>
<input type="hidden" name="task_comments" value="" />
</form>

<?php if($this->user->gid > 18 ){ ?>
<br />

```

```

<a style="display:block; border:1px solid gray; text-align:center; width:
10em; padding: 5px; background-color: #0000CC; color:#FFFFFF; font-
weight:bold;"
href="index.php?option=com_cimas&issue_id=<?php echo $this->issueId;
?>&controller=tasks&model=task&task=edit">New task</a>
<?php } ?>

```

```

<br />
<br />

```

site/views/tasks/tmpl/edit

```

<?php
defined('_JEXEC') or die('Restricted access');
require_once (JPATH_COMPONENT_ADMINISTRATOR.DS.'helpers'.DS.'helper.php');
// Your custom code here
    $task = $this->task;
?>

```

```

<form
action="index.php?option=com_cimas&controller=tasks&model=task&task=save"
method="post" name="adminForm">
    <input type="hidden" name="id" value="<?php echo $task->id; ?>" />

```

```

    <div class="col100">
    <fieldset class="adminform">
    <legend><?php echo JText::_('New Task' ); ?></legend>

```

```

    <table class="admintable">
    <tbody>
    <tr>
        <td width="20%" class="key">
            <label for="title">
                Title
            </label>
        </td>
        <td width="80%">
            <input class="inputbox" type="text" name="title" id="title" size="50"
            value="<?php echo $task->title;?>" />
        </td>
    </tr>

```

```

    <tr>
        <td width="20%" class="key">
            <label for="assigned_to">
                Assigned To
            </label>
        </td>
        <td width="80%">
            <?php echo getAssigned_toMenu($task->assigned_to);?>
        </td>
    </tr>

```

```

    <tr>
        <td width="20%" class="key">
            <label>
                Start date
            </label>
        </td>
        <td width="80%">
            <?php echo $task->start_date;?>
        </td>
    </tr>

```



```

        </td>
    </tr>
    <tr>
        <td width="20%" class="key">
            <label>
                End date
            </label>
        </td>
        <td width="80%">
            <?php echo $task->end_date
            ?>
        </td>
    </tr>
    <tr>
        <td width="20%" class="key">
            <label for="status">
                Status
            </label>
        </td>
        <td width="80%">
            <?php echo getStatusMenu($task->status); ?>
        </td>
    </tr>
    <tr>
        <td valign="top" class="key">
            <label for="content">
                Content
            </label>
        </td>
        <td>
            <textarea class="inputbox" cols="60" rows="3" name="tags"
            id="tags" value="<?php echo $task->content; ?>"></textarea>
        </td>
    </tr>
    <tr>
        <td valign="middle" class="key" colspan="2"
        style="padding: 15px 0 0 15px;">
            <input type="submit" value="Submit" />
        </td>
    </tr>
</tbody>
</table>
</fieldset>
</div>
<div class="clr"></div>
<input type="hidden" name="id" value="<?php echo $task->id; ?>" />
<input type="hidden" name="issue_id" value="<?php echo
$task->issue_id; ?>" />
<input type="hidden" name="user_id" value="<?php echo
$task->user_id; ?>" />

```

site/views/users/view

```
<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.view');

class cimasViewUsers extends Jview {
    function display($tpl = null) {
        global $option;

        $model = &$this->getModel();
    }

    function edit($userId = 0) {

        global $option;

        $model = &$this->getModel();
        $user = $model->getUser($userId);

        $this->assignRef('user', $user);

        parent :: display();
    }
}
?>
```

site/views/users/tmpl/edit

```
<?php
defined('_JEXEC') or die('Restricted access');

// Your custom code here
$user = $this->user;
?>
<form action="index.php?option=com_cimas&controller=users&model=user"
method="post" name="adminForm">
    <input type="hidden" name="id" value="<?php echo $issue->id; ?>" />

    <div class="col100">
        <fieldset class="adminform">
            <legend><?php echo JText::_('Details'); ?></legend>

            <table class="admintable">
                <tbody>
                    <tr>
                        <td width="20%" class="key">
                            <label>
                                Username
                            </label>
                        </td>
                        <td width="80%">
                            <?php echo $user->username; ?>
                        </td>
                    </tr>
                    <tr>
                        <td width="20%" class="key">
                            <label for="phone">
                                Phone

```

```

        </label>
    </td>
    <td width="80%">
        <input class="inputbox" type="text"
name="phone" id="phone" size="50" value="<?php echo $user->phone;?>" />
    </td>
</tr>
<tr>
    <td width="20%" class="key">
        <label for="address">
            Address
        </label>
    </td>
    <td width="80%">
        <input class="inputbox" type="text" name="address"
id="address" size="50" value="<?php echo $user->address;?>" />
    </td>
</tr>
<tr>
    <td width="20%" class="key">
        <label for="external_code">
            External code
        </label>
    </td>
    <td width="80%">
        <input class="inputbox" type="text"
name="external_code" id="external_code" size="50" value="<?php echo $user-
>external_code;?>" />
    </td>
</tr>
<tr>
    <td colspan="2" style="padding: 15px 0 0 15px;">
        <input type="submit" value="Submit" />
    </td>
</tr>
</tbody>
</table>
</fieldset>
</div>
<div class="clr"></div>
<input type="hidden" name="id" value="<?php echo $user->id;?>" />
<input type="hidden" name="task" value="save" />
<input type="hidden" name="user_id" value="<?php echo $user->user_id;?>" />
</form>

```

Administrator

Controllers

administrator/controllers/issues

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.controller');

class cimasControllerIssues extends JController {
    // Όνομα model
    // Όνομα του φακέλου που είναι το view
    // Όνομα του template == όνομα του task
    function display() {

```

```

$modelname = JRequest :: getVar('model', 'Issues');
$task = JRequest :: getVar('task', 'default');

$view = & $this->getView("Issues"); // Ψάξε στον φάκελο
views/issues

$model = & $this->getModel($modelname);
$view->setModel($model, true);

// Display the issues

switch ($task) {
    case 'add' :
    case 'edit' :
        $view->setLayout('edit'); // Φόρτωσε το
        $id = JRequest :: getVar('id', 0);
        $view->edit($id);
        break;
    case 'save':
        $this->do_save();
        $this->setRedirect('index.php?option=com_cimas',
'Issue saved.');
```

tpl/default.php

```

        break;
    case 'apply':
        $id = $this->do_save();
        $this->
>setRedirect('index.php?option=com_cimas&controller=issues&model=issue&task
=edit&id='.$id, 'Issue saved.');
```

'Issue saved.');

```

        break;
    case 'cancel' :
        $model = & $this->getModel('Issues');
        $view->setModel($model, true);
    default :
        $view->setLayout('default'); // Φόρτωσε το
        $view->display();
}
}

function do_save() {
    $row = & JTable :: getInstance('issue', 'Table');
    if (!$row->bind(JRequest :: get('post'))) {
        echo "<script> alert('" . $row->getError() . "')";
        window.history.go(-1); </script>\n";
        exit ();
    }

    if($row->user_id == 0){ // TODO αλλαγή όταν βάλουμε drop down
με τους χρηστές
        $user = & JFactory :: getUser();
        $row->user_id = $user->id;
    }

    if (!$row->store()) {
        echo "<script> alert('" . $row->getError() . "')";
        window.history.go(-1); </script>\n";
        exit ();
    }
}

```

```

        echo '<pre>';
        print_r( $row );
        echo '</pre>';

        return $row->id;
    }
}
?>
administrator/controllers/issue_comments
<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.controller');

class cimasControllerIssue_comments extends JController {
    // Όνομα model
    // Όνομα του φακέλου που είναι το view
    // Όνομα του template == όνομα του task
    function display() {
        $modelName = JRequest :: getVar('model', 'Issue_comments');
        $task = JRequest :: getVar('task', 'default');

        $view = & $this->getView("Issue_comments"); // Ψάξε στον φάκελο
views/issue_comments

        $model = & $this->getModel($modelName);
        $view->setModel($model, true);
        $issueId = JRequest :: getVar('issue_id', 0);

        // Display the task's comments

        switch ($task) {
            case 'add' :
            case 'edit' :
                $view->setLayout('edit'); // Φόρτωσε το
                $id = JRequest :: getVar('id', 0);
                $view->edit($id, $issueId);
                break;
            case 'save':
                $this->do_save();
                $this->
                >setRedirect('index.php?option=com_cimas&controller=issue_comments&model=issue_comments&issue_id='.$issueId, 'Comment for issue: saved.');
```

```

        $view->display();
    }
}

function do_save() {
    $row = & JTable :: getInstance('issue_comment', 'Table');
    if (!$row->bind(JRequest :: get('post'))) {
        echo "<script> alert('" . $row->getError() . "');
        window.history.go(-1); </script>\n";
        exit ();
    }

    if($row->user_id == 0
        $user = & JFactory :: getUser();
        $row->user_id = $user->id;
    }

    if (!$row->store()) {
        echo "<script> alert('" . $row->getError() . "');
        window.history.go(-1); </script>\n";
        exit ();
    }

    return $row->id;
}
}
?>

```

administrator/controllers/tasks

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.controller');

class cimasControllerTasks extends JController {
    // Όνομα model
    // Όνομα του φακέλου που είναι το view
    // Όνομα του template == όνομα του task
    function display() {
        $modelname = JRequest :: getVar('model', 'Tasks');
        $task = JRequest :: getVar('task', 'default');

        $view = & $this->getView("Tasks"); // Ψάξε στον φάκελο
views/issues

        $model = & $this->getModel($modelname);
        $view->setModel($model, true);
        $issueId = JRequest :: getVar('issue_id', 0);

        // Display the issues

        switch ($task) {
            case 'add' :
            case 'edit' :
                $view->setLayout('edit'); // Φόρτωσε το
tmpl/default.php

                $id = JRequest :: getVar('id', 0);
                $view->edit($id, $issueId);
                break;

```

```

        case 'save':
            $this->do_save();
            $this-
>setRedirect('index.php?option=com_cimas&controller=tasks&model=tasks&issue
_id='.$issueId, 'Task saved.');
```

