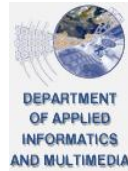




ΤΕΙ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ



ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΠΟΛΥΜΕΣΩΝ

Θέμα Εργασίας:

Μελέτη Εναλλακτικών Τρόπων
Αλληλεπίδρασης σε Φορητές Συσκευές
(PDA).

Σπουδαστής:

Στεφανακής Δημήτρης

Εισηγητής: Δρ. Ακουμιανάκης Δημοσθένης

Εισαγωγικό Σημείωμα

Η εργασία που υλοποιήθηκε είναι το αποτέλεσμα μίας επίπονης ερευνητικής δουλειάς διάρκειας 8 μηνών.

Θα ήθελα να ευχαριστήσω τον εισηγητή και επιβλέπων καθηγητή μου Κο Δημοσθένη Ακουμιανάκη που μου προσέφερε την ευκαιρία να εργαστώ σε ένα τόσο τεχνολογικά και ερευνητικά ενδιαφέρον θέμα καθώς και για την καθοδήγηση του σε όλη την διάρκεια πραγμάτωσης της εργασίας αυτής. Κατόπιν ευχαριστώ την ομάδα του ερευνητικού εργαστηρίου istLab (Βιδάκης Νικόλαος, Κότσαλη Δημήτριος, Βελλής Γεώργιος, Πλεμένος Ανάργυρος) για την χρήσιμη βοήθεια της και ιδιαίτερα τον Κο Γιάννη Μηλολιδάκη για την πολύτιμη βοήθεια του στην επίλυση τεχνικών προβλημάτων που προέκυψαν κατά την διάρκεια της εργασίας αυτής καθώς και για το ενδιαφέρον που επέδειξε.

Επίσης επιθυμώ να ευχαριστήσω τους γονείς μου για τις προσπάθειες που κατέβαλαν όλα αυτά τα χρόνια για να σπουδάσω καθώς και την στήριξη που μου παρείχαν. Τέλος θα ήθελα να ευχαριστήσω ιδιαίτερα την Χαρούλα που με στηρίζει επί 9 χρόνια και ανέχεται τις πολλές ιδιοτροπίες μου.

ΕΙΣΑΓΩΓΗ.....	2
ΚΕΦΑΛΑΙΟ 1	4
Η Γλώσσα Προγραμματισμού JAVA	4
<i>J2SE</i>	4
<i>J2EE</i>	4
<i>J2ME</i>	4
<i>Πλεονεκτήματα της Java</i>	5
Ασφαλής	5
Αξιόπιστη	5
Αντικειμενοστραφής.....	5
Δωρεάν	6
Απλή	6
Άλλα χρήσιμα χαρακτηριστικά.....	6
<i>J2ME</i>	7
<i>Στόχοι της J2ME Αρχιτεκτονικής</i>	8
<i>Επίπεδα της J2ME Αρχιτεκτονικής</i>	9
<i>Διαμορφώσεις (Configurations)</i>	9
CLDC (Connected Limited Device Configuration).....	9
CDC (Connected Device Configuration).....	10
<i>Προφίλ (profiles)</i>	11
<i>Προαιρετικά Πακέτα (Optional Packages)</i>	12
<i>Κοινότητα JCP και JSR</i>	14
<i>Εικονική Μηχανή (Virtual Machine)</i>	15
KVM.....	15
HotSpot Virtual Machine.....	15
CVM.....	15
<i>Συνολικά Ο Κόσμος Της J2ME</i>	15
<i>Σύνοψη</i>	17
ΚΕΦΑΛΑΙΟ 2	18
Λειτουργικά Συστήματα Φορητών Συσκευών	18
<i>Λειτουργικά Συστήματα Φορητών Συσκευών και J2ME</i>	19
<i>Εικονικές Μηχανές και Υλοποιήσεις Διαμορφώσεων Τρίτων Κατασκευαστών</i>	20
ΚΕΦΑΛΑΙΟ 3	22
E-kones Experience (Εργαλεία Ανάπτυξης).....	22
<i>MioP560</i>	22
Πρωτόκολλο NMEA.....	22
<i>IBM J9 MIDP 2.0 Java Emulator V2.3</i>	23

<i>MIDP 2.0 (JSR-118)</i>	24
<i>Αρχιτεκτονική</i>	24
<i>LWUIT Toolkit</i>	25
<i>Google Maps API</i>	26
<i>NetBeans IDE</i>	27
ΚΕΦΑΛΑΙΟ 4	28
Αρχιτεκτονική E-kones Experience	28
<i>Αντικείμενα της εφαρμογής</i>	30
IstlContainer	30
IstlLayout.....	30
IstlLabel	30
IstlComboBox.....	31
IstlMenuItem	33
IstlPopUp.....	34
IstlRateItem	34
<i>Η αρχιτεκτονική του αντικειμένου Calendar</i>	36
<i>Η αρχιτεκτονική του αντικειμένου istlMap</i>	38
<i>Αλληλεπίδραση με τον υποδοχέα MAP του αντικειμένου istlMap</i>	40
<i>Σημάνσεις Πάνω Στον Χάρτη</i>	40
<i>Νήματα</i>	41
<i>Αλγόριθμοι Μαθηματικών Υπολογισμών Χάρτη</i>	43
<i>Αντικείμενο Επικοινωνίας με Υπηρεσία Google Maps</i>	47
<i>Αντικείμενο GPS</i>	48
<i>Διαχείριση δεδομένων Εφαρμογής</i>	51
<i>Προσπέλαση Του Συστήματος Αρχείων της Συσκευής</i>	51
<i>Προσπέλαση Δεδομένων από την Υπηρεσία του E-kones</i>	51
<i>XML Parser</i>	60
<i>Παράδειγμα χρήσης του Parser από την εφαρμογή</i>	62
ΚΕΦΑΛΑΙΟ 5	64
Σενάριο Χρήσης της εφαρμογής E-kones Experience	64
<i>Σύνδεση με την υπηρεσία του e-kones</i>	64
<i>Επιλογή Πακέτου</i>	65
<i>Προσωπικές Πληροφορίες Χρήστη</i>	66
<i>Πληροφορίες Άλλων Χρηστών Εγγεγραμμένοι στο ίδιο Πακέτο</i>	66
<i>Χρήση Του Ημερολογίου</i>	67
<i>Υπο-Αντικείμενο Χρόνου</i>	67
<i>Υπο-Αντικείμενο Μήνα</i>	67
<i>Υπο-Αντικείμενο Ημέρες Μήνα</i>	67
<i>Χρήση Του Χάρτη</i>	73
ΚΕΦΑΛΑΙΟ 6	77

Κύκλοι Ανάπτυξης και Προβλήματα	77
<i>Σύνοψη</i>	<i>79</i>
ΠΑΡΑΡΤΗΜΑ.....	80
<i>Βοηθητική Εφαρμογή XML Parser</i>	<i>80</i>
<i>Βοηθητική Εφαρμογή GPS.....</i>	<i>81</i>
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	82
What are KVM and CVM in J2ME?	82
J2ME Core Concepts by Eric Giguère.....	82
A Brief Look at Java 2 Micro Edition by Michael Nygard.....	82
Types of configurations in J2ME By Ivan Lim	82
Articles by Eric Giguere	82
Dive into the J2ME configuration layer by Puneet Gupta	82
J2ME: Introduction, Configurations and Profiles by Urs Steiner University of Zürich	82
Think small with J2ME by Soma Ghosh	82
The K virtual machine (KVM)	82
Sun Central Webpage with links to white papers and data sheets for the Java 2 Platform, Micro Edition (J2ME) and Java Card technologies	82
Java Community Process Website	82
Introduction to the Java 2 Micro Edition (J2ME) Platform By Prentice Hall.....	82
J2ME Programming Models Technical Information Paper – v1.0 19 October 2004 by Kevin Horowitz.....	82
Overview of Java 2 Platform, Micro Edition (J2ME™) (Chapter 2).....	83
Java In Small things Manning 2002	83
J2ME: The Complete Reference McGraw-Hill/Osborne 2003.....	83
Introduction to Mobile Devices	83
Mobile operating system from Wikipedia	83
Mobile & Embedded Community on Java.net.....	83
SIRF NMEA Protocol Manual	83
SIRF StartIII GPS Chipset.....	83
NMEA Protocol.....	83
Google Static Maps API	83

ΕΙΣΑΓΩΓΗ

Τις τελευταίες δύο δεκαετίες η ανάπτυξη της ψηφιακής τεχνολογίας οδήγησε στην άνθηση των φορητών συσκευών. Κάνοντας μία ιστορική αναδρομή από την εμφάνιση του πρώτου προσωπικού υπολογιστή από την IBM (1981) μπορεί κανείς να παρατηρήσει ότι υπάρχει τάση για αύξηση της υπολογιστικής ισχύς με ταυτόχρονη μείωση του μεγέθους και του κόστους απόκτησης. Ταυτόχρονα η εξέλιξη στους υπολογιστές οδήγησε και σε μία άλλη εξέλιξη, την εξέλιξη των γλωσσών προγραμματισμού. Οι νέες γλώσσες προγραμματισμού μπορούν να εκμεταλλευτούν τις νέες δυνατότητες που προσφέρουν αυτοί οι υπολογιστές, ενώ εξελίσσεται και η ίδια η φιλοσοφία τους. Από τις δομημένες γλώσσες προγραμματισμού που προσφέρονταν για χαμηλού επιπέδου προγραμματισμό (C) έχει γίνει μετάβαση σε υψηλού επιπέδου αντικειμενοστραφείς γλώσσες προγραμματισμού (JAVA).

Ανέκαθεν για τον κλάδο της τεχνολογίας η κατηγορία των φορητών συσκευών υψηλών δυνατοτήτων αποτελούσε ιδιαίτερο στόχο. Αυτό είναι εμφανές ακόμα και σήμερα από την ραγδαία άνθηση των κινητών/φορητών συσκευών. Έτσι μπορεί κανείς να φανταστεί την εξέλιξη από τις πρώτες κινητές συσκευές που ήταν αρκετά ογκώδης σε μέγεθος με ασπρόμαυρη οθόνη, μονοφωνικό ήχο και με λογισμικό που προσέφερε τα απόλυτος απαραίτητα όπως τηλεφωνικό κατάλογο και διαχείριση μηνυμάτων, στις σύγχρονες μικρές και ελαφριές συσκευές με έγχρωμες οθόνες, ψηφιακό στερεοφωνικό ήχο, φωτογραφικές μηχανές, GPS, μεγάλη μνήμη και με λογισμικό υψηλών δυνατοτήτων. Η ίδια εξέλιξη είναι εμφανείς και στα PDA (Personal Digital Assistant) όπου όταν εμφανίστηκαν είχαν ελάχιστη μνήμη και δυνατότητες λίγο περισσότερες από ένα κομπιουτεράκι, φθάνοντας σήμερα να υπάρχουν συσκευές με έγχρωμες οθόνες, στερεοφωνικό ήχο, GPS, οθόνη αφής, μεγάλη μνήμη και με λειτουργικό σύστημα. Ακόμα και οι γνώριμοι σε όλους μας φορητοί υπολογιστές που αποτελούσαν ανέκαθεν ένα σημείο ενδιαφέροντος για τον κλάδο της πληροφορικής έχουν εξελιχθεί. Αρχικά οι φορητοί υπολογιστές ήταν ογκώδεις με μικρής χωρητικότητας σκληρό δίσκο, οθόνες χαμηλής ευκρίνειας και με ελάχιστες δυνατότητες επέκτασης ενώ πλέον οι σύγχρονοι φορητοί υπολογιστές έχουν μικρό βάρος, οθόνη υψηλής ευκρίνειας και με όλο των εξοπλισμό που έχει ένας υπολογιστής γραφείου, ενώ ήδη τα τελευταία χρόνια κάνουν την εμφάνιση τους και η mini φορητοί υπολογιστές με ακόμα μικρότερο μέγεθος. Η εξέλιξη των φορητών συσκευών είναι πλέον γεγονός.