```

            break;
        case 'apply':
            $id = $this->do_save();
            $this-
>setRedirect('index.php?option=com_cimas&controller=tasks&model=task&task=e
dit&id='.$id, 'Task saved.');
```

```

            break;

        case 'cancel' :
            $model = & $this->getModel('Tasks');
            $view->setModel($model, true);
        default :
            $view->setLayout('default'); // Φόρτωσε το
tmpl/default.php
            $view->display();
    }

}

function do_save() {
    $row = & JTable :: getInstance('task', 'Table');
    if (!$row->bind(JRequest :: get('post'))) {
        echo "<script> alert('" . $row->getError() . "')";
        window.history.go(-1); </script>\n";
        exit ();
    }

    if($row->user_id == 0){
        $user = & JFactory :: getUser();
        $row->user_id = $user->id;
    }

    if (!$row->store()) {
        echo "<script> alert('" . $row->getError() . "')";
        window.history.go(-1); </script>\n";
        exit ();
    }
    echo '<pre>';
    print_r( $row );
    echo '</pre>';

    return $row->id;
}

}
?>
```

administrator/controllers/task_comments

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.controller');

class cimasControllerTask_comments extends JController {
    // Όνομα model
    // Όνομα του φακέλου που είναι το view
```

```

// Όνομα του template == όνομα του task
function display() {
    $modelName = JRequest :: getVar('model', 'Task_comments');
    $task = JRequest :: getVar('task', 'default');

    $view = & $this->getView("Task_comments"); // Ψάξε στον φάκελο
views/task_comments

    $model = & $this->getModel($modelName);
    $view->setModel($model, true);
    $taskId = JRequest :: getVar('task_id', 0);

    switch ($task) {
        case 'add' :
        case 'edit' :
            $view->setLayout('edit'); // Φόρτωσε το
            $id = JRequest :: getVar('id', 0);
            $view->edit($id, $taskId);
            break;
        case 'save':
            $this->do_save();
            $this->
>setRedirect('index.php?option=com_cimas&controller=task_comments&model=task_comments&task_id='.$taskId, 'Tasks comment saved.');
```

tpl/edit.php

```

            break;
        case 'apply':
            $id = $this->do_save();
            $this->
>setRedirect('index.php?option=com_cimas&controller=task_comments&model=task_comment&task=edit&id='.$id, 'Tasks comment saved.');
```

tpl/default.php

```

            break;

        case 'cancel' :
            $model = & $this->getModel('Task_comments');
            $view->setModel($model, true);
        default :
            $view->setLayout('default'); // Φόρτωσε το
            $view->display();
    }
}

function do_save() {
    $row = & JTable :: getInstance('task_comment', 'Table');
    if (!$row->bind(JRequest :: get('post'))) {
        echo "<script> alert('" . $row->getError() . "')";
        window.history.go(-1); </script>\n";
        exit ();
    }

    if($row->user_id == 0){ // TODO αλλαγή όταν βάλουμε drop down
με τους χρηστές
        $user = & JFactory :: getUser();
        $row->user_id = $user->id;
    }

    if (!$row->store()) {
        echo "<script> alert('" . $row->getError() . "')";
        window.history.go(-1); </script>\n";
    }
}

```



```

        exit ();
    }

    return $row->id;
}

}
?>
administrator/controllers/users

<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.controller');

class cimasControllerUsers extends JController {
    // Όνομα model
    // Όνομα του φακέλου που είναι το view
    // Όνομα του template == όνομα του task
    function display() {
        $modelname = JRequest :: getVar('model', 'Users');
        $task = JRequest :: getVar('task', 'default');

        $view = & $this->getView("Users"); // Ψάξε στον φάκελο
views/users

        $model = & $this->getModel($modelname);
        $view->setModel($model, true);

        // Display the issues

        switch ($task) {
            case 'add' :
            case 'edit' :
                $view->setLayout('edit'); // Φόρτωσε το
tmpl/default.php

                $userId = JRequest :: getVar('user_id', 0);
                $view->edit($userId);
                break;
            case 'save':
                $this->do_save();
                $this->
>setRedirect('index.php?option=com_cimas&controller=users&model=users',
'User saved.');
```

```

                break;
            case 'apply':
                $id = $this->do_save();
                $this->
>setRedirect('index.php?option=com_cimas&controller=users&model=user&task=edit&user_id='.$id, 'User saved.');
```

```

                break;
            case 'cancel' :
                $model = & $this->getModel('Users');
                $view->setModel($model, true);
            default :
                $view->setLayout('default'); // Φόρτωσε το
tmpl/default.php

                $view->display();
        }
    }
}

```

```

    }

    function do_save() {
        $row = & JTable :: getInstance('user', 'Table');
        if (!$row->bind(JRequest :: get('post'))) {
            echo "<script> alert('" . $row->getError() . "');
            window.history.go(-1); </script>\n";
            exit ();
        }

        if (!$row->store()) {
            echo "<script> alert('" . $row->getError() . "');
            window.history.go(-1); </script>\n";
            exit ();
        }

        return $row->user_id;
    }
}
?>

```

Models

administrator/models/issues

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport( 'joomla.application.component.model' );

class cimasModelIssues extends Jmodel
{
    var $issues = null;
    var $_query = null;

    function __construct(){
        parent::__construct();
    }

    function getList()
    {
        global $mainframe;

        if(empty($this->issues))
        {
            $query = $this->_buildQuery();

            $this->issues = $this->_getList($query);
        }

        return $this->issues;
    }

    function _buildQuery()
    {
        if(!$this->_query)
        {

```

```

        $this->_query = "SELECT #__cimas_issue.id, title,
category, arrival_date, priority, 'status', attach, username";
        $this->_query .= " FROM #__cimas_issue, #__users";
        $this->_query .= " WHERE #__users.id =
#__cimas_issue.user_id ";
        $this->_query .= " ORDER BY id ";
    }

    return $this->_query;
}

```

```

}
?>

```

administrator/models/issue

```

<?php

```

```

defined('_JEXEC') or die('Restricted access');

```

```

jimport( 'joomla.application.component.model' );

```

```

class cimasModelIssue extends Jmodel

```

```

{

```

```

    var $issue = null;

```

```

    var $username = null;

```

```

    var $_query = null;

```

```

    function __construct(){
        parent::__construct();
    }

```

```

    function getIssue($id)

```

```

    {

```

```

        global $mainframe;

```

```

        if($id == 0){
            $this->issue =& JTable::getInstance('issue', 'Table');
        }

```

```

        if(empty($this->issue))

```

```

        {

```

```

            $query = $this->_buildQuery($id);

```

```

            $this->_db->setQuery( $query );

```

```

            $this->issue = $this->_db->loadObject();

```

```

            $user_id = $this->issue->user_id;

```

```

            $user_query = "SELECT username FROM #__users WHERE

```

```

`id`=".$user_id;

```

```

            $this->_db->setQuery( $query );

```

```

            $this->username = $this->_db->loadObject();

```

```

        }

```

```

        return $this->issue;

```

```

    }

```

```

    function _buildQuery($id)

```

```

        {
            if(!$this->_query)
            {
                $this->_query = "SELECT * FROM #__cimas_issue WHERE
id=".$id;
            }

            return $this->_query;
        }
    }
}
?>

```

administrator/models/issue_comments

```
<?php
```

```
defined('_JEXEC') or die('Restricted access');
```

```
jimport( 'joomla.application.component.model' );
```

```
class cimasModelIssue_comments extends JModel
```

```

{
    var $issue_comments = null;
    var $_query = null;

    function __construct(){
        parent::__construct();
    }

    function getList($issueId = 0)
    {
        global $mainframe;

        if(empty($this->issue_comments))
        {
            $query = $this->_buildQuery($issueId);

            $this->issue_comments = $this->_getList($query);
        }

        return $this->issue_comments;
    }

    function _buildQuery($issueId = 0)
    {
        if(!$this->_query)
        {
            $this->_query = "SELECT #__cimas_issue_comment.id,
#__cimas_issue.title, #__cimas_issue_comment.comment,
#__cimas_issue_comment.public_level,creation_date, issue_id, username";
            $this->_query .= " FROM #__cimas_issue_comment,
#__cimas_issue, #__users";
            $this->_query .= " WHERE #__cimas_issue_comment.issue_id
= #__cimas_issue.id AND #__users.id = #__cimas_issue_comment.user_id";
            if($issueId != 0){
                $this->_query .= " AND
#__cimas_issue_comment.issue_id = $issueId";
            }

            $this->_query .= " ORDER BY id ";
        }
    }
}

```

```

    }

    return $this->_query;
}

}
?>

administrator/models/issue_comment

<?php
defined('_JEXEC') or die('Restricted access');

jimport( 'joomla.application.component.model' );

class cimasModelIssue_comment extends Jmodel
{
    var $issue_comment = null;
    var $username = null;
    var $issue = null;
    var $_query = null;

    function __construct(){
        parent::__construct();
    }

    function getIssue_comment($id)
    {
        global $mainframe;

        if($id == 0){
            $this->issue_comment =&
JTable::getInstance('issue_comment', 'Table');
        }
        if(empty($this->issue_comment))
        {
            $query = $this->_buildQuery($id);

            $this->_db->setQuery( $query );
            $this->issue_comment = $this->_db->loadObject();

            $issue_id = $this->issue_comment->issue_id;
            $user_id = $this->issue_comment->user_id;

            $user_query = "SELECT username FROM #__users WHERE
`id`=" . $user_id;
            $this->_db->setQuery( $query );
            $this->username = $this->_db->loadObject();

        }

        return $this->issue_comment;
    }

    function _buildQuery($id)
    {
        if(!$this->_query)
        {
            $this->_query = "SELECT * FROM #__cimas_issue_comment
WHERE id=" . $id;

```

```

    }

    return $this->_query;
}

}
?>

administrator/models/tasks

<?php
defined('_JEXEC') or die('Restricted access');

jimport( 'joomla.application.component.model' );

class cimasModelTasks extends JModel
{
    var $tasks = null;
    var $_query = null;

    function __construct(){
        parent::__construct();
    }

    function getList($issueId = 0)
    {
        global $mainframe;

        if(empty($this->tasks))
        {
            $query = $this->_buildQuery($issueId);