Αυτή η εξέλιξη οδηγεί αναπόφευκτα σε νέες προγραμματιστικές ανάγκες, οι επεξεργαστές νέας γενιάς περιλαμβάνουν μεγαλύτερο εύρος εντολών, η εμφάνιση των οθονών αφής δημιουργεί νέες δυνατότητες στον τομέα της αλληλεπίδρασης, η αύξηση του χώρου αποθήκευσης και η δυνατότητα επεκτασιμότητας με κάρτες μνήμης άρει το φράγμα του αποθηκευτικού περιορισμού. Ενώ η ενσωμάτωση ηχείων ακόμα και μικροφώνου σε συνδυασμό με οθόνες υψηλής ευκρίνειας προσφέρει πολυμεσικές δυνατότητες που παλαιότερα δεν υπήρχαν. Η δυνατότητα σύνδεσης με το δίκτυο που προσφέρει η άνθηση των ψηφιακών τεχνολογιών (3G GPRS, WIFI) δημιουργεί νέες δυνατότητες στον τομέα της επικοινωνίας, αυτά και άλλα πολλά

οδηγούν στο άνοιγμα των καθιερωμένων γλωσσών προγραμματισμού από τον χώρο των υπολογιστών γραφείου στον χώρο των φορητών συσκευών. Πλέον ο προγραμματισμός φορητών συσκευών αποτελεί μία πραγματικότητα, γλώσσες προγραμματισμού όπως η JAVA, C++, C# προσφέρουν δυνατότητες στο κάθε προγραμματιστή να δημιουργήσει την δική του εφαρμογή και να προσωποποιήσει την συσκευή του με βάση τις ανάγκες του.

Στην εργασία αυτή θα εξετάσουμε φορητές συσκευές μικρού μεγέθους (όχι φορητούς υπολογιστές) και θα γίνει μία προσπάθεια μελέτης για τους σύγχρονους τρόπους αλληλεπίδρασης σε φορητή συσκευή. Για τον σκοπό αυτό επιλέχθηκε η γλώσσα προγραμματισμού Java και υλοποιήθηκε πρόγραμμα (E-kones client) σε φορητή συσκευή τύπου PDA. Η εργασία αυτή θα επικεντρωθεί στο κομμάτι ανάπτυξης λογισμικού σε φορητή συσκευή με την χρήση της γλώσσας προγραμματισμού Java MicroEdition.

ΚΕΦΑΛΑΙΟ 1

Η Γλώσσα Προγραμματισμού JAVA

Η γλώσσα προγραμματισμού JAVA έχει εξελιχθεί όπως όλα τα περιβάλλοντα ανάπτυξης και γλώσσες προγραμματισμού. Πολλά νέα χαρακτηριστικά και δυνατότητες έχουν προστεθεί στην γλώσσα προγραμματισμού JAVA από την αρχική της έκδοση. Βελτιώσεις υπάρχουν επίσης και στον τομέα της απόδοσης, ευελιξίας και αξιοπιστίας. Έτσι πλέον έχει φτάσει αισίως στην 2^η έκδοση. Η εταιρία Sun Microsystems δημιουργός και διαχειριστής της JAVA έχει κατηγοριοποιήσει την JAVA σε 3 εκδόσεις.



Standard Edition (J2SE)
Enterprise Edition (J2EE)
Micro Edition (J2ME)

Η κάθε έκδοση διευθετεί τις ανάγκες της JAVA για συγκεκριμένες κατηγορίες εφαρμογών. Η J2ME είναι η τρίτη και τελευταία από τις τρεις εκδόσεις της 2^{ης} έκδοσης της JAVA.

J2SE

Η 2^η έκδοση της JAVA (J2SE) είναι η έκδοση της JAVA για τα βασικά περιβάλλοντα. Ενσωματώνει τις βασικές κλάσεις και διασυνδέσεις που επιτρέπουν την ανάπτυξη και εκτέλεση βασικών εφαρμογών όπως διακοσμητή-πελάτη, εφαρμογών γραφείου καθώς και εφαρμογές που εκτελούνται σε περιβάλλοντα περιηγητή διαδικτύου.

J2EE

Η J2EE είναι ένα σύνολο από διάφορα APIs και τεχνολογίες όχι-JAVA. Η γενική της χρήση είναι για την δημιουργία και πιθανή διανομή εφαρμογών διαφορετικών τεχνολογιών.

J2ME

Η Sun εισήγαγε, την Micro Edition (J2ME) τον Ιούνιο του 1999. Η J2ME σχεδιάστηκε για να διευθετήσει τις ανάγκες της JAVA στην κοινότητα των φορητών και φορετών συσκευών. Αρχικά η J2ME κτίστηκε για συσκευές περιορισμένης ισχύος, με σύνδεση διαδικτύου και δυνατότητες γραφικών διεπαφών. Σήμερα πλέον η J2ME έχει επεκταθεί έτσι ώστε να καλύψει μία μεγαλύτερη συλλογή από φορητές συσκευές υψηλών δυνατοτήτων μέχρι απλές συσκευές ειδοποίησης (βομβητές).

Πλεονεκτήματα της Java

Η JAVA έχει πολλά να προσφέρει. Πέρα από την μη εξάρτηση της με συγκεκριμένη πλατφόρμα, έχει μία σειρά από χαρακτηριστικά που την κάνουν ελκυστική γλώσσα προγραμματισμού. Αυτά τα χαρακτηριστικά κάνουν την JAVA πιθανό υποψήφιο για εφαρμογές λογισμικού ανεξάρτητα από το μέγεθος και το σχήμα της πλατφόρμας ανάπτυξης.

Ασφαλής

Η ασφάλεια αποτελούσε ιδιαίτερο σημείο ενδιαφέροντος για τους σχεδιαστές και προγραμματιστές από την αρχή. Ακόμα και κατά την διάρκεια ανάπτυξης της γλώσσας οι οικιακές ηλεκτρικές συσκευές αποτελούσαν στόχο. Οι κατασκευαστές γνώριζαν ότι αυτές οι συσκευές θα είναι δικτυωμένες και γι' αυτό θα είναι ευάλωτες σε επιθέσεις από άλλες συσκευές ή δίκτυα. Καθώς η JAVA προοριζόταν να γίνει η γλώσσα προγραμματισμού για το διαδίκτυο, και επειδή η ασφάλεια είχε ενσωματωθεί από την αρχή, προσφέρεται για να διευθετήσει θέματα ασφαλείας σχετικά με το διαδίκτυο.

Αξιόπιστη

Η αξιοπιστία της JAVA έγκειται στο γεγονός ότι δεν χρειάζεται να διαχειριστεί μερικούς από τους χαμηλού επιπέδου πόρους όπως η μνήμη και οι δείκτες. Οι προγραμματιστές μπορούν να διαχειριστούν πληροφορίες όπως αντικείμενο ή συλλογή αντικειμένων, με τρόπο πιο εύχρηστο. Οι χαμηλού επιπέδου πόροι όπως οι δείκτες έχουν την τάση να ξεχνιούνται σε προγράμματα μεγάλης κλίμακας με αποτέλεσμα να δημιουργούνται διαρροές και προβλήματα προσπέλασης μνήμης. Η JAVA τοποθετεί το φορτίο της διαχείρισης αναφοράς μνήμης σε ένα σύστημα συλλογής σκουπιδιών όπου αυτές οι αναφορές μνήμης διατηρούνται με μεγαλύτερη μεθοδικότητα από ότι σε άλλες γλώσσες προγραμματισμού όπως η C++ όπου το βάρος της διαχείρισης πέφτει στον προγραμματιστή.

Αντικειμενοστραφής

Η JAVA θεωρείται ως γνήσια αντικειμενοστραφής γλώσσα προγραμματισμού. Οι λόγοι που καθιστούν μια αντικειμενοστραφής γλώσσα προγραμματισμού τόσο σημαντική είναι καταρχήν ότι είναι πιο εύκολο στο κτίσιμο και συντήρηση εφαρμογών με την χρήση αντικειμενοστραφής γλώσσας προγραμματισμού. Αυτό γιατί οι αντικειμενοστραφείς προγραμματιστικές δομές ενεργά ενθαρρύνουν τους προγραμματιστές να οργανώνουν τις εφαρμογές τους σε μικρά κατανοητά διαχειρίσιμα τμήματα. Μεγάλες περίπλοκες εφαρμογές μπορούν να κτιστούν με συνδυασμό μικρότερων τμημάτων. Αυτά τα τμήματα δηλαδή τα αντικείμενα περιέχουν συμπεριφορά αλλά και δεδομένα. Έχοντας τα δεδομένα και την συμπεριφορά σε ένα αντικείμενο υπάρχει σιγουριά ότι μόνο το τμήμα κώδικα υπεύθυνο για τα δεδομένα μπορεί να τα αλλάξει. Έν τέλει τα αντικείμενα ως αναπαράσταση πραγμάτων από τον πραγματικό κόσμο παρέχουν την δυνατότητα σε περισσότερους ανθρώπους να πάρουν μέρος και να κατανοήσουν την εσωτερική λειτουργία της εφαρμογής.

Δωρεάν

Τα βασικά περιβάλλοντα ανάπτυξης και εκτέλεσης παρέχονται δωρεάν. Η SUN έχει στον έλεγχο της την JAVA. Δωρεάν παρέχονται οι βασικές υλοποιήσεις αναφοράς, εργαλεία ανάπτυξης, κλάσεις και περιβάλλον εκτέλεσης.

Απλή

Η JAVA είναι απλή, γι' αυτούς που έχουν ήδη μία εμπειρία συγκριτικά με τις άλλες γλώσσες προγραμματισμού. Η σύνταξη της JAVA είναι απλή και κατανοητή. Η σύνταξη της μοιάζει με αυτή της C++ και περιλαμβάνει πολλά από τα πλεονεκτήματα που εντοπίζονται σε άλλες γλώσσες προγραμματισμού.

Άλλα χρήσιμα χαρακτηριστικά

Υπάρχουν πολλά χαρακτηριστικά που κάνουν την JAVA ιδανική γλώσσα προγραμματισμού, όπως Πολύ-νημάτωση (Multi-threading), εξαιρέσεις/Διαχείριση Σφαλμάτων, Δυναμικό δέσιμο, και απόδοση.

J2ME

Η SUN παρουσίασε τον Ιούνιο του 1999 στο συνέδριο JavaOne την J2ME μαζί με την πρώτη Εικονική Μηχανή (Virtual Machine) ονόματι KVM που θα την εκτελούσε. Η J2ME απευθύνεται σε κατασκευαστές έξυπνων ασύρματων και μικρών υπολογιστικών συσκευών με ανάγκες επικοινωνίας σε διαφορετικές πλατφόρμες. Υπάρχουν μεγάλες απαιτήσεις από τους καταναλωτές κινητών συσκευών και μικρών υπολογιστικών συσκευών στον τομέα της. Απαιτήσεις για γρήγορο χρόνο απόκρισης, συμβατότητα με εταιρικές υπηρεσίες και εφαρμογές με πλήρη χαρακτηριστικά σε μία μικρή συσκευή καθώς και η παραδοχή ότι το ίδιο λογισμικό και δυνατότητες που εκτελούνται σε ένα υπολογιστή γραφείου και φορητό υπολογιστή να είναι διαθέσιμες σε κινητά καθώς και σε PDA. Για να υλοποιηθούν αυτές οι απαιτήσεις οι προγραμματιστές έπρεπε να ξανασκεφτούν τον τρόπο με τον οποίο κατασκεύαζαν υπολογιστικά συστήματα. Οι προγραμματιστές έπρεπε να τιθασεύσουν την ισχύ των ήδη υπάρχων προγραμμάτων και να μεταφέρουν αυτή την ισχύ σε μικρές κινητές και ασύρματες υπολογιστικές συσκευές. Η J2ME επιτρέπει αυτή την μεταφορά με μικρές μετατροπές, δεδομένου ότι οι εφαρμογές είναι σχεδιαστικά κλιμακωτές έτσι ώστε η εφαρμογή να μπορεί να προσαρμοστεί στους διαθέσιμους πόρους των μικρών υπολογιστικών συσκευών.

Ένα χαρακτηριστικό πλεονέκτημα που προσφέρει η χρήση της J2ME είναι ότι είναι συμβατή με όλες τις συσκευές που μπορούν να εκτελέσουν JAVA. Συσκευή που τρέχει JAVA είναι ένας υπολογιστής που εκτελεί μία Java Virtual Machine. Η Ericson, Sony, Motorola, Nextel, Nokia, Panasonic και RIM είναι μερικοί από τους κατασκευαστές που έχουν συσκευές οι οποίες μπορούν να εκτελέσουν JAVA. Επιπρόσθετα η J2ME διατηρεί τα ισχυρά χαρακτηριστικά ασφαλείας που εντοπίζονται στην γλώσσα προγραμματισμού JAVA.

Στόχοι της J2ME Αρχιτεκτονικής

Οι στόχοι της J2ME αρχιτεκτονικής διαφέρουν κατά πολύ από αυτούς της J2SE και J2EE. Παρακάτω ακολουθούν τα κυριότερα χαρακτηριστικά της αρχιτεκτονικής.