            $this->tasks = $this->_getList($query);
        }

        return $this->tasks;
    }

    function _buildQuery($issueId = 0)
    {
        if(!$this->_query)
        {
            $this->_query = "SELECT #__cimas_task.id,
#__cimas_issue.title AS issue_title, #__cimas_task.title,
#__cimas_task.content, start_date, end_date, assigned_to,
#__cimas_task.status, issue_id, username";
            $this->_query .= " FROM #__cimas_task, #__cimas_issue,
#__users";
            $this->_query .= " WHERE #__cimas_task.issue_id =
#__cimas_issue.id AND #__users.id = #__cimas_task.user_id";
            if($issueId != 0){
                $this->_query .= " AND #__cimas_task.issue_id =
$issueId";
            }

            $this->_query .= " ORDER BY id ";
        }

        return $this->_query;
    }
}

```

```

}
?>
administrator/models/task
<?php
defined( '_JEXEC' ) or die( 'Restricted access' );

jimport( 'joomla.application.component.model' );

class cimasModelTask extends Jmodel
{
    var $task = null;
    var $username = null;
    var $issue = null;
    var $_query = null;

    function __construct(){
        parent::__construct();
    }

    function getTask($id)
    {
        global $mainframe;

        if($id == 0){
            $this->task =& JTable::getInstance( 'task', 'Table' );
        }
        if(empty($this->task))
        {
            $query = $this->_buildQuery($id);

            $this->_db->setQuery( $query );
            $this->task = $this->_db->loadObject();

            $issue_id = $this->task->issue_id;
            $user_id = $this->task->user_id;

            $user_query = "SELECT username FROM #__users WHERE
`id`=" . $user_id;
            $this->_db->setQuery( $query );
            $this->username = $this->_db->loadObject();

        }

        return $this->task;
    }

    function _buildQuery($id)
    {
        if(!$this->_query)
        {
            $this->_query = "SELECT * FROM #__cimas_task WHERE
id=" . $id;
        }

        return $this->_query;
    }
}

```

```

}
?>
administrator/models/task_comments

<?php
defined('_JEXEC') or die('Restricted access');

jimport( 'joomla.application.component.model' );

class cimasModelTask_comments extends JModel
{
    var $task_comments = null;
    var $_query = null;

    function __construct(){
        parent::__construct();
    }

    function getList($taskId = 0)
    {
        global $mainframe;

        if(empty($this->task_comments))
        {
            $query = $this->_buildQuery($taskId);

            $this->task_comments = $this->_getList($query);
        }

        return $this->task_comments;
    }

    function _buildQuery($taskId = 0)
    {
        if(!$this->_query)
        {
            $this->_query = "SELECT #__cimas_task_comment.id,
#__cimas_task.title, #__cimas_task_comment.comment,
#__cimas_task_comment.public_level,creation_date, task_id, username";
            $this->_query .= " FROM #__cimas_task_comment,
#__cimas_task, #__users";
            $this->_query .= " WHERE #__cimas_task_comment.task_id =
#__cimas_task.id AND #__users.id = #__cimas_task_comment.user_id";
            if($taskId != 0){
                $this->_query .= " AND
#__cimas_task_comment.task_id = $taskId";
            }

            $this->_query .= " ORDER BY id ";
        }

        return $this->_query;
    }
}
?>
administrator/models/task_comment

```



```

<?php
defined( '_JEXEC' ) or die( 'Restricted access' );

jimport( 'joomla.application.component.model' );

class cimasModelTask_comment extends Jmodel
{
    var $task_comment = null;
    var $username = null;
    var $task = null;
    var $_query = null;

    function __construct(){
        parent::__construct();
    }

    function getTask_comment($id)
    {
        global $mainframe;

        if($id == 0){
            $this->task_comment =&
JTable::getInstance( 'task_comment', 'Table' );
        }
        if(empty($this->task_comment))
        {
            $query = $this->_buildQuery($id);

            $this->_db->setQuery( $query );
            $this->task_comment = $this->_db->loadObject();

            $task_id = $this->task_comment->task_id;
            $user_id = $this->task_comment->user_id;

            $user_query = "SELECT username FROM #__users WHERE
`id`=" . $user_id;
            $this->_db->setQuery( $query );
            $this->username = $this->_db->loadObject();

            // TODO Comments
        }

        return $this->task_comment;
    }

    function _buildQuery($id)
    {
        if(!$this->_query)
        {
            $this->_query = "SELECT * FROM #__cimas_task_comment
WHERE id=" . $id;
        }

        return $this->_query;
    }
}
?>
administrator/models/users

```

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport( 'joomla.application.component.model' );

class cimasModelUsers extends Jmodel
{
    var $users = null;
    var $_query = null;

    function __construct(){
        parent::__construct();
    }

    function getList()
    {
        global $mainframe;

        if(empty($this->users))
        {
            $query = $this->_buildQuery();

            $this->users = $this->_getList($query);
        }

        return $this->users;
    }

    function _buildQuery()
    {
        if(!$this->_query)
        {
            $this->_query = "SELECT #__users.id, phone, address,
external_code, username, gid";
            $this->_query .= " FROM #__users LEFT JOIN
#__cimas_user";
            $this->_query .= " ON #__users.id = #__cimas_user.id ";
            $this->_query .= " ORDER BY id";
        }

        return $this->_query;
    }
}
?>

```

administrator/models/user

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport( 'joomla.application.component.model' );

class cimasModelUser extends Jmodel
{
    var $user = null;
    var $username = null;

    var $_query = null;

```

```

function __construct(){
    parent::__construct();
}

function getUser($userId)
{
    global $mainframe;

    if(empty($this->user))
    {
        $query = $this->_buildQuery($userId);

        $this->_db->setQuery( $query );
        $this->user = $this->_db->loadObject();

        if($this->user == null){ //
            $this->user =& JTable::getInstance('user',
'Table');

            $this->user->user_id = $userId;
        }

        $user_query = "SELECT username FROM #__users WHERE
`id`=" . $userId;
        $this->_db->setQuery( $user_query );
        $this->user->username = $this->_db->loadObject()-
>username;
    }

    return $this->user;
}

function _buildQuery($userId)
{
    if(!$this->_query)
    {
        $this->_query = "SELECT * FROM #__cimas_user WHERE
user_id=" . $userId;
    }

    return $this->_query;
}
}
?>

```

Helper

administrator/helpers/helper

```

<?php
defined('_JEXEC') or die('Restricted Access');

////////////////////////////////////
function getPriorityName($priorityNumber){
    $PRIORITY = array(
        1=>"Low",
        2=>"Normal",
        3=>"High",
        4=>"Very High"
    );
}

```

```

        foreach($PRIORITY as $key=>$value){
            if($key == $priorityNumber){
                return $value;
            }
        }
        return "None";
    }
}
////////////////////////////////////
function getAssigned_toName($assigned_toNumber){
    $ASSIGNED_TO = array(
        1=>"Rena Orfanoudaki",
        2=>"Eleni Androulaki",
        3=>"Antonis Lenidakis",
        4=>"George Fragkiadakis"
    );

    foreach($ASSIGNED_TO as $key=>$value){
        if($key == $assigned_toNumber){
            return $value;
        }
    }
    return "None";
}

////////////////////////////////////
function getStatusName($statusNumber){
    $STATUS = array(
        1=>"Pending assignment",
        2=>"Assigned",
        3=>"Testing",
        4=>"Solved"
    );

    foreach($STATUS as $key=>$value){
        if($key == $statusNumber){
            return $value;
        }
    }
    return "None";
}

////////////////////////////////////
function getCategoryName($categoryNumber){
    $CATEGORY = array(
        1=>"Service",
        2=>"Sales",
        3=>"Accountant",
        4=>"Other"
    );

    foreach($CATEGORY as $key=>$value){
        if($key == $categoryNumber){
            return $value;
        }
    }
    return "None";
}

////////////////////////////////////
function getPublic_levelName($public_levelNumber){
    $PUBLIC_LEVELNAME = array(
        1=>"All users",

```

```

        2=>"Manager-Employers",
        3=>"Manager",
    );

    foreach($PUBLIC_LEVELNAME as $key=>$value){
        if($key == $public_levelNumber){
            return $value;
        }
    }
    return "None";
}
////////////////////////////////////
function fdate($date){
    $time=strtotime($date);
    return date("d/m/Y, H:i", $time);
}
////////////////////////////////////
function getPriorityMenu($selected){

    $PRIORITY = array(
        1=>"Low",
        2=>"Normal",
        3=>"High",
        4=>"Very High"
    );

    $res = '<select name="priority" id="priority" class="inputbox"
size="1">';

    foreach($PRIORITY as $key=>$value){
        if($key == $selected){
            $res .= '<option value="'. $key. '"
selected="selected">'. $value. '</option>';
        }else{
            $res .= '<option value="'. $key. '">'. $value. '</option>';
        }
    }
    $res .= '</select>';
    return $res;
}
////////////////////////////////////
function getStatusMenu($selected){

    $STATUS = array(
        1=>"Pending assignment",
        2=>"Assigned",
        3=>"Testing",
        4=>"Solved"
    );

    $res = '<select name="status" id="status" class="inputbox"
size="1">';

    foreach($STATUS as $key=>$value){
        if($key == $selected){
            $res .= '<option value="'. $key. '"
selected="selected">'. $value. '</option>';
        }else{
            $res .= '<option value="'. $key. '">'. $value. '</option>';
        }
    }
    $res .= '</select>';
    return $res;
}

```

```

}

////////////////////////////////////
function getCategoryMenu($selected){

    $CATEGORY = array(
        1=>"Service",
        2=>"Sales",
        3=>"Accountant",
        4=>"Other"
    );
    $res = '<select name="category" id="category" class="inputbox"
size="1">';

    foreach($CATEGORY as $key=>$value){
        if($key == $selected){
            $res .= '<option value="' . $key . '"
selected="selected">' . $value . '</option>';
        }else{
            $res .= '<option value="' . $key . '">' . $value . '</option>';
        }
    }
    $res .= '</select>';
    return $res;
}