Παροχή υποστήριξης για μία πληθώρα συσκευών με διαφορετικές δυνατότητες. Αυτές οι συσκευές πολλές φορές έχουν διαφοροποιήσεις όπως στον τομέα της διεπαφής χρήστη, αποθήκευσης δεδομένων, συνδεσιμότητας δικτύου και δικτυακού εύρους ζώνης, διαχείρισης μνήμης, κατανάλωσης ενέργειας και απαιτήσεις εγκατάστασης.

Παροχή μίας αρχιτεκτονικής που μπορεί να βελτιστοποιηθεί για μικρούς αποθηκευτικούς χώρους και μικρότερο αποτύπωμα μνήμης.

Εστίαση σε συσκευές με υψηλές δυνατότητες προσωποποίησης.

Παροχή δικτυακής συνδεσιμότητας μέσα από ένα πεδίο δικτυακών δυνατοτήτων και υπηρεσιών. Η δικτυακή συνδεσιμότητα είναι καίριας σημασίας για συσκευές στον χώρο της J2ME και οι δυνατότητες τους ποικίλουν από χαμηλού εύρους, ασύρματες και εσωτερικές συνδέσεις σε συνδέσεις υψηλής ταχύτητας και υψηλού εύρους ζώνης.

Παροχή βέλτιστων μέσων για μετακίνηση εφαρμογών και δεδομένων μέσω σύνδεσης δικτύου. Πολλές φορές το δίκτυο είναι προτιμότερη μέθοδος για μεταφορά J2ME εφαρμογών σε συσκευές. Οι εφαρμογές πρέπει να έχουν την δυνατότητα να εγκαθίστανται σε συσκευές ή να φορτώνονται απευθείας στην μνήμη και να αποβάλλονται μετά την εκτέλεση.

Μεγιστοποίηση των δυνατοτήτων της γλώσσας προγραμματισμού Java μεταξύ διαφορετικών συσκευών, κάνοντας χρήση των μοναδικών δυνατοτήτων και περιορισμών της κάθε συσκευής.

Μεγιστοποίηση της ευελιξίας και παροχή τρόπων για υποστήριξη της όλο ένα αυξανόμενης αλλαγής στην αγορά καθώς και υιοθέτηση το ήδη και μελλοντικών εφαρμογών.

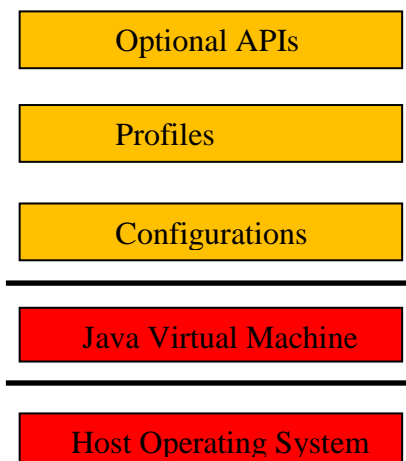
Παροχή μέσων προς τρίτους κατασκευαστές για την δημιουργία εφαρμογών στην J2ME.

Παροχή μέσων για κλιμάκωση των εφαρμογών σε συσκευές με διαφορετικές δυνατότητες, χαρακτηριστικά και τύπους επεξεργαστών.

Επίπεδα της J2ME Αρχιτεκτονικής

Η Αρχιτεκτονική της J2ME από την αρχή έπρεπε να διευθετήσει το μεγάλο θέμα της ποικιλίας των συσκευών καθώς και των δυνατοτήτων τους. Ο σκοπός ήταν να δημιουργηθεί μία αρχιτεκτονική που να καλύψει από απλές φορητές συσκευές με περιορισμένους πόρους συστήματος και συνδεσιμότητας μέχρι συσκευές με πολλούς πόρους, υψηλές δυνατοτήτας συνδεσιμότητας καθώς και ιδιαίτερους τύπος αλληλεπίδρασης (οθόνες αφής), ενώ ταυτόχρονα έπρεπε να διατηρηθεί η αρχή του “Γράφεται μία Φορά, Τρέχει Παντού” (Write Once Run Everywhere). Για να επιτευχθεί αυτό η sun δημιούργησε ένα μοντέλο αρχιτεκτονικής τριών επιπέδων.

Στο χαμηλότερο επίπεδο βρίσκονται οι διαμορφώσεις (Configurations) με βασική λειτουργία τον προσδιορισμό των απαραίτητων χαρακτηριστικών που πρέπει να παρέχει μία εικονική μηχανή (virtual machine) για να εκτελέσει την συγκεκριμένη διαμόρφωση καθώς και τις βασικές κλάσεις. Στο αμέσως επόμενο επίπεδο βρίσκονται τα προφίλ (profiles) όπου επεκτείνουν και εμπλουτίζουν τις βασικές κλάσεις ενώ στο τελευταίο επίπεδο βρίσκονται τα προαιρετικά APIs (JSRs) τα οποία προσφέρουν πακέτα κλάσεων με συγκεκριμένες λειτουργίες. Κάτω από τα παραπάνω επίπεδα βρίσκεται η εικονική μηχανή (Virtual Machine) η οποία είναι υπεύθυνη για την μετάφραση του Java byte code σε γλώσσα μηχανής του αντίστοιχου συστήματος.



Εικόνα 1.1 Τα επίπεδα αρχιτεκτονικής της J2ME

Διαμορφώσεις (Configurations)

Για να μπορέσει η Sun να υποστηρίξει την μεγάλη πληθώρα συσκευών διαφορετικών δυνατοτήτων, δημιούργησε δύο βασικές διαμορφώσεις. Η πρώτη διαμόρφωση στοχεύει σε φορητές συσκευές χαμηλών δυνατοτήτων (CLDC - Connected Limited Device Configuration) ενώ η δεύτερη σε φορητές συσκευές υψηλών δυνατοτήτων (CDC - Connected Device Configuration).

CLDC (Connected Limited Device Configuration)

Η CLDC διαμόρφωση απευθύνεται σε συσκευές που πληρούν τα παρακάτω κριτήρια:

- Το λιγότερο 192KB συνολικής μνήμης.

- Επεξεργαστής 16-bit ή 32-bit.
- Χαμηλή κατανάλωση ισχύος, συχνά με την χρήση μπαταρίας.
- Υποτυπώδης συνδεσιμότητα με δίκτυα, όπως ασύρματα και με περιορισμένο εύρος ζώνης.

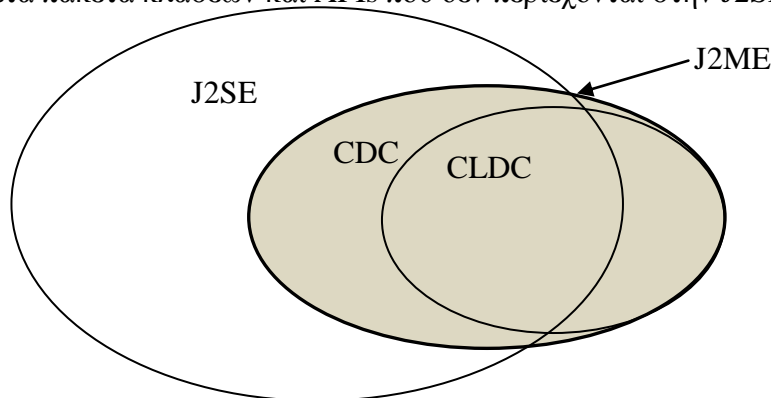
Πιο συγκεκριμένα η CLDC διαμόρφωση προσδιορίζει μία πλατφόρμα ανάπτυξης JAVA εφαρμογών με τα παρακάτω χαρακτηριστικά και στόχους.

- **Μικρό αποτύπωμα μνήμης.** Η CLDC διαμόρφωση προσδιορίζει το ελάχιστο σύνολο από τα χαρακτηριστικά την γλώσσας προγραμματισμού JAVA (κλάσεις, APIs).
- **Επικέντρωση στον προγραμματισμό εφαρμογών, αντί προγραμματισμό συστήματος.** Η CLDC διαμόρφωση περιλαμβάνει μόνο βιβλιοθήκες υψηλού επιπέδου που παρέχουν την απαραίτητη προγραμματιστική δυνατότητα στον προγραμματιστή εφαρμογών. Ενώ δίνεται μεγάλη έμφαση στην φορητότητα και γενικότητα. Η διαμόρφωση CLDC δεν παρέχει κανένα API που είναι δεμένο με συγκεκριμένη κατηγορία συσκευών, κάθετη αγορά ή λειτουργία συστήματος.
- **Δυνατότητα δυναμικού κατεβάσματος εφαρμογών και ενθάρρυνση τρίτων προγραμματιστών εφαρμογών.** Ένας από τους κύριους στόχους της διαμόρφωσης CLDC είναι να προσδιορίσει ένα περιβάλλον όπου θα μπορεί να γίνει ασφαλές κατέβασμα εφαρμογών JAVA μέσα από διαφόρους τύπους δικτύου στην συσκευή του χρήστη.

CDC (Connected Device Configuration)

Η διαμόρφωση CDC έχει σαν σκοπό να καλύψει το κενό μεταξύ της διαμόρφωσης CLDC και της J2SE που απευθύνεται σε υπολογιστές γραφείου και φορητούς υπολογιστές. Η διαμόρφωση CDC έχει σαν στόχο φορητές συσκευές που διαθέτουν περισσότερες υπολογιστικές δυνατότητες από αυτές που προσδιορίζει η διαμόρφωση CLDC.

Όπως είναι φανερό οι δύο διαμορφώσεις έχουν ως στόχο να καλύψουν διαφορετικές ανάγκες. Οι δύο διαμορφώσεις αποτελούν την J2ME η οποία με την σειρά της είναι ένα υποσύνολο της J2SE, για την ακρίβεια μάλιστα η J2ME περιλαμβάνει επιπρόσθετα πακέτα κλάσεων και APIs που δεν περιέχονται στην J2SE (εικόνα 1.2).



Εικόνα 1.2 Ο Κόσμος της J2ME

Προφίλ (profiles)

Οι διαμορφώσεις παρέχουν βασικές υπηρεσίες εφαρμογών, αλλά όχι κλάσεις για την διαχείριση του χρόνου ζωής της εφαρμογής, οδήγησης της διεπαφής χρήστη, διατήρησης και ενημέρωσης δεδομένων στην συσκευή, ή για ασφαλή πρόσβαση σε πληροφορίες αποθηκευμένες σε διακοσμητή διαδικτύου. Αυτή την λειτουργικότητα την παρέχουν τα προφίλ και τα προαιρετικά πακέτα. Τα προφίλ είναι κτισμένα πάνω σε μία διαμόρφωση και επεκτείνουν αυτήν παρέχοντας κάθετη λειτουργικότητα που σχετίζεται με χαρακτηριστικά από μία οικογένεια συσκευών η οποία λείπει από την διαμόρφωση. Με βάση τις δύο διαμορφώσεις υπάρχουν δύο οικογένειες προφίλ.

- **CLDC**
 - **MIDP (Mobile Information Profile)** είναι το βασικότερο προφίλ στοχεύοντας κινητές κυρίως συσκευές. Αυτή την στιγμή βρίσκεται στην έκδοση 2.1, με την έκδοση 3.0 να είναι υπό ανάπτυξη.
 - **IMP (Information Module Profile)** έχει σαν στόχο ακόμα μικρότερες συσκευές από ότι το MIDP όπως βιομηχανικές συσκευές, συστήματα ασφαλείας κλπ. Η τελευταία έκδοση είναι η IMP-NG (Next Generation).
 - **Digital Set-Top Box Profile**
- **CDC**
 - **Foundation Profile** version 1.1.2 (JSR 219)
 - **Personal Basis Profile** version 1.1.2 (JSR 217) παρέχει ένα περιβάλλον εφαρμογών για εφαρμογές βασισμένες σε AWT οι οποίες βασίζονται σε ελαφρού τύπου (lightweight) αντικείμενα. Παρέχει βασική υποστήριξη του AWT αλλά όχι υποστήριξη για τα βαρύ τύπου αντικείμενα. Αυτό το προφίλ έχει το ρόλο της βάσης πάνω στην οποία μπορούν να κτιστούν βιβλιοθήκες σαν το Swing καθώς και άλλες ελαφρού τύπου και ενισχύει την βασική λειτουργικότητα του AWT που παρείχε το Personal Profile.
 - **Personal Profile** version 1.1.2 (JSR 216) παρέχει ένα περιβάλλον ανάπτυξης εφαρμογών που βασίζεται στα βαρύ τύπου αντικείμενα του AWT. Παρέχει υποστήριξη AWT συμβατή με την έκδοση 1.1 του JDK και περιέχει μία πληθώρα από τις τελευταίες βελτιώσεις στα χαρακτηριστικά αυτά. Το συγκεκριμένο προφίλ εξυπηρετεί ως μία κατάλληλη πλατφόρμα για web applets και σαν ένα μεταβατικό μονοπάτι προς τις προσωπικές εφαρμογές Java.