////////////////////////////////////
function getPublic_levelMenu($selected=1){

    $PUBLIC_LEVELNAME = array(
        1=>"All users",
        2=>"Manager-Employers",
        3=>"Manager",
    );
    $res = '<select name="public_level" id="public_level"
class="inputbox" size="1">';

    foreach($PUBLIC_LEVELNAME as $key=>$value){
        if($key == $selected){
            $res .= '<option value="' . $key . '"
selected="selected">' . $value . '</option>';
        }else{
            $res .= '<option value="' . $key . '">' . $value . '</option>';
        }
    }
    $res .= '</select>';
    return $res;
}

////////////////////////////////////
function getAssigned_toMenu($selected=1){

    $ASSIGNED_TONAME = array(
        1=>"Rena Orfanoudaki",
        2=>"Eleni Androulaki",
        3=>"Antonis Lenidakis",
        4=>"George Fragkiadakis"
    );
    $res = '<select name="assigned_to" id="assigned_to" class="inputbox"
size="1">';

    foreach($ASSIGNED_TONAME as $key=>$value){

```

```

        if($key == $selected){
            $res .= '<option value="'. $key. '"
selected="selected">'. $value. '</option>';
        }else{
            $res .= '<option value="'. $key. '">'. $value. '</option>';
        }
    }
    $res .= '</select>';
    return $res;
}
?>

```

Tables

administrator/tables/issue

```

<?php
defined('_JEXEC') or die('Restricted access');

class TableIssue extends JTable{
    var $id = 0;
    var $title = null;
    var $category = null;
    var $content = null;
    var $arrival_date;
    var $priority = null;
    var $address = null;
    var $phone = null;
    var $email = null;
    var $status = null;
    var $user_id = null;
    var $attach = null;

    function __construct(&$db)
    {
        parent::__construct('#__cimas_issue', 'id', $db);
        $this->arrival_date = date( 'Y-m-d H:i:s' );
    }
}
?>

```

administrator/tables/issue_comment

```

<?php
defined('_JEXEC') or die('Restricted access');

class TableIssue_comment extends JTable{
    var $id = null;
    var $comment = null;
    var $creation_date = null;
    var $user_id = null;
    var $issue_id = null;
    var $public_level = 1;

    function __construct(&$db)
    {
        parent::__construct('#__cimas_issue_comment', 'id', $db);
        $this->creation_date = date( 'Y-m-d H:i:s' );
    }
}

```

?>

administrator/tables/task

<?php

```
defined('_JEXEC') or die('Restricted access');
```

```
class TableTask extends JTable{
    var $id = null;
    var $title = null;
    var $content = null;
    var $start_date = null;
    var $end_date = null;
    var $assigned_to = null;
    var $status = null;
    var $issue_id = 0;
    var $user_id = null;

    function __construct(&$db)
    {
        parent::__construct('#__cimas_task', 'id', $db);
        $this->start_date = date( 'Y-m-d H:i:s' );
    }
}
```

?>

administrator/tables/task_comment

<?php

```
defined('_JEXEC') or die('Restricted access');
```

```
class TableTask_comment extends JTable{
    var $id = null;
    var $comment = null;
    var $creation_date = null;
    var $user_id = null;
    var $task_id = null;
    var $public_level = null;

    function __construct(&$db)
    {
        parent::__construct('#__cimas_task_comment', 'id', $db);
        $this->creation_date = date( 'Y-m-d H:i:s' );
    }
}
```

?>

administrator/tables/user

<?php

```
defined('_JEXEC') or die('Restricted access');
```

```
class TableUser extends JTable{
    var $id = 0;
    var $user_id = 0;
    var $phone = null;
    var $address = null;
```



```

var $external_code = null;

function __construct(&$db)
{
    parent::__construct('#__cimas_user', 'id', $db);
}
}
?>

administrator/views/issue_comments/view

<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.view');

class cimasViewIssue_comments extends Jview {
    function display($tpl = null) {
        global $option;
        $issueId = JRequest::getVar( 'issue_id', '0' );

        $model = &$this->getModel();
        $list = $model->getList( $issueId );

        // global $mainframe;
        for ($i = 0; $i < count($list); $i++) {
            $row = & $list[$i];
            $row->link = JRoute :: _('index.php?option=' . $option .
                '&id=' . $row->id .
                '&controller=issue_comments&model=issue_comment&task=edit');
        }

        $this->assignRef('list', $list);
        $this->assignRef('issueId', $issueId);
        parent :: display($tpl);
    }

    function edit($id = 0, $issueId = 0) {

        global $option;

        $model = &$this->getModel();

        $issue_comment = $model->getIssue_comment($id);
        if($id == 0){
            $issue_comment->issue_id = $issueId;
        }

        $this->assignRef('issue_comment', $issue_comment);

        parent :: display();
    }
}
?>

administrator/views/issue_comments/tmpl/default

```

```

<?php
defined( '_JEXEC' ) or die( 'Restricted access' );

// Your custom code here
require_once (JPATH_COMPONENT.DS.'helpers'.DS.'helper.php');

?>

<form
action="index.php?option=com_cimas&controller=issue_comments&model=
issue_comment" method="post" name="adminForm">

<table class="adminlist" cellspacing="1">
<thead>
  <tr>
    <th width="2%">
      <?php echo JText::_ ( 'Num' ); ?>
    </th>
    <th class="title" width="20%" nowrap="nowrap">
      <?php echo 'Comment'; ?>
    </th>
    <th align="center" width="10%">
      <?php echo 'Creation date'; ?>
    </th>
    <th class="title" width="20%" nowrap="nowrap">
      <?php echo 'Issue'; ?>
    </th>
    <th width="10%" class="title">
      <?php echo 'Username'; ?>
    </th>
    <th class="title" width="15%" nowrap="nowrap">
      <?php echo 'Access Level'; ?>
    </th>
  </tr>
</thead>

<tbody>
<?php
$k = 0;
for ( $i=0, $n=count( $this->list ); $i < $n; $i++)
{
  $row = $this->list[ $i ]; // XXX &

  $link      = '';
  ?>
  <tr class="<?php echo "row$k"; ?>">
    <td>
      <?php echo ( $i+1 ); ?>
    </td>
    <td>
      <a href="<?php echo $row->link; ?>"><?php echo $row-
>comment; ?></a>
    </td>
    <td nowrap="nowrap">
      <?php echo fdate( $row->creation_date ); ?>
    </td>
    <td align="center">
      <?php echo $row->title; ?>
    </td>
  </tr>
}
}

```

```

        <td align="center">
            <?php echo $row->username; ?>
        </td>
        <td align="center">
            <?php echo getPublic_levelName($row->public_level);?>
        </td>
    </tr>
    <?php
        $k = 1 - $k;
    }
?>
</tbody>
</table>

<input type="hidden" name="issue_id" value="<?php echo $this->issueId; ?>"
/>
<input type="hidden" name="task" value="" />
</form>

```

administrator/views/issue_comments/tmpl/edit

```

<?php
defined('_JEXEC') or die('Restricted access');
require_once (JPATH_COMPONENT.DS.'helpers'.DS.'helper.php');

// Your custom code here
    $issue_comment = $this->issue_comment;
?>

<form
action="index.php?option=com_cimas&controller=issue_comments&model=issue_co
mment&task=save" method="post" name="adminForm">
    <input type="hidden" name="id" value="<?php echo $issue_comment->id;
?>" />

<div class="col100">
    <fieldset class="adminform">
        <legend><?php 'Details'; ?></legend>

        <table class="admintable">
            <tbody>
                <tr>
                    <td width="20%" class="key">
                        <label for="comment">
                            Comment
                        </label>
                    </td>
                    <td width="80%">
                        <textarea class="inputbox" cols="70" rows="3"
name="comment" id="comment"><?php echo $issue_comment->comment;?>
</textarea>
                    </td>
                </tr>
                <tr>
                    <td width="20%" class="key">
                        <label for="public_level">
                            Access Level
                        </label>
                    </td>
                    <td width="80%">

```

```

        <?php echo getPublic_levelMenu($issue_comment-
>public_level);?>
        </td>
    </tr>
    <tr>
        <td width="20%" class="key">
            <label>
                Creation date
            </label>
        </td>
        <td width="80%">
            <?php echo $issue_comment->creation_date;?>
        </td>
    </tr>
</tbody>
</table>
</fieldset>
</div>
<div class="clr"></div>
<input type="hidden" name="id" value="<?php echo $issue_comment->id;?>" />
<input type="hidden" name="issue_comment" value="" />
<input type="hidden" name="issue_id" value="<?php echo $issue_comment-
>issue_id;?>" />
<input type="hidden" name="user_id" value="<?php echo $issue_comment-
>user_id;?>" />
<input type="hidden" name="task" value="" />
</form>

```

administrator/views/issues/view

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.view');

class CimasViewIssues extends Jview {
    function display($tpl = null) {
        global $option;

        $model = &$this->getModel();
        $list = $model->getList();

        // global $mainframe;
        for ($i = 0; $i < count($list); $i++) {
            $row = & $list[$i];
            $row->link = JRoute :: _('index.php?option=' . $option .
            '&id=' . $row->id . '&controller=issues&model=issue&task=edit');
            $row->task = JRoute :: _('index.php?option=' . $option .
            '&issue_id=' . $row->id . '&controller=tasks&model=tasks');
            $row->issue_comments = JRoute :: _('index.php?option=' .
            $option . '&issue_id=' . $row->id .
            '&controller=issue_comments&model=issue_comments');
            $row->attachment =
            '/Joomla/components/com_cimas/files/' . $row->attach;
        }

        $this->assignRef('list', $list);
        parent :: display($tpl);
    }
}

```

```

function edit($id = 0) {

    global $option;

    $model = &$this->getModel();
    $issue = $model->getIssue($id);

    $this->assignRef('issue', $issue);

    parent :: display();

}

```

```

}
?>

```

administrator/views/issues/tmpl/default

```

<?php
defined('_JEXEC') or die('Restricted access');
require_once (JPATH_COMPONENT.DS.'helpers'.DS.'helper.php');
?>
<form
action="index.php?option=com_cimas&controller=issues&model=issue"
method="post" name="adminForm">

<table class="adminlist" cellspacing="1">
<thead>
    <tr>
        <th width="5">
            <?php echo 'Num' ; ?>
        </th>
        <th class="title" width="15%" nowrap="nowrap">
            <?php echo 'Title'; ?>
        </th>
        <th class="title" width="15%" nowrap="nowrap">
            <?php echo 'Category'; ?>
        </th>
        <th align="center" width="10">
            <?php echo 'Arrival date'; ?>
        </th>
        <th class="title" width="15%" nowrap="nowrap">
            <?php echo 'Priority'; ?>
        </th>
        <th class="title" width="15%" nowrap="nowrap">
            <?php echo 'Status'; ?>
        </th>
        <th width="2%" class="title">
            <?php echo 'Username'; ?>
        </th>
        <th width="2%" class="title">
            <?php echo 'ID'; ?>
        </th>

        <th class="title" width="15%" nowrap="nowrap">
            <?php echo 'Action'; ?>
        </th>
        <th class="title" width="15%" nowrap="nowrap">
            <?php echo 'Comments'; ?>
        </th>
        <th class="title" width="15%" nowrap="nowrap">

```

```

                <?php echo 'Attachments'; ?>
            </th>
        </tr>
    </thead>

    <tbody>
        <?php
        $k = 0;
        for ($i=0, $n=count( $this->list ); $i < $n; $i++)
        {
            $row = $this->list[$i]; // XXX &

            $link      = '';
            $date = JHTML::_('date', $row->arrival_date,
JText::_('DATE_FORMAT_LC4') );
            ?>
            <tr class="<?php echo "row$k"; ?>">
                <td>
                    <?php echo ($i+1); ?>
                </td>
                <td align="center">
                    <a href="<?php echo $row->link; ?>"><?php echo $row-
>title; ?></a>
                </td>
                <td align="center">
                    <?php echo getCategoryName($row->category); ?>
                </td>
                <td nowrap="nowrap">
                    <?php echo $date; ?>
                </td>
                <td align="center">
                    <?php echo getPriorityName($row->priority); ?>
                </td>
                <td align="center">
                    <?php echo getStatusName($row->status); ?>
                </td>
                <td align="center">
                    <?php echo $row->username; ?>
                </td>
                <td>
                    <?php echo $row->id; ?>
                </td>

                <td align="center">
                    <a href="<?php echo $row->task; ?>">Task</a>
                </td>
                <td align="center">
                    <a href="<?php echo $row->issue_comments; ?>">Comments</a>
                </td>
                <td align="center">
                    <a href="<?php echo $row->attachment; ?>"><?php echo $row-
>attach; ?></a>
                </td>
            </tr>
            <?php
            $k = 1 - $k;
        }
    </tbody>
</table>

```

```

<input type="hidden" name="task" value="" />
<input type="hidden" name="issue_comments" value="" />
</form>

```

administrator/views/issues/tmpl/edit

```

<?php
defined('_JEXEC') or die('Restricted access');
require_once (JPATH_COMPONENT.DS.'helpers'.DS.'helper.php');

// Your custom code here
    $issue = $this->issue;
?>

<form
action="index.php?option=com_cimas&controller=issues&model=issue&task=save"
method="post" name="adminForm">
    <input type="hidden" name="id" value="<?php echo $issue->id; ?>" />

<div class="col100">
    <fieldset class="adminform">
        <legend><?php echo JText::_('Details'); ?></legend>

        <table class="admintable">
            <tbody>
                <tr>
                    <td width="20%" class="key">
                        <label for="title">
                            Title
                        </label>
                    </td>
                    <td width="80%">
                        <input class="inputbox" type="text" name="title"
id="title" size="50" value="<?php echo $issue->title; ?>"
/>
                    </td>
                </tr>
                <tr>
                    <td width="20%" class="key">
                        <label for="category">
                            Category
                        </label>
                    </td>
                    <td width="80%">
                        <?php echo getCategoryMenu($issue->category); ?>
                    </td>
                </tr>
                <tr>
                    <td width="20%" class="key">
                        <label>
                            Arrival date
                        </label>
                    </td>
                    <td width="80%">
                        <?php echo $issue->arrival_date; ?>
                    </td>
                </tr>
                <tr>
                    <td width="20%" class="key">

```

```

                <label for="priority">
                    Priority
                </label>
            </td>
            <td width="80%">
                <?php echo getPriorityMenu($issue->priority);?>
        </tr>
        <tr>
            <td width="20%" class="key">
                <label for="status">
                    Status
                </label>
            </td>
            <td width="80%">
                <?php echo getStatusMenu($issue->status);?>
        </tr>
        <tr>
            <td width="20%" class="key">
                <label for="status">
                    User
                </label>
            </td>
            <td width="80%">
                <?php echo JHTML::_('list.users', 'user_id',
$issue->user_id, 1, NULL, 'name', 0 );?>
            </td>
        </tr>
        <tr>
            <td width="20%" class="key">
                <label for="attach">
                    Attachment
                </label>
            </td>
            <td width="80%">
                <input class="inputbox" type="text"
name="attach" id="attach" size="50" value="<?php echo $issue->attach;?>" />
            </td>
        </tr>
        <tr>
            <td valign="top" class="key">
                <label for="content">
                    Content
                </label>
            </td>
            <td>
                <textarea class="inputbox" cols="70" rows="3"
name="tags" id="tags"><?php echo $issue->content;?></textarea>
            </td>
        </tr>
    </tbody>
</table>
</fieldset>
</div>
<div class="clr"></div>
<input type="hidden" name="id" value="<?php echo $issue->id;?>" />
<input type="hidden" name="task" value="" />
</form>

```


administrator/views/tasks/view

```
<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.view');

class cimasViewTasks extends Jview {
    function display($tpl = null) {
        global $option;
        $issueId = JRequest::getVar( 'issue_id', '0' );

        $model = &$this->getModel();
        $list = $model->getList( $issueId );

        // global $mainframe;
        for ($i = 0; $i < count($list); $i++) {
            $row = & $list[$i];
            $row->link = JRoute :: _('index.php?option=' . $option .
                '&id=' . $row->id .
                '&controller=tasks&model=task&task=edit');
            $row->task_comments = JRoute :: _('index.php?option=' .
                $option . '&task_id=' . $row->id .
                '&controller=task_comments&model=task_comments');
        }

        $this->assignRef('list', $list);
        $this->assignRef('issueId', $issueId);
        parent :: display($tpl);
    }

    function edit($id = 0, $issueId = 0) {

        global $option;

        $model = &$this->getModel();

        $task = $model->getTask($id);
        if($id == 0){
            $task->issue_id = $issueId;
        }

        $this->assignRef('task', $task);

        parent :: display();
    }
}
?>
```

administrator/views/tasks/tmpl/default

```
<?php
defined('_JEXEC') or die('Restricted access');

// Your custom code here
require_once (JPATH_COMPONENT.DS.'helpers'.DS.'helper.php');

?>
<form
action="index.php?option=com_cimas&controller=tasks&model=task"
method="post" name="adminForm">
```

```

<table class="adminlist" cellspacing="1">
<thead>
  <tr>
    <th width="5">
      <?php echo 'Num'; ?>
    </th>
    <th class="title">
      <?php echo 'Title'; ?>
    </th>
    <th class="title" width="15%" nowrap="nowrap">
      <?php echo 'Assigned to'; ?>
    </th>
    <th width="2%" class="title">
      <?php echo 'ID'; ?>
    </th>
    <th align="center" width="10">
      <?php echo 'Start date'; ?>
    </th>
    <th align="center" width="10">
      <?php echo 'End date'; ?>
    </th>
    <th class="title" width="15%" nowrap="nowrap">
      <?php echo 'Status'; ?>
    </th>
    <th class="title" width="15%" nowrap="nowrap">
      <?php echo 'Issue'; ?>
    </th>
    <th width="2%" class="title">
      <?php 'Username'; ?>
    </th>

    <th class="title" width="15%" nowrap="nowrap">
      <?php echo 'Comments'; ?>
    </th>

  </tr>
</thead>

<tbody>
<?php
$k = 0;
for ($i=0, $n=count( $this->list ); $i < $n; $i++)
{
  $row = $this->list[$i]; // XXX &

  $link      = '';
  ?>
  <tr class="<?php echo "row$k"; ?>">
    <td>
      <?php echo ($i+1); ?>
    </td>
    <td>
      <a href="<?php echo $row->link; ?>"><?php echo $row->title; ?></a>
    </td>
    <td align="center">
      <?php echo getAssigned_toName($row->assigned_to); ?>
    </td>
    <td>
      <?php echo $row->id; ?>

```

```

        </td>
        <td nowrap="nowrap">
            <?php echo fdate($row->start_date); ?>
        </td>
        <td nowrap="nowrap">
            <?php echo fdate($row->end_date); ?>
        </td>
        <td align="center">
            <?php echo getStatusName($row->status); ?>
        </td>
        <td align="center">
            <?php echo $row->issue_title; ?>
        </td>
        <td align="center">
            <?php echo $row->username; ?>
        </td>
        <td align="center">
            <a href="<?php echo $row->task_comments; ?>">Comments</a>
        </td>
    </tr>
    <?php
        $k = 1 - $k;
    }
?>
</tbody>
</table>

<input type="hidden" name="issue_id" value="<?php echo $this->issueId; ?>"
/>
<input type="hidden" name="task" value="" />
<input type="hidden" name="task_comments" value="" />
</form>

```

administrator/views/tasks/tmpl/edit

```

<?php
defined('_JEXEC') or die('Restricted access');
require_once (JPATH_COMPONENT.DS.'helpers'.DS.'helper.php');
// Your custom code here
    $task = $this->task;
?>

<form
action="index.php?option=com_cimas&controller=tasks&model=task&task=save"
method="post" name="adminForm">
<input type="hidden" name="id" value="<?php echo $task->id; ?>" />
<div class="col100">
    <fieldset class="adminform">
        <legend><?php echo JText::_('Task'); ?></legend>
        <table class="admintable">
            <tbody>
                <tr>
                    <td width="20%" class="key">
                        <label for="title">
                            Title
                        </label>
                    </td>
                    <td width="80%">