Στην πορεία ανάπτυξης και βελτίωσης των προφίλ προτάθηκαν και άλλα εκτός από αυτά που παρουσιάστηκαν προηγούμενος, πολλά από αυτά μάλιστα στόχευαν στο να υποστηρίξουν και τις δύο διαμορφώσεις. Ένα από αυτά ήταν το Java Gaming Profile (JSR 134) το οποίο όμως το 2003 απορρίφθηκε.

Προαιρετικά Πακέτα (Optional Packages)

Τα προαιρετικά πακέτα αποτελούν επιπρόσθετα APIs που έχουν σαν στόχο μια συγκεκριμένη λειτουργία η οποία είναι συνδεδεμένη με την συσκευή. Τέτοια λειτουργία θα μπορούσε να θεωρηθεί η λειτουργία ενός GPS ή File I/O. Δεν προορίζονται αυστηρώς προς κάποια συγκεκριμένη διαμόρφωση αφού η γενική αρχιτεκτονική της J2ME επιτρέπει ευελιξία, παρόλα αυτά όμως θέτουν κάποιες συγκεκριμένες απαιτήσεις που πιθανότατα να μην καλύπτει κάποια διαμόρφωση. Παρακάτω θα γίνει μία αναφορά στα περισσότερα από αυτά.

- **Optional Packages for the J2ME(TM) Platform - FileConnection Optional Package (JSR-75)** Ο βασικός στόχος αυτού του πακέτου είναι παρέχει πρόσβαση στο σύστημα διαχείρισης αρχείων (File System) συσκευών ή καρτών μνήμης. Η συνδεσιμότητα με το σύστημα διαχείρισης αρχείων μπορεί να επιτευχθεί εφόσον η συσκευή από πλευράς υλικού υποστηρίζει κάτι τέτοιο. Απαιτεί την ύπαρξη πλατφόρμας CLDC 1.0 ή καλύτερης (CDC).
- **Wireless Messaging API (JSR-120)** βασίζεται στο γενικό πλαίσιο συνδεσιμότητας (Generic Connection Framework – GCF), το οποίο προσδιορίζεται στα χαρακτηριστικά του CLDC 1.0. Το πακέτο *javax.microedition.io* προσδιορίζει τις προδιαγραφές και υποστηρίζει είσοδο/έξοδο και τις λειτουργίες δικτύου στα προφίλ της J2ME. Παρέχει ένα συνεκτικό τρόπο πρόσβασης και οργάνωσης δεδομένων σε περιβάλλοντα περιορισμένων πόρων. Ήδη υπάρχει και η δεύτερη έκδοση (JSR-205)
- **Mobile Media API (JSR-135)** αποτελεί ένα API με υψηλό βαθμό αφαιρετικότητας έτσι ώστε να υποστηρίζει την πληθώρα συσκευών με διαφορετικές πολυμεσικές δυνατότητες.
- **JDBC connectivity (JSR-169)** αποτελεί υποσύνολο του JDBC 3.0 API και προσφέρει συνδεσιμότητα με βάσεις δεδομένων. Προορίζεται για συσκευές που υποστηρίζουν την διαμόρφωση CDC και έχουν εγκατεστημένο το Foundation Profile.
- **Web Services (JSR-172)** παρέχει δυνατότητες πρόσβασης σε απομακρυσμένες υπηρεσίες βασισμένες σε SOAP και XML και προσφέρει τρόπους προσπέλασης XML δεδομένων.
- **Security and Send Services API (JSR-177)** προσδιορίζει προαιρετικά πακέτα με σκοπό την παροχή ασφαλών και έμπιστων υπηρεσιών με την χρήση κρυπτογραφικών μεθόδων, μεθόδων ασφαλείας κλπ.
- **Location API (JSR-179)** επιτρέπει λειτουργίες εντοπισμού θέσης από την συσκευή με την χρήση GPS. Πλέον βρίσκεται στην έκδοση 2.0 (JSR-293).
- **Mobile 3D Graphics API (JSR-184)**, ήδη στην έκδοση 2.0 (JSR-297)
- **Event Tracking API (JSR-190)** έχει σαν στόχο να προσδιορίζει τον τρόπο με τον οποίο παράγονται γεγονότα από την συσκευή (exporter) σε

ένα σύστημα συλλογής (Java API), ο τρόπος με τον οποίο θα προσδιορίζονται οι διαμορφώσεις, το πρωτόκολλο που θα είναι υπεύθυνο μεταξύ της συσκευής και του συστήματος συλλογής γεγονότων.

- **Content Handler API (JSR-211)** περιγράφει τον χειρισμό περιεχομένου καθώς και το μοντέλο εκτέλεσης. Το API και το μοντέλο επιτρέπουν σε μία εφαρμογή σε J2ME να εμπλέξει μία άλλη μη κτισμένη σε Java για περιεχόμενο με χρήση URL, ή τύπο περιεχομένου ή ID χειριστή περιεχομένου. Απαιτεί MIDP ή Personal Basis Profile.
- **Scalable 2D Vector Graphics API (JSR-226)** έχει σαν στόχο φορητές συσκευές χαμηλών δυνατοτήτων με περιορισμένη μνήμη, μέγεθος οθόνης, και υπολογιστική ισχύ. Προσδιορίζει τον τρόπο δημιουργίας κλιμακωτών γραφικών δύο διαστάσεων, συμπεριλαμβανομένου εξωτερικές εικόνες τύπου SVG, με σκοπό την χρήση τους για γραφικές απεικονίσεις (visualizations), κλιμακωτά εικονίδια, και άλλες εφαρμογές που απαιτούν κλιμακωτά γραφικά. Στην παρούσα φάση βρίσκεται στην έκδοση 2.0 (JSR-287)
- **Payment API (JSR-229)** επιτρέπει σύστημα πληρωμών για την J2ME
- **Data-Sync API (JSR-230)** επιτρέπει σε εφαρμογές τον συγχρονισμό δεδομένων που βρίσκονται αποθηκευμένα στο τερματικό με τα αντίστοιχα δεδομένα που βρίσκονται αποθηκευμένα σε διακοσμητική έτσι ώστε να παρουσιάζονται οποιεσδήποτε αλλαγές σε οποιοδήποτε στιγμιότυπο των δεδομένων. Προϋποθέτει διαμόρφωση CLDC ή CDC.
- **Mobile Internationalization API (JSR-238)** επιτρέπει σε εφαρμογές γραμμένες σε MIDP να διεθνοποιούνται, το οποίο είναι πολύ σημαντικό για την υποστήριξη ανάπτυξης πολυγλωσσικών εφαρμογών
- **Device Management API (JSR-246)**
- **Mobile Telephony API (JSR-253)** επιτρέπει τον χειρισμό φωνητικών κλήσεων και τη χρήση δικτύου για εφαρμογές γραμμένες σε J2ME. Ήδη βρίσκεται στην έκδοση 2.0 (JSR-304)
- **Mobile Sensor API (JSR-256)** επιτρέπει την προσπέλαση δεδομένων με εύκολο ομοιόμορφο τρόπο από αισθητήρες (sensors).
- **Contactless Communication API (JSR-257)** επιτρέπει την πρόσβαση σε πληροφορίες που βρίσκονται σε συσκευές που λειτουργούν με χρήση ραδιοκυμάτων όπως bar-codes, RFID ταυτότητες.
- **Mobile Customization User Interface API (JSR-258)** επιτρέπει στους προγραμματιστές να μεταχειρίζονται την εμφάνιση της διεπαφής σε μία κινητή συσκευή.
- **Mobile Broadcast Service API for handheld Terminals (JSR-272)** επιτρέπει διαδραστικές εκπομπές από υπηρεσίες για κινητά τερματικά.
- **Service Connection API (JSR-279)** παρέχει ομοιόμορφη πρόσβαση σε διαφορετικές υπηρεσίες οι οποίες είναι κτισμένες σε τεχνολογίες όπως AtomPub, REST καθώς και σε απλούστερες όπως SOAP αλλά και εσωτερικές υπηρεσίες.

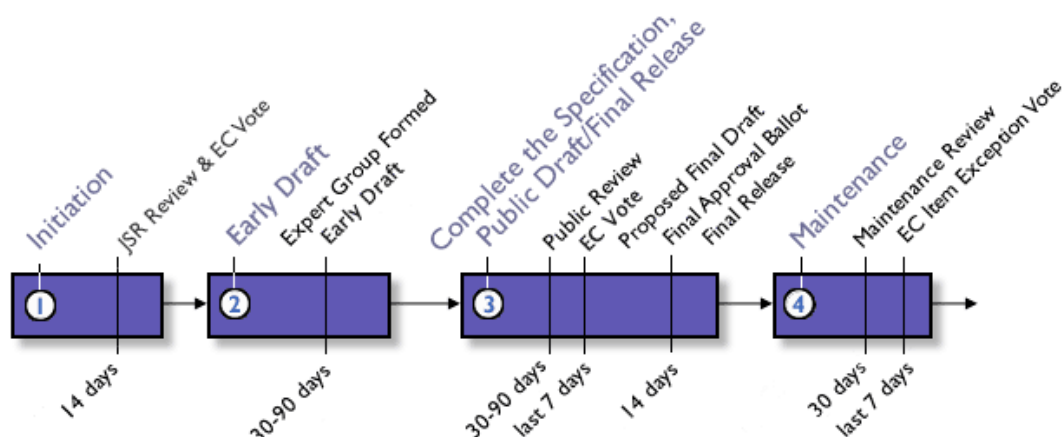
- **XML API (JSR-280)**
- **IMS Services API (JSR-281)**
- **Telematics API (JSR-298)** παρέχει τηλεματικές υπηρεσίες από συσκευές που υποστηρίζουν την J2ME
- **DRM API (JSR-300)**

Η παραπάνω λίστα από προαιρετικά πακέτα καθιστά σαφές το ενδιαφέρον που υπάρχει από την βιομηχανική κοινότητα αλλά και την SUN για την J2ME. Μια παρατήρηση είναι αρκετή για να δημιουργήσει την εντύπωση ότι εφόσον υπάρχουν προδιαγραφές σχεδόν για όλες τις προγραμματιστικές ανάγκες, η J2ME αποτελεί μία πολύ ώριμη τεχνολογία που έχει προβλέψει και λύσει όλα της τα προβλήματα. Στην πραγματικότητα κάτι τέτοιο δεν συμβαίνει, πριν γίνει όμως αναφορά στα προβλήματα και τις ελλείψεις που υπάρχουν είναι αναγκαίο να γίνει αναφορά σε δύο όρους.

Κοινότητα JCP και JSR

Η JCP (Java Community Process) είναι μία παγκόσμια κοινότητα της Java από το 1998 που έχει σαν σκοπό να αναπτύσσει και να ελέγχει προδιαγραφές, αναφορές για υλοποιήσεις και για λογισμικό ελέγχου που σχετίζονται με την τεχνολογία της Java. Στην JCP κοινότητα εμπλέκονται μεγάλες εταιρίες από τον χώρο των κατασκευαστών (Nokia, Sony, Ericson, Motorola, SAGEM Communications) αλλά και από τον χώρο του λογισμικού (IBM).

Το JSR είναι μία αίτηση για δημιουργία προδιαγραφής. Το έγγραφο υποβάλλεται στο Γραφείο Διαχείρισης Προγραμμάτων (Program Management Office –PMO) από ένα ή περισσότερα μέλη για να προταθεί η ανάπτυξη μίας νέας προδιαγραφής ή η σημαντική επανεξέταση μίας ήδη υπάρχουσας προδιαγραφής.



Εικόνα 1.3 JCP Timeline

Από τα παραπάνω φαίνεται ότι η JCP κοινότητα προτείνει, σχεδιάζει, επανεξετάζει προδιαγραφές αλλά δεν τις υλοποιεί. Αυτό έχει αντίκτυπο και στην J2ME τεχνολογία αφού όπως θα δειχθεί παρακάτω οι διαμορφώσεις, τα προφίλ αλλά και τα προαιρετικά πακέτα είναι σε στάδιο προδιαγραφών, όπου υπάρχουν οδηγίες για την υλοποίηση αλλά όχι η ίδια η υλοποίηση.

Εικονική Μηχανή (Virtual Machine)

Στον κόσμο της γλώσσας προγραμματισμού Java μία εφαρμογή για να μπορεί να εκτελεστεί και άρα να έχει νόημα χρειάζεται μία εικονική μηχανή η οποία θα μεταφράζει τον Java byte code σε γλωσσά που θα καταλαβαίνει η συγκεκριμένη συσκευή. Στην ουσία η εικονική μηχανή είναι υπεύθυνη για την διατήρηση του «Γράφετε μία φορά τρέχει παντού».

Με βάση τις δύο προδιαγραφές αναπτύχθηκαν και δύο αντίστοιχες εικονικές μηχανές.

KVM

Η KVM (kilo Virtual Machine) αποτελεί εξέλιξη και τμήμα της Java εικονικής μηχανής και έχει γραφτεί εξολοκλήρου σε C, με ιδιαίτερη έμφαση σε βέλτιστη απόδοση για μικρό μέγεθος, ευκολία στην μεταφορά από πλατφόρμα σε πλατφόρμα, τμηματική και επεκτάσιμη. Χρησιμοποιείται από την διαμόρφωση CLDC και στοχεύει κυρίως σε συσκευές με επεξεργαστή 16bit/32bit και συνολική μνήμη περίπου στα 256KB.

HotSpot Virtual Machine

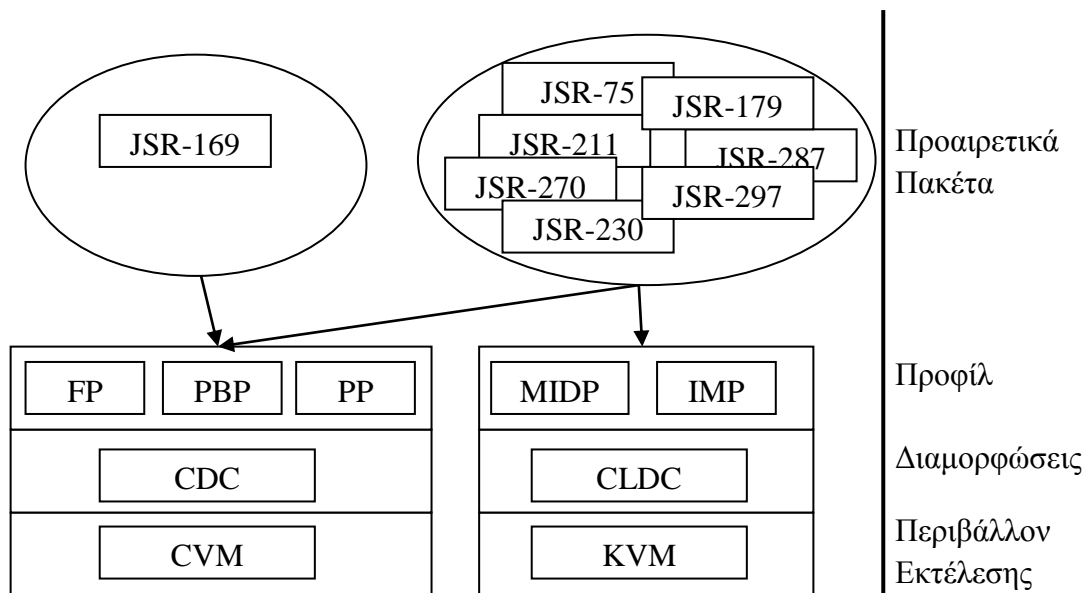
Η HotSpot Virtual Machine αναπτύχθηκε κυρίως για τις J2SE και J2EE εκδόσεις με σκοπό να προσφέρει καλύτερη απόδοση σε σχέση με την προγενέστερη εικονική μηχανή. Στην περίπτωση της J2ME αποτελεί μία εξέλιξη της KVM με καλύτερη απόδοση και λιγότερες απαιτήσεις.

CVM

Είναι η αντίστοιχη εικονική μηχανή που χρησιμοποιείται από την διαμόρφωση CDC. Περιλαμβάνει πλήρης εικονική μηχανή Java και μεγαλύτερο σύνολο από πηγαίες κλάσεις, με αποτέλεσμα να απαιτεί περισσότερη μνήμη και ταχύτερο επεξεργαστή από την KVM. Η CVM αποτελεί πλήρης JVM σχεδιασμένη για συσκευές υψηλών επιδόσεων νέας γενιάς, με επεξεργαστή 32bit και συνολική μνήμη μεγαλύτερη των 2MB.

Συνολικά Ο Κόσμος Της J2ME

Με βάση τα προηγούμενα είναι φανερό ότι η J2ME είναι σε διαρκή ανάπτυξη μέχρι και σήμερα, ενώ αποτελεί ένα ιδιαίτερα ενδιαφέρον θέμα για τις εταιρίες αλλά και για τους τελικούς χρήστες. Σε αυτό το σημείο θα γίνει μία προσπάθεια αναπαράστασης του κόσμου της J2ME με σχεδιάγραμμα (εικόνα 1.4) καθώς και σημείωση των κυριότερων διαφορών και ομοιοτήτων ανάμεσα στις διαμορφώσεις CLDC και CDC.



Εικόνα 1.4 Ο Κόσμος της J2ME (επίπεδα, διαμορφώσεις, προφίλ, προαιρετικά πακέτα).

Στο σχήμα είναι εμφανή τα επίπεδα της αρχιτεκτονικής όπως εδύχθησαν προηγούμενος, πλέον όμως σε μία πιο συγκεκριμένη μορφή.

Αξίζει να σημειωθούν δύο πράγματα.

- Η εικονική μηχανή που βρίσκεται στο χαμηλότερο επίπεδο είναι άρρηκτα συνδεδεμένη με το περιβάλλον εκτέλεσης. Αυτό σημαίνει ότι μια εικονική μηχανή είναι διαφορετική για linux από windows mobile, και διαφορετική ανάλογα με τον επεξεργαστή της συσκευής. Μια εικονική μηχανή υλοποιεί και εκτελεί μία διαμόρφωση με ένα συγκεκριμένο προφίλ. Ενώ με την σειρά τους τα προαιρετικά πακέτα απαιτούν αντίστοιχα συγκεκριμένες διαμορφώσεις ή/και προφίλ.
- Μεταξύ διαμορφώσεων και προφίλ δεν υπάρχει οριζόντια συμβατότητα. Αυτό σημαίνει ότι ένα MIDP προφίλ που έχει δημιουργηθεί για CLDC διαμόρφωση δεν μπορεί να εκτελεστεί σε μία διαμόρφωση CDC και το αντίστροφο. Αυτό ίσως φαίνεται περίεργο μια και όπως ήδη γράφτηκε η CLDC διαμόρφωση αποτελεί υποσύνολο της διαμόρφωσης CDC. Όντως αυτό είναι αληθές όσον αφορά τις κλάσεις και τα APIs, εκεί που γίνεται η διαφοροποίηση είναι στην εικονική μηχανή για την οποία προορίζεται το κάθε προφίλ και διαμόρφωση. Φυσικά θα μπορούσε να δημιουργηθεί MIDP προφίλ για CDC ίδιο με της διαμόρφωσης CLDC, δεν θα είχε όμως νόημα αφού δεν θα εκμεταλλευόταν πλήρως τις δυνατότητες της συγκεκριμένης διαμόρφωσης και άρα δεν θα εκπλήρωνε την προϋπόθεση της επέκτασης των βασικών δυνατοτήτων που παρέχει μία διαμόρφωση. Αντίθετα δεν θα μπορούσαν να υπάρξουν προφίλ αντίστοιχα με αυτά της CDC διαμόρφωσης λόγω περιορισμένων δυνατοτήτων που παρέχει η διαμόρφωση CLDC.

Σύνοψη

Η τεχνολογία της J2ME αποτελεί πόλο έλξης και μεγάλου ενδιαφέροντος για εταιρίες παραγωγής φορητών ή μικρών ηλεκτρονικών συσκευών, για τους τελικούς χρήστες αλλά και για την ίδια την Sun. Η ραγδαία ανάπτυξη της τεχνολογίας στον τομέα των φορητών συσκευών καθώς και οι νέες τεχνολογίες στον τομέα των δικτύων προσφέρουν πρόσφορο έδαφος για την συνεχόμενη ανάπτυξη και παγίωση της J2ME στον τομέα αυτό. Τα πλεονεκτήματα της γλώσσας προγραμματισμού Java τα οποία υπάρχουν και στην J2ME ως υποσύνολο αυτής δίνουν την δυνατότητα στις εταιρίες να δημιουργήσουν λογισμικό για τις συσκευές τους εύκολα και γρήγορα, σε τρίτους προγραμματιστές να αναπτύξουν μεγάλη γκάμα εφαρμογών αλλά ακόμη και οι τελικοί χρήστες να μπορούν να προσωποποιήσουν την συσκευή τους δημιουργώντας δικές τους εφαρμογές ή κατεβάζοντας από το διαδίκτυο νέες εφαρμογές για την συσκευή τους. Ήδη στο διαδίκτυο υπάρχει μεγάλη γκάμα από εφαρμογές γραμμένες σε MIDP (αφού το συγκεκριμένο προφίλ είναι το πιο διαδομένο λόγω των κινητών), ενώ παιχνίδια σε Java έχουν κατακλύσει κάθε φορητή συσκευή. Το μέλλον της J2ME προμηνύεται άκρως ενδιαφέρον και σίγουρα η παγκοσμία κοινότητα χρηστών είναι αυτή που θα βγει κερδισμένη.

ΚΕΦΑΛΑΙΟ 2

Λειτουργικά Συστήματα Φορητών Συσκευών

Όπως τα λειτουργικά συστήματα Ηλεκτρονικών Υπολογιστών, έτσι και τα λειτουργικά συστήματα φορητών συσκευών είναι μία πλατφόρμα λογισμικού πάνω στην οποία εκτελούνται άλλα προγράμματα. Με την αγορά της φορητής συσκευής ο κατασκευαστής έχει επιλέξει και εγκαταστήσει το λειτουργικό σύστημα της συσκευής. Το λειτουργικό σύστημα είναι υπεύθυνο για τον προσδιορισμό των λειτουργιών και χαρακτηριστικών που είναι διαθέσιμα στην συσκευή όπως το πληκτρολόγιο, πένακι αφής, συγχρονισμός με εφαρμογές, ηλεκτρονικό ταχυδρομείο, μηνύματα κειμένου κλπ. Το λειτουργικό σύστημα της φορητής συσκευής καθορίζει επίσης ποιες εφαρμογές τρίτων μπορούν να εγκατασταθούν στην συσκευή. Μερικά από τα πιο γνωστά λειτουργικά συστήματα φορητών συσκευών είναι τα ακόλουθα:

- **Symbian OS από την Symbian Ltd (μερίδιο αγοράς 46.6% το τρίτο τρίμηνο του 2008).** Μερικές από τις εταιρίες που χρησιμοποιούν το συγκεκριμένο λειτουργικό είναι η Nokia, BenQ, LG, Motorola, Samsung και Sony Ericson.



- **iPhone OS από την Apple Inc (μερίδιο αγοράς 17.3% το τρίτο τρίμηνο του 2008).** Τα iPhone και iPod Touch χρησιμοποιούν αυτό το λειτουργικό το οποίο προέρχεται από το λειτουργικό Mac OS X. Εφαρμογές τρίτων δεν υποστηρίζονταν επίσημα μέχρι την έκδοση του iPhone OS 2.0.



- **BlackBerry OS από την RIM (μερίδιο αγοράς 15.2% το τρίτο τρίμηνο του 2008).** Το συγκεκριμένο λειτουργικό επικεντρώνεται στην ευκολία χρήσης, και δημιουργήθηκε αρχικά για εταιρικούς σκοπούς.



- **Windows Mobile από την Microsoft (μερίδιο αγοράς 13.6% το τρίτο τρίμηνο του 2008).** Η πρώτη έκδοση ήταν η Windows CE με μεγάλη απήχηση στην Ασία. Αυτή την στιγμή υπάρχουν οι εκδόσεις Windows Mobile Professional (για συσκευές με οθόνη αφής) και Windows Mobile Standard, ενώ αναμένεται η νέα έκδοση Windows Mobile 6.5 .



- **Linux (μερίδιο αγοράς 5.1% το τρίτο τρίμηνο του 2008).** Το Linux χρησιμοποιείται ευρέως στην Κίνα από την Motorola και στην Ιαπωνία από την Docomo. Χρησιμοποιείται περισσότερο σαν πλατφόρμα βάσης για περισσότερες πλατφόρμες τρίτων όπως της Motorola, NEC, NTT,



DoCoMo, Panasonic, Samsung, Vodafone και TrollTech παρά σαν καθεαυτή πλατφόρμα.