```

```

        <input class="inputbox" type="text"
        name="title" id="title" size="50"
        value="<?php echo $task->title;?>" />
    </td>
</tr>
<tr>
    <td width="20%" class="key">
        <label for="assigned_to">
            Assigned To
        </label>
    </td>
    <td width="80%">
        <?php echo getAssigned_toMenu($task->assigned_to);?>
    </td>
</tr>
<tr>
    <td width="20%" class="key">
        <label>
            Start date
        </label>
    </td>
    <td width="80%">
        <?php echo $task->start_date;?>
    </td>
</tr>
<tr>
    <td width="20%" class="key">
        <label>
            End date
        </label>
    </td>
    <td width="80%">
        <?php echo $task->end_date;?>
    </td>
</tr>
<tr>
    <td width="20%" class="key">
        <label for="status">
            Status
        </label>
    </td>
    <td width="80%">
        <?php echo getStatusMenu($task->status);?>
    </td>
</tr>
<tr>
    <td valign="top" class="key">
        <label for="content">
            Content
        </label>
    </td>
    <td>
        <textarea class="inputbox" cols="70" rows="3"
        name="tags" id="tags"><?php echo
        $task->content;?></textarea>
    </td>
</tr>

```

```

        </tr>
</tbody>
</table>
</fieldset>
</div>
<div class="clr"></div>
<input type="hidden" name="id" value="<?php echo $task->id;?>" />
<input type="hidden" name="task" value="" />
<input type="hidden" name="issue_id" value="<?php echo
    $task->issue_id;?>" />
<input type="hidden" name="user_id" value="<?php echo
    $task->user_id;?>" />
</form>

```

administrator/views/task_comments/view

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.view');

class cimasViewTask_comments extends Jview {
    function display($tpl = null) {
        global $option;
        $taskId = JRequest::getVar( 'task_id', '0' );

        $model = &$this->getModel();
        $list = $model->getList( $taskId );

        // global $mainframe;
        for ($i = 0; $i < count($list); $i++) {
            $row = & $list[$i];
            $row->link = JRoute :: _('index.php?option=' . $option .
                '&id=' . $row->id .
                '&controller=task_comments&model=task_comment&task=edit');
        }

        $this->assignRef('list', $list);
        $this->assignRef('taskId', $taskId);
        parent :: display($tpl);
    }

    function edit($id = 0, $taskId = 0) {

        global $option;

        $model = &$this->getModel();

        $task_comment = $model->getTask_comment($id);
        if($id == 0){
            $task_comment->task_id = $taskId;
        }

        $this->assignRef('task_comment', $task_comment);

        parent :: display();
    }
}
?>

```

administrator/views/task_comments/tmpl/default

```

<?php
defined('_JEXEC') or die('Restricted access');

// Your custom code here
require_once (JPATH_COMPONENT.DS.'helpers'.DS.'helper.php');

// print "<pre>";
//print_r($this->list);
// print "</pre>";
?>

<form
action="index.php?option=com_cimas&controller=task_comments&model=t
ask_comment" method="post" name="adminForm">

<table class="adminlist" cellspacing="1">
<thead>
  <tr>
    <th width="5">
      <?php echo 'Num' ; ?>
    </th>
    <th width="40%" class="title">
      <?php echo 'Comment'; ?>
    </th>
    <th align="center" width="10">
      <?php echo 'Creation date'; ?>
    </th>
    <th class="title" width="15%" nowrap="nowrap">
      <?php echo 'Task'; ?>
    </th>
    <th width="2%" class="title">
      <?php echo 'Username'; ?>
    </th>
    <th class="title" width="15%" nowrap="nowrap">
      <?php echo 'Access Level' ; ?>
    </th>
  </tr>
</thead>

<tbody>
<?php
$k = 0;
for ($i=0, $n=count( $this->list ); $i < $n; $i++)
{
  $row = $this->list[$i]; // XXX &

  $link      = '';
  ?>
  <tr class="<?php echo "row$k"; ?>">
    <td>
      <?php echo ($i+1); ?>
    </td>
    <td>
      <a href="<?php echo $row->link; ?>"><?php echo
      $row->comment; ?></a>
    </td>
    <td nowrap="nowrap">
      <?php echo fdate($row->creation_date); ?>
    </td>
  </tr>
}
}

```

```

        <td align="center">
            <?php echo $row->title;?>
        </td>
        <td align="center">
            <?php echo $row->username; ?>
        </td>
        <td align="center">
            <?php echo getPublic_levelName($row->public_level);?>
        </td>
    </tr>
    <?php
    $k = 1 - $k;
}
?>
</tbody>
</table>

```

```

<input type="hidden" name="task_id" value="<?php echo $this->taskId; ?>" />
<input type="hidden" name="task" value="" />
</form>

```

administrator/views/task_comments/tmpl/edit

```

<?php
defined('_JEXEC') or die('Restricted access');
require_once (JPATH_COMPONENT.DS.'helpers'.DS.'helper.php');

// Your custom code here
    $task_comment = $this->task_comment;

//     print "<pre>";
//     print_r($task_comment);
//     print "</pre>";
?>

<form
action="index.php?option=com_cimas&controller=task_comments&model=task_comm
ent&task=save" method="post" name="adminForm">
    <input type="hidden" name="id" value="<?php echo $task_comment->id;
?>" />

        <div class="col100">
            <fieldset class="adminform">
                <legend><?php echo JText::_('Details' );
?></legend>
<table class="admintable">
    <tbody>
        <tr>
            <td width="20%" class="key">
                <label for="title">
                    Comment
                </label>
            </td>
            <td width="80%">
                <textarea class="inputbox" cols="70" rows="3"
name="comment" id="comment"><?php echo $task_comment-
>comment;?> </textarea>
            </td>
        </tr>
        <tr>

```

```

        <td width="20%" class="key">
            <label for="public_level">
                Access Level
            </label>
        </td>
        <td width="80%">
            <?php echo getPublic_levelMenu
            ($task_comment->public_level);?>
        </td>
    </tr>
</tr>
<tr>
    <td width="20%" class="key">
        <label>
            Creation date
        </label>
    </td>
    <td width="80%">
        <?php echo $task_comment->creation_date;?>
    </td>
</tr>
</tbody>
</table>
</fieldset>
</div>
    <div class="clr"></div>
    <input type="hidden" name="id" value="<?php echo
    $task_comment->id;?>" />
    <input type="hidden" name="task" value="" />
    <input type="hidden" name="task_id" value="<?php echo
    $task_comment->task_id;?>" />
    <input type="hidden" name="user_id" value="<?php echo
    $task_comment->user_id;?>" />
</form>

```

administrator/views/users/view

```

<?php
defined('_JEXEC') or die('Restricted access');

jimport('joomla.application.component.view');

class cimasViewUsers extends Jview {
    function display($tpl = null) {
        global $option;

        $model = &$this->getModel();
        $list = $model->getList();

        // global $mainframe;
        for ($i = 0; $i < count($list); $i++) {
            $row = & $list[$i];
            $row->link = JRoute :: _('index.php?option=' . $option .
            '&user_id=' . $row->id . '&controller=users&model=user&task=edit');
        }

        $this->assignRef('list', $list);
        parent :: display($tpl);

    }

    function edit($userId = 0) {

```



```

        global $option;

        $model = &$this->getModel();
        $user = $model->getUser($userId);

        $this->assignRef('user', $user);

        parent :: display();
    }
}
?>
administrator/views/users/tmpl/default
<?php
defined('_JEXEC') or die('Restricted access');

?>

<form
action="index.php?option=com_cimas&amp;controller=issues&amp;model=issue"
method="post" name="adminForm">

<table class="adminlist" cellspacing="1">
<thead>
    <tr>
        <th width="5">
            <?php echo 'Num'; ?>
        </th>
        <th class="title">
            <?php echo 'Username'; ?>
        </th>
        <th width="2%" class="title">
            <?php echo 'GID'; ?>
        </th>
        <th align="center" width="10">
            <?php echo 'Phone'; ?>
        </th>
        <th class="title" width="15%" nowrap="nowrap">
            <?php echo 'Address'; ?>
        </th>
        <th class="title" width="15%" nowrap="nowrap">
            <?php echo 'External code'; ?>
        </th>
        <th width="2%" class="title">
            <?php echo 'ID'; ?>
        </th>
    </thead>

    <tbody>
        <?php
        $k = 0;
        for ($i=0, $n=count( $this->list ); $i < $n; $i++)
        {
            $row = $this->list[$i]; // XXX &

            $link          = '';
            ?>
            <tr class="<?php echo "row$k"; ?>">
                <td>
                    <?php echo ($i+1); ?>

```

```

        </td>
        <td nowrap="nowrap">
        <a href="<?php echo $row->link; ?>"><?php echo $row->username;
?></a>
        </td>
        <td>
                <?php echo $row->gid; ?>
        </td>
        <td align="center">
                <?php echo $row->phone; ?>
        </td>
        <td align="center">
                <?php echo $row->address; ?>
        </td>
        <td align="center">
                <?php echo $row->external_code; ?>
        </td>
        <td>
                <?php echo $row->id; ?>
        </td>
    </tr>
    <?php
    $k = 1 - $k;
}
?>
</tbody>
</table>

<input type="hidden" name="task" value="" />
</form>

```

administrator/views/users/tmpl/edit

```

<?php
defined('_JEXEC') or die('Restricted access');

$user = $this->user;
?>

<form action="index.php?option=com_cimas&controller=users&model=users"
method="post" name="adminForm">
    <input type="hidden" name="id" value="<?php echo $issue->id; ?>" />

    <div class="col100">
        <fieldset class="adminform">
            <legend><?php echo JText::_('Details'); ?>
            </legend>