- **Palm OS από την PalmSource (Εξαγοράστηκε από την ACCESS).** Η PalmSource παραδοσιακά χρησιμοποιούσε την δική της πλατφόρμα από την Palm Inc.



- **Binary Runtime Environment for Wireless (Brew).** Το BREW αναπτύχθηκε στην Αμερική από την Qualcomm και χρησιμοποιείται ευρέως στην Βόρεια Αμερική

και στην Ιαπωνία.

- **Android προωθείται από την Open Handset Alliance με άδεια της Google.** Το Android αναπτύχθηκε από την Google, ως πλατφόρμα που μπορεί να εκτελεστεί σε πολλές συσκευές. Ακόμα βρίσκεται στα αρχικά του βήματα.



Τα δεδομένα των μεριδίων αγοράς προέρχονται από την Canalys "Worldwide smart mobile device market, Canalys Q3 2008"

Λειτουργικά Συστήματα Φορητών Συσκευών και J2ME

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο η εικονική μηχανή είναι δεμένη με το περιβάλλον εκτέλεσης ενώ όπως αναφέρθηκε η Sun και η κοινότητα JCP δεν υλοποιεί αλλά δημιουργεί προδιαγραφές προς υλοποίηση. Το βάρος της υλοποίησης πέφτει στον εκάστοτε κατασκευαστή, ο οποίος θα αποφασίσει αν θα υλοποιήσει μία διαμόρφωση CDC ή CLDC για το λειτουργικό που έχει ενσωματώσει στην συσκευή του καθώς και ποιο ή ποια προφίλ με βάση την διαμόρφωση θα υποστηρίξει, ακόμα και ποια προαιρετικά πακέτα που ενδεχόμενος θα προσφέρει. Την παρούσα στιγμή οι περισσότεροι κατασκευαστές υλοποιούν διαμορφώσεις CLDC με προφίλ MIDP για τις κινητές συσκευές που κατασκευάζουν ενώ συνήθως δεν υλοποιούν προαιρετικά πακέτα ή υλοποιούν πακέτα ανάλογα με τα χαρακτηριστικά που προσφέρουν στην συσκευή τους. Τέτοιο παράδειγμα αποτελεί η NOKIA η οποία έχει υλοποιήσει και αναπτύξει το Location API (JSR-179) για τις συσκευές τις που φέρουν GPS το οποίο και προσφέρει. Σχεδόν κάθε συσκευή NOKIA των τελευταίων χρόνων υποστηρίζει Java CLDC MIDP με το Location API.

Σε αυτό το σημείο προκύπτουν κάποια ερωτήματα, όπως το αν μπορεί μία συσκευή με λειτουργικό σύστημα (όπως τα προηγούμενα) να υποστηρίξει J2ME όταν ο κατασκευαστής δεν έχει υλοποιήσει ή αν μπορεί ο χρήστης να προσθέσει μια άλλη υλοποίηση διαφορετικής διαμόρφωσης αντί αυτής που έχει ο κατασκευαστής της συσκευής. Η ίδια η SUN δεν προσφέρει υλοποίηση εικονικής μηχανής (όπως στους σταθερούς και φορητούς υπολογιστές), κάτι που είναι φυσικό αφού τα λειτουργικά συστήματα των φορητών συσκευών είναι πολλά ενώ και η αρχιτεκτονική από συσκευή σε συσκευή διαφέρει. Το πρόβλημα είναι εντονότερο σε φορητές συσκευές όπως τα PDA αφού τα περισσότερα κινητά προσφέρουν υλοποίηση J2ME.

Εικονικές Μηχανές και Υλοποιήσεις Διαμορφώσεων Τρίτων Κατασκευαστών

Το κενό της υποστήριξης J2ME σε συσκευές που δεν έχει υλοποιήσει ο κατασκευαστής αλλά μπορούν να την υποστηρίξουν έχουν αναλάβει τρίτες εταιρίες ή κοινότητες. Στην περίπτωση των εταιριών η απόκτηση μιας εικονικής μηχανής που υλοποιεί κάποια διαμόρφωση και άρα επιτρέπει εκτέλεση J2ME δεν είναι δωρεάν ενώ στοχεύει περισσότερο σε κατασκευαστές συσκευών από ότι σε τελικούς χρήστες. Αντίθετα στην περίπτωση ανάπτυξης εικονικής μηχανής από χρήστες ή κοινότητες που γίνεται με ανοικτό και δωρεάν χαρακτήρα έως τώρα δεν υπάρχει κάποια υλοποίηση που να προσφέρει τα ποιοτικά χαρακτηριστικά και την ευκολία που προσφέρει μία εμπορική υλοποίηση. Πάντως μέχρι και σήμερα δεν υπάρχει κάποια πλήρως ολοκληρωμένη δωρεάν υλοποίηση είτε από εταιρία είτε από κάποιο χρήστη ή κοινότητα. Παρακάτω θα γίνει μία παρουσίαση των κυριότερων εικονικών μηχανών που κυκλοφορούν.

- **IBM WebSphere Everyplace Micro Environment.**

Η υλοποίηση της IBM (γνωστή και με το όνομα της εικονικής μηχανής που χρησιμοποιεί **J9**) παρέχει την δυνατότητα ανάπτυξης και εκτέλεσης εφαρμογών σε φορητές συσκευές.




- Παρέχει μικρό αποτύπωμα σε φορητές συσκευές όπως κινητά, PDAs, παιχνιδιομηχανές και άλλες φορητές συσκευές για σύνδεση με εταιρικές εφαρμογές.
- Υλοποιεί ανοικτά πρότυπα και εικονική μηχανή Java Micro Edition (J2ME).
- Παρέχει δυνατότητες παραμετροποίησης για βελτιστοποίηση μνήμης και απόδοσης.
- Υποστηρίζει διαμόρφωση CLDC 1.1 και προφίλ MIDP 2.0
- Υποστηρίζει διαμόρφωση CDC 1.0, 1.1 και προφίλ Foundation Profile και Personal Profile.
- Δεν διατίθεται δωρεάν (παρέχεται σε μεγάλες ποσότητες για κατασκευαστές 10.000+).
- Υποστηρίζει τα λειτουργικά συστήματα Neutrino RTOS (QNX Software Systems), Palm Operating System, Windows.

- **NsiCom CrEme.** Η CrEme είναι ίσως η καλύτερη εικονική μηχανή για PDA και φορητές συσκευές που έχουν εγκατεστημένο λειτουργικό MS Windows Mobile.



- Η εικονική μηχανή CrEme 4 υλοποιεί την διαμόρφωση CDC και το προφίλ Personal Profile 1.0.
- Υποστηρίζει Swing. Η προεπιλεγμένη και προτεινόμενη γραφική βιβλιοθήκη βασίζεται σε υλοποίηση του AWT από την Truffle. Ενώ προσφέρεται εναλλακτικά το tiny AWT.
- Είναι βελτιστοποιημένη και υποστηρίζει πλήρως γλώσσες μη δυτικού τύπου.
- Συμβατή με JDK 1.3.1

- Διατίθεται plug-in που προσθέτει υποστήριξη λειτουργίας των Java Applets σε Internet Explorer.
 - Υποστηρίζει βιβλιοθήκες τρίτων κατασκευαστών όπως είναι η γραφική βιβλιοθήκη SWT.
 - Δεν διατίθεται δωρεάν (1000\$ για 40 άδειες ή 100\$ για προσωπική άδεια χρήσης)
- **EsmerTec Jbed.** Είναι μία υλοποίηση εικονικής μηχανής που υποστηρίζει CDC διαμόρφωση και προφίλ Personal Profile.



 - Περιβάλλον εκτέλεσης συμβατό με CDC/PP.
 - Plug-in για Pocket Internet Explorer με υποστήριξη Java Applets χωρίς υποστήριξη όμως JavaScript.
 - Έκδοση JBed pJava Virtual Machine με συμβατότητα JDBC και JRMI.
 - Δεν διατίθεται δωρεάν (49.90\$ ατομική άδεια).
- **Mysaifu JVM.** Αποτελεί μία δωρεάν λύση για φορητές συσκευές με λειτουργικό σύστημα windows mobile. Βρίσκεται σε συνεχή ανάπτυξη και ενημέρωση από την ομάδα ανάπτυξης.
 - **PhoneME feature και PhoneME Advanced από Java.net Community.** Αποτελούν τις δύο πιο ελπιδοφόρες προσπάθειες για υλοποίηση CLDC και CDC αντίστοιχα. Και τα δύο είναι ανοικτού κώδικα και διατίθενται δωρεάν. Αυτή την στιγμή βρίσκονται στη έκδοση PhoneMe feature (MR4 – Milestone Release) και PhoneMe Advanced (MR2 – Milestone Release). Σκοπός του έργου PhoneME είναι να υλοποιηθούν όλες οι διαμορφώσεις, τα προφίλ αλλά και τα περισσότερα προαιρετικά πακέτα έτσι ώστε να παρέχονται οι μέγιστες δυνατότητες στους προγραμματιστές. Μέχρι και την τελευταία έκδοση και επειδή το έργο είναι σε συνεχή ανάπτυξη ο τρόπος εγκατάστασης στις συσκευές δεν είναι εύκολος και απαιτεί χρήση εργαλείων και γνώσεις scripting σε αντίθεση με τις εμπορικές υλοποιήσεις όπου η εγκατάστασή τους είναι μία απλή αντιγραφή-επικόλληση. Παρόλα αυτά το συγκεκριμένο έργο αποτελεί την μελλοντική λύση για μία πραγματικά δωρεάν J2ME.

ΚΕΦΑΛΑΙΟ 3

E-kones Experience (Εργαλεία Ανάπτυξης)



Η εργασία που υλοποιήθηκε είναι μία έκδοση χρήστη για το λογισμικό E-kones. Εξετάζεται το κομμάτι των χρηστών που έχουν στην διάθεση τους φορητές συσκευές υψηλών δυνατοτήτων με προηγμένα λειτουργικά συστήματα εγκατεστημένα (windows mobile). Η γλώσσα που επιλέχθηκε για την ανάπτυξη του λογισμικού ήταν η Java Micro Edition (συμβατή με το πρότυπο CLDC-Midp2.0) και σαν Virtual Machine στην συσκευή χρησιμοποιήθηκε το J9 της IBM ενώ η εφαρμογή κτίστηκε πάνω στο LWUIT toolkit. Ως υλικό ανάπτυξης και δοκιμών χρησιμοποιήθηκε ένα PDA της Mio (P560). Η εργασία αυτή χρησιμοποίησε το Static Google Maps API για την υποστήριξη γεωγραφικών χαρτών ενώ κατασκευάστηκε με την χρήση περιβάλλοντος ανάπτυξης NetBeans IDE έκδοση 6.5.

MioP560



Συσκευή δοκιμών με εγκαταστημένα Window Mobile 6.0, οθόνη αφής με πένακι, Wifi και Bluetooth και δυνατότητα σύνδεσης με υπολογιστή μέσω σύνδεσης mini USB.

Η συσκευή MioP560 που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής έχει ενσωματωμένο ολοκληρωμένο κύκλωμα GPS από την Sirf (StartIII), υποστηρίζει τα πρωτόκολλα NMEA, SIRF Binary, AI3/F.

Πρωτόκολλο NMEA

Το NMEA είναι ένας συνδυασμός προδιαγραφών ηλεκτρονικών και δεδομένων για επικοινωνία μεταξύ θαλάσσιων ηλεκτρονικών συσκευών όπως αυτόματους πιλότους, γυροσκοπικές πυξίδες, σόναρ, GPS δέκτες καθώς και πολλών άλλων οργάνων. Έχει καθοριστεί και ελέγχεται από τον Αμερικάνικο Οργανισμό [National Marine Electronics Association](http://www.nmea.org).



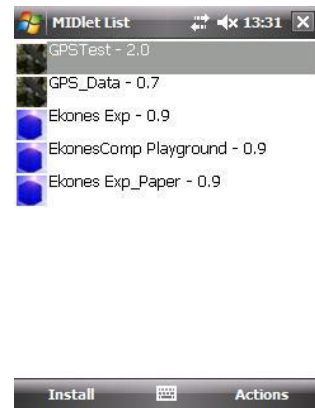
Το NMEA είναι ένα απλό σειριακό ASCII πρωτόκολλο επικοινωνίας που καθορίζει πως θα μεταδίδονται τα δεδομένα σε μορφή πρότασης, από ένα πομπό σε πολλαπλούς δέκτες. Στο επίπεδο εφαρμογών, το πρότυπο καθορίζει τα περιεχόμενα κάθε πρότασης (μηνύματος) έτσι ώστε να μπορούν όλοι οι ακροατές να τα αναγνώσουν με ακρίβεια.