<table class="admintable">
    <tbody>
        <tr>
            <td width="20%" class="key">
                <label>
                    Username
                </label>
            </td>
            <td width="80%">
                <?php echo $user->username; ?>
            </td>
        </tr>
    </tbody>
</table>

```

```

        <tr>
            <td width="20%" class="key">
                <label for="phone">
                    Phone
                </label>
            </td>
            <td width="80%">
                <input class="inputbox" type="text" name="phone" id="phone"
size="50" value="<?php echo $user->phone;?>" />
            </td>
        </tr>

        <tr>
            <td width="20%" class="key">
                <label for="address">
                    Address
                </label>
            </td>
            <td width="80%">
                <input class="inputbox" type="text" name="address"
id="address" size="50" value="<?php echo $user->address;?>" />
            </td>
        </tr>

        <tr>
            <td width="20%" class="key">
                <label for="external_code">
                    External code
                </label>
            </td>
            <td width="80%">
                <input class="inputbox" type="text" name="external_code"
id="external_code" size="50" value="<?php echo $user->external_code;?>" />
            </td>
        </tr>

</tbody>
</table>
</fieldset>
</div>
<div class="clr"></div>
    <input type="hidden" name="id" value="<?php echo $user->id;?>" />
    <input type="hidden" name="task" value="" />
    <input type="hidden" name="user_id" value="<?php echo $user-
>user_id;?>" />
</form>

```

administrator/admin.cimas

```

<?php
defined('_JEXEC') or die('Restricted access');

// Require the base controller
require_once (JPATH_COMPONENT.DS.'controller.php');
JTable::addIncludePath(JPATH_COMPONENT.DS.'tables');

// Require specific controller if requested
$controller = JRequest::getVar( 'controller', 'Issues' );

require_once(JPATH_COMPONENT.DS.'controllers'.DS.$controller.'.php');

```

```
// Create the controller
$controller = 'CimasController'.$controller;
$controller = new $controller();

$controller->execute(JRequest::getCmd('task', 'issues' ));
// redirect
$controller->redirect();
?>
```

administrator/install.mysql.utf8.sql

```
DROP TABLE IF EXISTS `#__cimas_issue`;
CREATE TABLE `#__cimas_issue`(
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `title` varchar(50) NOT NULL default '',
  `category` varchar(20) NOT NULL default '',
  `content` text NOT NULL default '',
  `arrival_date` datetime NOT NULL default '0000-00-00 00:00:00',
  `priority` varchar(20) NOT NULL default '',
  `status` varchar(20) NOT NULL default '',
  `user_id` int(11) NOT NULL default 0,
  `attach` VARCHAR( 250 ) NOT NULL ,
  PRIMARY KEY (`id`)
) ENGINE= MyISAM AUTO_INCREMENT=0 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `#__cimas_task`;
CREATE TABLE `#__cimas_task`(
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `title` varchar(50) NOT NULL default '',
  `content` text NOT NULL default '',
  `start_date` datetime NOT NULL default '0000-00-00 00:00:00',
  `end_date` datetime NOT NULL default '0000-00-00 00:00:00',
  `assigned_to` varchar(255) default '',
  `status` varchar(20) default '',
  `issue_id` int(11) NOT NULL default 0,
  `user_id` int(11) NOT NULL default 0,
  PRIMARY KEY (`id`)
) ENGINE= MyISAM AUTO_INCREMENT=0 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `#__cimas_issue_comment`;
CREATE TABLE `#__cimas_issue_comment`(
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `comment` text NOT NULL default '',
  `creation_date` datetime NOT NULL default '0000-00-00 00:00:00',
  `user_id` int(11) NOT NULL default 0,
  `issue_id` int(11) NOT NULL default 0,
  `public_level` varchar(20) NOT NULL default '',
  PRIMARY KEY (`id`)
) ENGINE= MyISAM AUTO_INCREMENT=0 DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `#__cimas_task_comment`;
CREATE TABLE `#__cimas_task_comment`(
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `comment` text NOT NULL default '',
  `creation_date` datetime NOT NULL default '0000-00-00 00:00:00',
  `user_id` int(11) NOT NULL default 0,
  `task_id` int(11) NOT NULL default 0,
  `public_level` varchar(20) NOT NULL default '',
  PRIMARY KEY (`id`)
) ENGINE= MyISAM AUTO_INCREMENT=0 DEFAULT CHARSET=utf8;
```

```

DROP TABLE IF EXISTS `#__cimas_user`;
CREATE TABLE `#__cimas_user` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `user_id` INT NOT NULL ,
  `phone` VARCHAR( 50 ) NOT NULL ,
  `address` VARCHAR( 250 ) NOT NULL ,
  `external_code` INT NOT NULL ,
  PRIMARY KEY ( `id` )
) ENGINE = MYISAM DEFAULT CHARSET=utf8;

```

administrator/toolbar.cimas.html

```

<?php
defined ( '_JEXEC' ) or die ( 'Restricted access' );
class TOOLBAR_cimas
{
    function _NEW()
    {
        JToolBarHelper::save();
        JToolBarHelper::apply();
        JToolBarHelper::cancel();
    }
    function _DEFAULT()
    {
        JToolBarHelper::title ( JText::_ ( 'CIMaS' ), 'generic.png' );
        JToolBarHelper::addNew();
        JToolBarHelper::back();
    }
}
?>

```

administrator/toolbar.cimas

```

<?php
defined ( '_JEXEC' ) or die ( 'Restricted access' );
require_once ( JApplicationHelper::getPath ( 'toolbar_html' ) );
switch ( $task )
{
    case 'edit':
    case 'add':
        TOOLBAR_cimas::_NEW();
        break;
    default:
        TOOLBAR_cimas::_DEFAULT();
        break;
}
?>

```

Το XML αρχείο εγκατάστασης

```

<?xml version="1.0" encoding="utf-8"?>
<install type="component" version="1.5.0" method="upgrade">
  <name>com_cimas</name>
  <author>Rena</author>
  <creationDate>29/10/2010</creationDate>
  <license>GNU/GPL</license>
  <version>1.0</version>
  <description>This component is an issue tracking system.</description>

```

```

<!-- user part -->
<files folder="components/com_cimas">
  <filename>cimas.php</filename>
    <filename>controller.php</filename>
    <filename>controllers/index.html</filename>
    <filename>controllers/users.php</filename>
    <filename>controllers/tasks.php</filename>
    <filename>controllers/task_comments.php</filename>
    <filename>controllers/issues.php</filename>
    <filename>controllers/history.php</filename>
    <filename>models/index.html</filename>
    <filename>models/issue.php</filename>
    <filename>models/issues.php</filename>
    <filename>models/issue_comment.php</filename>
    <filename>models/issue_comments.php</filename>
    <filename>models/task.php</filename>
    <filename>models/tasks.php</filename>
    <filename>models/task_comment.php</filename>
    <filename>models/task_comments.php</filename>
    <filename>models/user.php</filename>
    <filename>models/history.php</filename>
    <filename>tables/index.html</filename>
    <filename>views/issues/view.php</filename>
    <filename>views/index.html</filename>
    <filename>views/issues/index.html</filename>
    <filename>views/issues/tmpl/default.php</filename>
    <filename>views/issues/tmpl/edit.php</filename>
    <filename>views/issues/tmpl/index.html</filename>
    <filename>views/issue_comments/view.php</filename>
    <filename>views/issue_comments/index.html</filename>
    <filename>views/issue_comments/tmpl/default.php</filename>
    <filename>views/issue_comments/tmpl/edit.php</filename>
    <filename>views/issue_comments/tmpl/index.html</filename>
    <filename>views/task_comments/view.php</filename>
    <filename>views/task_comments/index.html</filename>
    <filename>views/task_comments/tmpl/default.php</filename>
    <filename>views/task_comments/tmpl/edit.php</filename>
    <filename>views/task_comments/tmpl/index.html</filename>
    <filename>views/tasks/view.php</filename>
    <filename>views/tasks/index.html</filename>
    <filename>views/tasks/tmpl/default.php</filename>
    <filename>views/tasks/tmpl/edit.php</filename>
    <filename>views/tasks/tmpl/index.html</filename>
    <filename>views/users/view.php</filename>
    <filename>views/users/index.html</filename>
    <filename>views/users/tmpl/edit.php</filename>
    <filename>views/users/tmpl/index.html</filename>
    <filename>views/history/view.php</filename>
    <filename>views/history/index.html</filename>
    <filename>views/history/tmpl/default.php</filename>
    <filename>views/history/tmpl/index.html</filename>
    <filename>index.html</filename>
  </files>

  <languages folder="language/en-GB">
    <language tag="en-GB">en-GB.com_com_cimas.ini</language>
  </languages>

  <!-- administration part -->
  <administration>

```

```

<menu task="default"
img="js/ThemeOffice/component.png">com_cimas</menu>
<submenu>
  <menu link="option=com_com_cimas">com_cimas</menu>
</submenu>

<files>
  <filename>com_cimas.xml</filename>
</files>
<files folder="administrator/components/com_cimas">
  <filename>admin.cimas.php</filename>
    <filename>controller.php</filename>
    <filename>controllers/index.html</filename>
    <filename>controllers/users.php</filename>
    <filename>controllers/tasks.php</filename>
    <filename>controllers/task_comments.php</filename>
    <filename>controllers/issues.php</filename>
    <filename>controllers/issue_comments.php</filename>
    <filename>helpers/helper.php</filename>
    <filename>helpers/index.html</filename>
    <filename>models/index.html</filename>
    <filename>models/issue.php</filename>
    <filename>models/issues.php</filename>
    <filename>models/issue_comment.php</filename>
    <filename>models/issue_comments.php</filename>
    <filename>models/task.php</filename>
    <filename>models/tasks.php</filename>
    <filename>models/task_comment.php</filename>
    <filename>models/task_comments.php</filename>
    <filename>models/user.php</filename>
    <filename>models/users.php</filename>
    <filename>tables/index.html</filename>
    <filename>tables/issue.php</filename>
    <filename>tables/issue_comment.php</filename>
    <filename>tables/task.php</filename>
    <filename>tables/task_comment.php</filename>
    <filename>tables/user.php</filename>
    <filename>views/issues/view.php</filename>
    <filename>views/index.html</filename>
    <filename>views/issues/index.html</filename>
    <filename>views/issues/tmpl/default.php</filename>
    <filename>views/issues/tmpl/edit.php</filename>
    <filename>views/issues/tmpl/index.html</filename>
    <filename>views/issue_comments/view.php</filename>
    <filename>views/issue_comments/index.html</filename>

  <filename>views/issue_comments/tmpl/default.php</filename>
    <filename>views/issue_comments/tmpl/edit.php</filename>
    <filename>views/issue_comments/tmpl/index.html</filename>
    <filename>views/task_comments/view.php</filename>
    <filename>views/task_comments/index.html</filename>
    <filename>views/task_comments/tmpl/default.php</filename>
    <filename>views/task_comments/tmpl/edit.php</filename>
    <filename>views/task_comments/tmpl/index.html</filename>
    <filename>views/tasks/view.php</filename>
    <filename>views/tasks/index.html</filename>
    <filename>views/tasks/tmpl/default.php</filename>
    <filename>views/tasks/tmpl/edit.php</filename>
    <filename>views/tasks/tmpl/index.html</filename>
    <filename>views/users/view.php</filename>

```

```
<filename>views/users/index.html</filename>
<filename>views/users/tmpl/default.php</filename>
<filename>views/users/tmpl/edit.php</filename>
<filename>views/users/tmpl/index.html</filename>
<filename>index.html</filename>
<filename>install.mysql.utf8.sql</filename>
<filename>toolbar.cimas.html.php</filename>
<filename>toolbar.cimas.php</filename>
</files>

</administration>

<params>

</params>
```


Παράρτημα Β: Εγκατάσταση CIMaS component

Το CIMaS όπως έχει αναφερθεί αρκετές φορές στην παρούσα πτυχιακή εργασία, είναι ένα component για το CMS Joomla!

Για να εγκατασταθεί λοιπόν το CIMaS, τα βήματα που ακολουθούνται είναι τα εξής:

Είσοδος στο backend → extentions → install → αναζήτηση αρχείου (com_cimas.zip) → upload file & install.

Παράρτημα Γ: Παρουσίαση Πτυχιακής Εργασίας (Διαφάνειες)

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
Σχολή Τεχνολογικών εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Θέμα Πτυχιακής Εργασίας

ONLINE ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΖΗΤΗΜΑΤΩΝ
(ONLINE ISSUE MANAGEMENT SYSTEM)



Σπουδάστρια:
Ορφανουδάκη Ειρήνη

Επόπτης Καθηγητής:
κ. Παπαδουράκης Γεωργιος

Συστήματα Διαχείρισης Περιεχομένου

- Τι είναι CMS;
- Joomla!
- Drupal
- Wordpress
- Σύγκριση Μεταξύ των Παραπάνω CMSs

Συστήματα Διαχείρισης Ζητημάτων

- Τι είναι Issue Tracking System
- Πως Εξυπηρετεί κάποια επιχείρηση
- Ροή Εργασίας σε Issue Tracking System
- Παραδείγματα:
 - SysAid
 - HuruHelpdesk

CIMaS

- Τι είναι το CIMaS
- Τρόπος Υλοποίησης
 - Model
 - View
 - Controller

CIMaS - Απαιτήσεις Πελάτη

- Δημιουργία λογαριασμού χρήσης
- Προβολή και επεξεργασία στοιχείων
- Δυνατότητα αποστολής και λήψης αρχείων ενδιαφέροντος
- Ευχρηστία συστήματος
- Απλότητα στην περιήγηση
- Λακωνικότητα στη συμπλήρωση δεδομένων – πληροφοριών
- Απόδοση σπουδαιότητας
- Ενημέρωση για εργασίες που είναι ανοιχτές
- Ενημέρωση για το status του αιτήματος του
- Δυνατότητα εμφάνισης ιστορικού

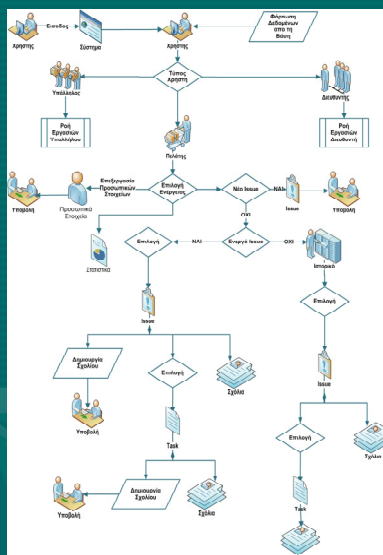
CIMaS - Απαιτήσεις Υπαλλήλου

- Έγκαιρη ενημέρωση για Issue
- Δυνατότητα εμφάνισης στοιχείων πελάτη (π.χ. ιστορικού κ.α)
- Να εμφανίζονται με προκαθορισμένο από το σύστημα τρόπο οι πληροφορίες που έχει δώσει ο πελάτης για το Issue
- Να έχει τη δυνατότητα να ενημερώνει για εργασίες που έχει κάνει για τα Issues και το Tasks
- Να έχει τη δυνατότητα να ενημερώνει το status του Issue
- Να μπορεί να δημιουργήσει νέο μήνυμα για κάποιο πελάτη
- Να ενημερώνει το σύστημα για το κόστος

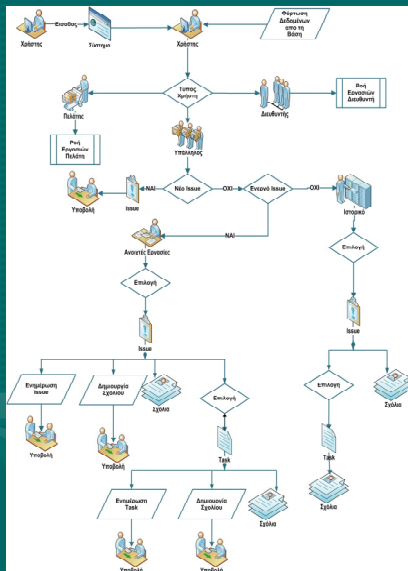
CIMaS - Απαιτήσεις Διευθυντή

- Δυνατότητα διασύνδεσης συστήματος με άλλα συστήματα καταγραφής κινήσεων πελατών
- Έγκαιρη ενημέρωση για Issue
- Δυνατότητα εμφάνισης στοιχείων πελάτη (π.χ. ιστορικού κ.α)
- Δυνατότητα διάσπασης Issue σε Tasks
- Δυνατότητα ανάθεσης Task σε συγκεκριμένο υπάλληλο
- Δυνατότητα ενημέρωσης για την ικανοποίηση του πελάτη
- Διαχείριση χρηστών
- Δυνατότητα επεξεργασίας περιεχομένου Issue και Task
- Να μπορεί να δημιουργήσει νέο μήνυμα για κάποιον πελάτη
- Ύπαρξη πεδίου για πληροφορίες / συμβουλές προς τους υπαλλήλους

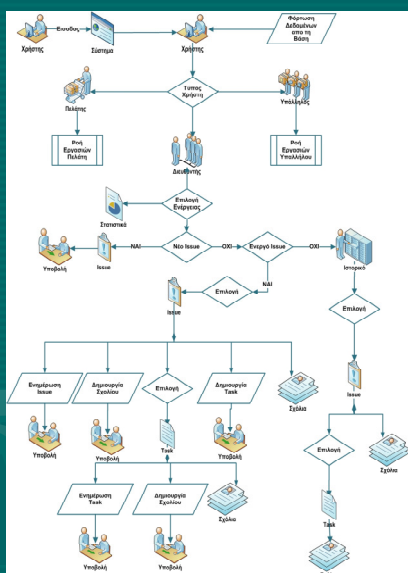
CIMaS Διάγραμμα Ροής Εργασιών Πελάτη



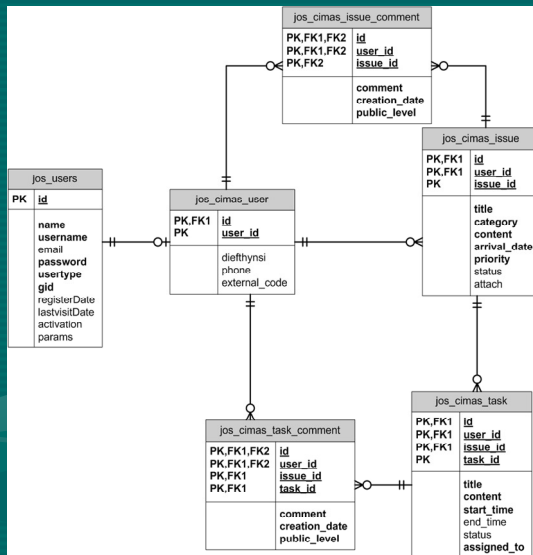
CIMaS Διάγραμμα Ροής Εργασιών Υπαλλήλου



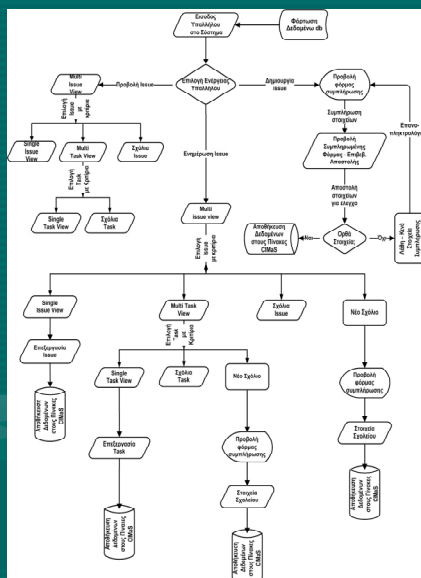
CIMaS Διάγραμμα Ροής Εργασιών Διευθυντή



CIMaS Διάγραμμα Βάσης Δεδομένων



CIMaS Παράδειγμα Διάγραμμα Ροής Δεδομένων



CIMaS Παρουσίαση Εφαρμογής



Ερωτήσεις – Απορίες



ΤΕΛΟΣ