- Σε κάθε μήνυμα ο χαρακτήρας εκκίνησης είναι το \$.
- Οι επόμενοι πέντε χαρακτήρες προσδιορίζουν τον πομπό (δύο χαρακτήρες) και τύπος μηνύματος (τρεις χαρακτήρες).
- Τα πεδία δεδομένα που ακολουθούν χωρίζονται με κόμμα (,).
- Ο πρώτος χαρακτήρας που ακολουθεί αμέσως μετά το τελευταίο πεδίο δεδομένων είναι ο χαρακτήρας αστερίσκος (*).
- Ο αστερίσκος ακολουθείται αμέσως από ένα έλεγχο αθροίσματος δύο ψηφίων που παριστά δεκαεξάρικο αριθμό. Ο έλεγχος αθροίσματος είναι το αποκλειστικό OR για όλους του χαρακτήρες μεταξύ '\$' και '*'. Σύμφωνα με την επίσημη προδιαγραφή ο έλεγχος αθροίσματος είναι προαιρετικός για τις περισσότερες προτάσεις δεδομένων.
- [<CR><LF>](#) κλείνουν το μήνυμα.

Με την χρήση αυτού του πρωτοκόλλου όπως θα αναφερθεί παρακάτω έχει γίνει υλοποίηση κλάσεων για την ανάγνωση μηνυμάτων GPS.

IBM J9 MIDP 2.0 Java Emulator V2.3

Στο προηγούμενο κεφαλαίο έγινε μία συνοπτική αναφορά στην εικονική μηχανή J9 της IBM. Στην εργασία αυτή χρησιμοποιήθηκε η έκδοση 2.3 της J9 σαν περιβάλλον εκτέλεσης. Η συγκεκριμένη ήταν η μοναδική δωρεάν έκδοση της IBM, η οποία πλέον έχει αποσυρθεί. Νεότερη έκδοση διατίθεται πλέον μόνο για αγορά από την IBM. Αξίζει να σημειωθεί ότι λόγω παλαιότητας της συγκεκριμένης έκδοσης υπήρχε υποστήριξη μέχρι της έκδοση 2.0 του προφίλ MIDP και όχι του τρέχων 2.1, ενώ δεν υπήρχε υλοποίηση κανενός προαιρετικού πακέτου. Το συγκεκριμένο περιβάλλον εκτέλεσης θεωρείται απλό στην χρήση του ενώ δυστυχώς στον τομέα της συμβατότητας με το LWUIT (εργαλείο για το κτίσιμο της εφαρμογής) εμφανίστηκαν αρκετά προβλήματα.



Στο σημείο αυτό αξίζει να σημειωθεί ότι στα πλαίσια της έρευνας για επιλογή εικονικής μηχανής, εκτός από την J9 που επιλέχτηκε δοκιμάστηκαν ή έγιναν προσπάθειες δοκιμής για:

- NsiCom CreMe, δοκιμάστηκε η τελευταία της έκδοση αλλά σε δοκιμαστική μορφή για 40 ημέρες με πολύ θεαματικά αποτελέσματα. Παρόλα αυτά η αγορά της στα πλαίσια αυτής της εργασίας δεν κρίθηκε απαραίτητη.
- Esmertec JBED, η συγκεκριμένη εικονική μηχανή δεν κατάφερε ποτέ να εκτελεστεί λόγω ασυμβατότητας με την συσκευή.
- PhoneMe Advanced, εγκαταστάθηκε και λειτούργησαν τα δοκιμαστικά προγράμματα που περιελάμβανε. Παρόλα αυτά δεν μπόρεσαν να εκτελεστούν άλλες εφαρμογές, οπότε και περισσότερη ενασχόληση με

την συγκεκριμένης υλοποίηση στα πλαίσια αυτής της πτυχιακής δεν θεωρήθηκε απαραίτητη.

MIDP 2.0 (JSR-118)

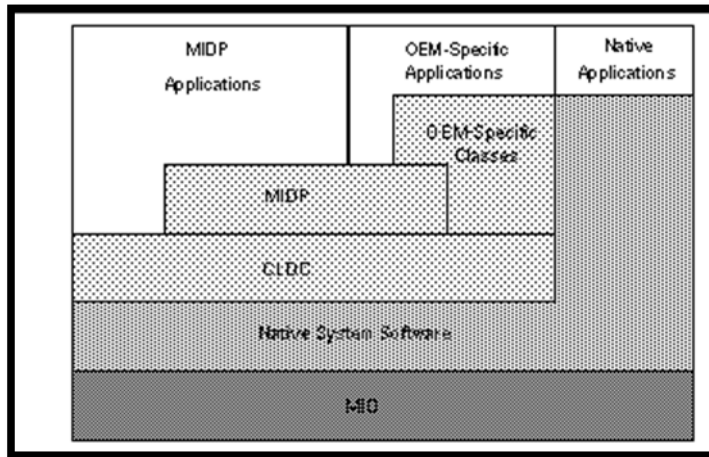
Το προφίλ Midp 2.0 εξετάστηκε συνοπτικά στο κεφάλαιο 1. Στο σημείο αυτό θα δοθεί μία εκτενέστερη ανάλυση στην αρχιτεκτονική του. Το προφίλ Midp 2.0, επεκτείνει την πρώτη έκδοση ενώ διατηρεί προς τα πίσω συμβατότητα. Αυτό σημαίνει ότι εφαρμογές γραμμένες σε προφίλ Midp 1.0 μπορούν να εκτελεστούν στην έκδοση 2.0. Το συγκεκριμένο προφίλ λειτουργεί πάνω σε διαμόρφωση CLDC 1.0 (JSR-30) ή μεταγενέστερη.

Οι φορητές συσκευές παρέχουν μία πληθώρα δυνατοτήτων. Για να μπορέσει να διατηρηθεί οριζόντια συμβατότητα ανάμεσα στις περισσότερες συσκευές οι προδιαγραφές Midp 1.0 (JSR-37) και Midp 2.0 (JSR-118) επικεντρώνονται και διευθετούν μόνο τις βασικότερες όπως

- Παραλαβή και Χρέωση εφαρμογών
- Κύκλος Ζωής Εφαρμογής
- Μοντέλο υπογραφής εφαρμογών και μοντέλο ασφαλείας για τομείς δικαιωμάτων
- Ασφάλεια μεταξύ άκρων (https)
- MIDlet push registration (server push model)
- Δικτύωση
- Μόνιμη Αποθήκευση
- Ήχος
- Χρονοδιακόπτες
- Διεπαφή Χρήστη (συμπεριλαμβανομένου συσκευές εισόδου καθώς και απαιτήσεις παιχνιδιών)

Αρχιτεκτονική

Στην υψηλού επιπέδου αρχιτεκτονική (εικόνα 3.1), από άποψη υλικού στο χαμηλότερο επίπεδο (MID) βρίσκεται η συσκευή. Ακριβώς από πάνω βρίσκεται το επίπεδο λογισμικού συμπεριλαμβανομένου και του λειτουργικού καθώς και όλες τις βιβλιοθήκες που βρίσκονται την συσκευή. Στο επόμενο επίπεδο από αριστερά προς τα δεξιά είναι το CLDC, στο οποίο βρίσκεται η εικονική μηχανή και η πρόσθετες βιβλιοθήκες που καθορίζονται από την διαμόρφωση CLDC. Αυτό το επίπεδο λειτουργεί ως το υπο-επίπεδο πάνω στο οποίο μπορούν να κτιστούν υψηλότερου επιπέδου APIs.



Εικόνα 3.1 Όψη της Αρχιτεκτονικής Υψηλού Επιπέδου

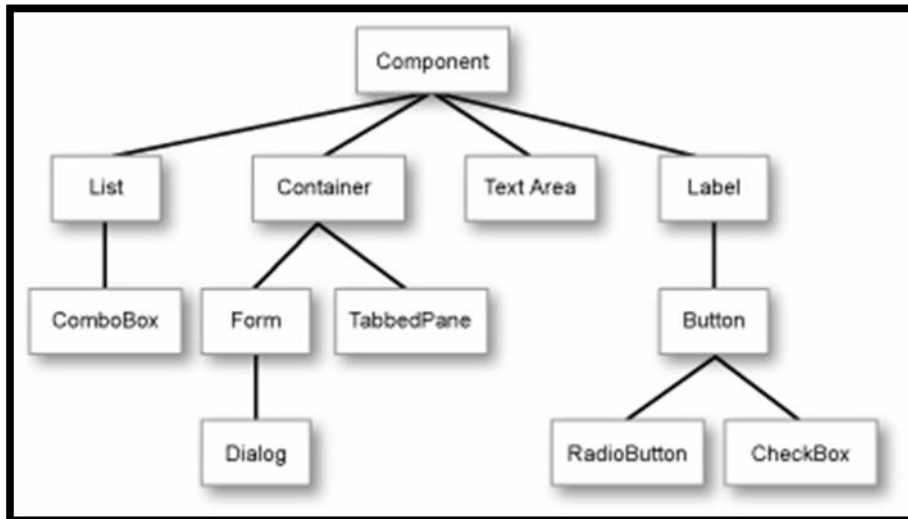
Δύο κατηγορίες από APIs εμφανίζονται πάνω από το CLDC

- Τα APIs του MIDP προφίλ
- APIs προσδιορισμένα από τρίτους κατασκευαστές που επεκτείνουν κάθετα τις δυνατότητες μιας συσκευής ή ενός συνόλου συσκευών, τέτοια APIs είναι τα προαιρετικά πακέτα (JSRs).

LWUIT Toolkit

Η εργαλειοθήκη LWUIT είναι εμπνευσμένη από το SWING και απευθύνεται σε φορητές συσκευές που υποστηρίζουν είτε την CLDC είτε την CDC διαμόρφωση. Υποστηρίζει δυνατότητες πολλαπλών θεμάτων, ιεραρχία αντικειμένων, υποδοχέα και αφαιρετικότητα από την βιβλιοθήκη GUI του υποστρώματος. Ο όρος ελαφριά (lightweight) υποδηλώνει ότι τα αντικείμενα στην βιβλιοθήκη σχεδιάζουν την κατάσταση τους στον πηγαίο κώδικα της Java χωρίς γηγενή ερμηνεία από το σύστημα. Εσωτερικές διασυνδέσεις και αφηρημένες κλάσεις παρέχουν αφαιρετικότητα στις διασυνδέσεις και APIs του προφίλ που βρίσκεται στο υπόστρωμα. Αυτό επιτρέπει μεταφορά για τα τρέχοντα προφίλ αλλά και τα μελλοντικά.

Το LWUIT υλοποιεί τις κλάσεις προδιαγραφής του προφίλ Midp σε κάτι που θυμίζει πάρα πολύ έντονα Swing. Με αποτέλεσμα να προσφέρει στον προγραμματιστή οικία αντικείμενα διεπαφής (εικόνα 3.2) όπως ComboBox, Checkbox, Label, Button, List, Dialogs, RadioButton, Textbox, TextArea, ενώ υλοποιεί την έννοια του γεγονότος (event) όπως την αντιλαμβάνονται οι προγραμματιστές από το Swing με την χρήση ActionListeners.



Εικόνα 3.2 Απλοποιημένη Ιεραρχία των τμημάτων του LWUIT.

Παράλληλα όμως προσφέρει και δικά του χαρακτηριστικά και διαφοροποιήσεις από τις κλασικές εργαλειοθήκες όπως το Swing και AWT. Στο LWUIT ο κεντρικός υποδοχέας (Top Level) είναι η φόρμα (Form) πάνω στην οποία τοποθετούνται τα αντικείμενα, ενώ συνδέεται με την κλάση Display που είναι υπεύθυνη για την αποστολή και παραλαβή των γεγονότων που παράγει ο χρήστης.

Google Maps API

Η επιτυχία του Google Earth και google maps, καθώς και η ζήτηση για προβολή χαρτών από ιστοσελίδες τρίτων, οδήγησε την Google στο να δημιουργήσει και να προσφέρει το google maps API. Το συγκεκριμένο API επιτρέπει σε ιστοσελίδες να εμφανίζουν χάρτες του google maps με την χρήση javascript. Αντίστοιχα δημιουργήθηκε και το Static google Maps API, το οποίο δεν απαιτεί javascript ή κάποιον άλλο δυναμικό τρόπο φόρτωσης σελίδας. Στην εργασία αυτή χρησιμοποιήθηκε το Static Google Maps API.

Για να υπάρξει πρόσβαση στην υπηρεσία Static Google Maps API απαιτείται η δημιουργία αναγνωριστικού κλειδιού χρήστη, το οποίο μπορεί να αποκτηθεί με εγγραφή. Κατόπιν η εφαρμογή μπορεί να αιτείται χάρτες με την χρήση ενός υπερσυνδέσμου και του κλειδιού αυτού.

Μία αίτηση προς την υπηρεσία Static Google Maps φαίνεται παρακάτω.

`http://maps.google.com/staticmap?center=40.714728,-73.998672&format=png8&zoom=16&size=128x128&maptype=mobile&key=MAPS_API_KEY&sensor=true`

Αυτός ο υπερσύνδεσμος μπορεί να αναλυθεί στα παρακάτω τμήματα.

<code>http://maps.google.com/staticmap?</code>	Διεύθυνση της υπηρεσίας
<code>center=40.714728,-73.998672</code>	Επιθυμητό κέντρο χάρτη
<code>format=png8</code>	Τύπος εικόνας χάρτη
<code>zoom=16</code>	Επιθυμητό επίπεδο μεγέθυνσης

size=128x128	Επιθυμητό μέγεθος εικόνας
maptype=mobile	Τύπος εικόνας
key=MAPS_API_KEY	Κλειδί χρήστη για την υπηρεσία
sensor=true	Ύπαρξη συσκευής GPS

Στον παραπάνω σύνδεσμο τα ορίσματα χωρίζονται μεταξύ τους με το σύμβολο '&'. Παρακάτω θα γίνει ανάλυση των σημαντικότερων ορισμάτων.

- **format=png8**, δηλώνει τον τύπο του αρχείου εικόνας που επιθυμεί ο χρήστης να χρησιμοποιήσει. Υποστηριζόμενοι τύποι είναι οι gif,jpg,jpg-aseline,png8,png32. Η εφαρμογή που υλοποιήθηκε κάνει χρήση του png8, που είναι και ο μόνος τύπος εικόνας που λειτουργούσε με το LWUIT και Midp 2.0.
- **zoom=16**, επιθυμητό επίπεδο μεγέθυνσης, μικρότερη τιμή είναι το 0 και μέγιστη τιμή η 19 (έκδοση 3). Να σημειωθεί ότι την περίοδο που δημιουργήθηκε η εφαρμογή η παλαιότερη έκδοση της υπηρεσία Google Static Maps υποστήριζε μέγιστο επίπεδο μεγέθυνσης την τιμή 17, η οποία θεωρείται και μέγιστη για αυτή την εργασία όπως θα παρουσιασθεί παρακάτω.
- **size=128x128**, Επιθυμητή ανάλυση εικόνας, με μέγιστη τιμή 640X640. Το LWUIT και η υλοποίηση J9, κατάφερε να λειτουργήσει με μέγιστη ανάλυση 144X144 για συγκεκριμένο όμως τύπο.
- **maptype=mobile**, Η υπηρεσία static google Maps προσφέρει τέσσερις τύπους χαρτών (roadmap,mobile, terrain, hybrid, satellite).

Με βάση τα προηγούμενα δημιουργήθηκαν κλάσεις που να υποστηρίζουν την λειτουργία της υπηρεσίας στην συσκευή. Παρόλα αυτά όπως θα αναλυθεί παρακάτω λόγω τεχνικών προβλημάτων δεν ήταν εφικτή η δημιουργία λειτουργίας που θα επέτρεπε στον χρήστη την παραμετροποίηση του χάρτη της υπηρεσίας Google Maps.

NetBeans IDE



Το περιβάλλον ανάπτυξης NetBeans 6.5 αποτελεί ένα από τα πιο σύγχρονα και ολοκληρωμένα περιβάλλοντα ανάπτυξης για την Java και όχι μόνο. Για την J2ME υπάρχουν επιπρόσθετα plug-ins που προσφέρουν περισσότερες δυνατότητες καθώς και έτοιμο προσχέδιο για την ανάπτυξη εφαρμογών Midp. Πολλοί κατασκευαστές φορητών συσκευών παρέχουν εξομοιωτή συσκευής που λειτουργεί μέσω του NetBeans αλλά και η ίδια η SUN παρέχει το Java ME SDK 3.0 (<http://java.sun.com/javame/downloads/sdk30.jsp>). Ενώ η τελευταία έκδοση του NetBeans 6.7 (βρίσκεται σε τελικό δοκιμαστικό στάδιο) περιλαμβάνει ακόμα περισσότερα χαρακτηριστικά της J2ME ενσωματωμένα.

ΚΕΦΑΛΑΙΟ 4

Αρχιτεκτονική E-kones Experience



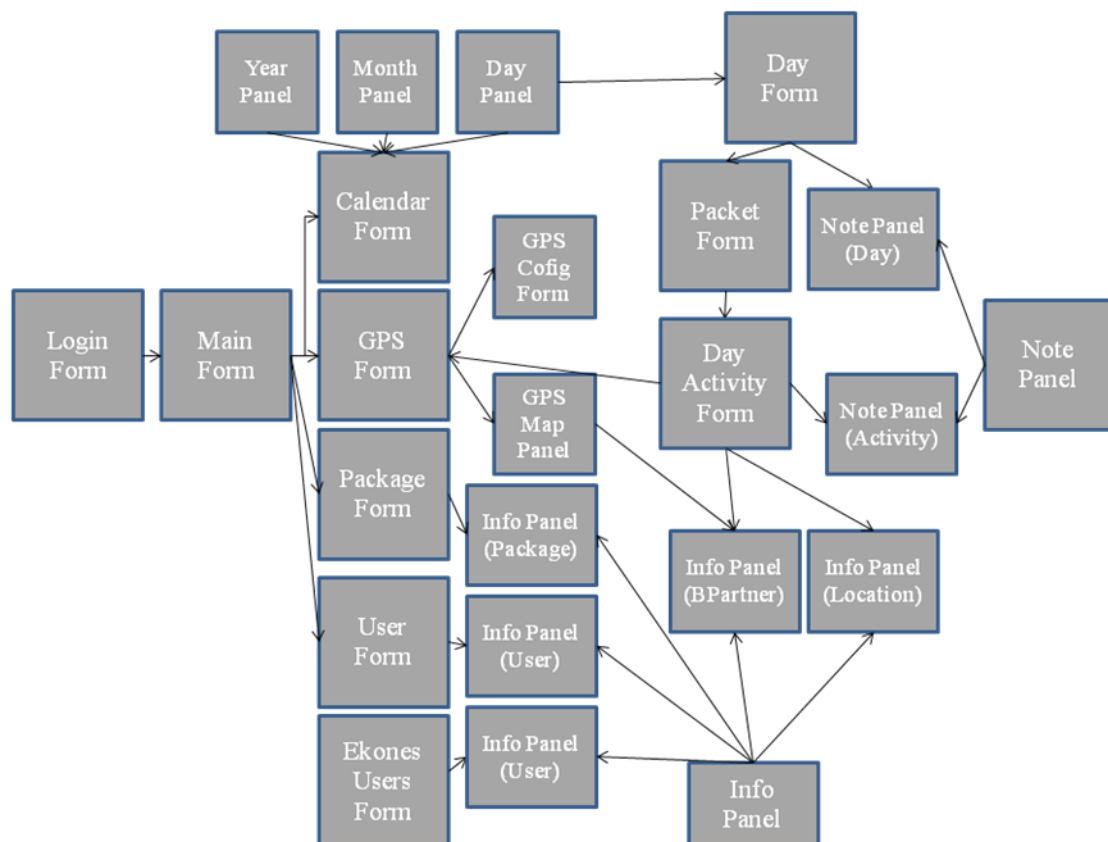
Η εργασία εξετάζει την υποστήριξη με λογισμικό του χρήστη που έχει αγοράσει πακέτο/α από την υπηρεσία του E-kones και πλέον «ζει» το εκδρομικό πακέτο που αγόρασε. Έτσι εξηγείται το όνομα E-kones “Experience”.

Για την υλοποίηση λήφθηκαν υπόψη τα χαρακτηριστικά μίας φορητής συσκευής (όπως μικρό μέγεθος οθόνης, περιορισμένη μνήμη). Η προσπάθεια επικεντρώθηκε στο να δημιουργηθεί μια εφαρμογή σε πρότυπα desktop και όχι σε τύπου κινητού. Για αυτό τον λόγο αναπτύχθηκαν συστατικά διεπαφής (components) που να εξομοιώνουν τέτοια συμπεριφορά. Το αποτέλεσμα είναι μία υβριδική εφαρμογή που σε ορισμένα σημεία εκμεταλλεύεται όμορφα χαρακτηριστικά από τα κινητά (βλέπε κεντρικό μενού), ενώ προσπαθεί να προσφέρει αντικείμενα όπου είναι κοινότυπα σε κλασικές διεπαφές (combobox, popup, menus etc). Ένα άλλο χαρακτηριστικό αυτής της εργασίας είναι ότι έχει γίνει προσπάθεια τα αντικείμενα που σχετίζονται με την γραφική παρουσίαση να είναι ομαδοποιημένα σε κλάσεις για να είναι εύκολη η αλλαγή τους έτσι ώστε να μπορεί να υποστηριχθεί και δυνατότητα themes ακόμα και μέσω web (αλλαγής κουμπιών, χρωμάτων κλπ).

Για τις ανάγκες της εφαρμογής αναπτύχθηκαν τα εξής απαραίτητα στοιχεία.

- XML Parser (Reader)
- Custom Components (MenuItem, ComboBox, PopUp, RateItem, Container)
- Calendar
- Map

Η εφαρμογή είναι κτισμένη γύρω από δύο κεντρικά αντικείμενα το Ημερολόγιο (Calendar) και τον Χάρτη (GPS/Map), ενώ όλα τα υπόλοιπα αποτελούν υποστηρικτικά αυτών. Θα μπορούσαμε να φανταστούμε ότι το αντικείμενο του Ημερολογίου έχει τον χαρακτήρα της «Χρονικής» τοποθέτησης του χρήστη μέσα στο πακέτο ενώ το αντικείμενο του Χάρτη έχει τον χαρακτήρα της «Χωρικής» τοποθέτησης. Στο σχεδιάγραμμα που ακολουθεί (εικόνα 4.1) παρουσιάζεται η αρχιτεκτονική της εφαρμογής όπως είναι στην τελική πρωτότυπη έκδοση.



Εικόνα 4.1 Η αρχιτεκτονική της εφαρμογής στην τελική πρωτότυπη έκδοση

Αντικείμενα της εφαρμογής

- **IstlContainer**
- **IstlLayout**
- **IstlComboBox**
- **IstlMenuItem**
- **IstlLabel**
- **IstlPopUp**
- **IstlRateItem**

IstlContainer

Το αντικείμενο `istlContainer` επεκτείνει τον υποδοχέα `Container` του `LWUIT`. Η μόνη διαφορά με τον `Container` του `LWUIT` είναι ότι μπορεί να έχει ακροατές γεγονότων. Ο λόγος που αναπτύχθηκε είναι για να μπορεί να αντιληφθεί γεγονότα της «πένας» τα οποία χρειάζονται για την εκπομπή της θέσης της πένας πάνω στην οθόνη. Περιλαμβάνει δύο μεθόδους.

Τύπος Επιστροφής Μεθόδου	Όνομα Μεθόδου	Χρήση
int	<code>getXcor()</code>	Θέση X συντεταγμένης στην οθόνη
int	<code>getYcor()</code>	Θέση Y συντεταγμένης στην οθόνη

Ο `istlContainer` χρησιμοποιείται από την εφαρμογή κυρίως στο αντικείμενο `istlMap`.

IstlLayout

Ο διαχειριστής διάταξης αναλαμβάνει να τοποθετήσει τα αντικείμενα σε θέση που ορίζει ο χρήστης. Με την χρήση των μεθόδων `setX()` και `setY()`. Ο διαχειριστής διάταξης `istlLayout` χρησιμοποιείται ευρέως από την εφαρμογή για την τοποθέτηση των περισσότερων αντικειμένων πάνω στους υποδοχείς. Κυρίως όμως χρησιμοποιείται από το αντικείμενο του ημερολογίου όπου αναλαμβάνει να τοποθετήσει τα αντικείμενα πάνω στους αντίστοιχους υποδοχείς, από το αντικείμενο `istlMap` όπου επιτρέπει την τοποθέτηση σημάνσεων σε οποιαδήποτε περιοχή πάνω στον υποδοχέα καθώς και από τα αντικείμενα `istlPopUp`, `istlMenuItem`, `IstlCombobox` για την τοποθέτησή τους.

IstlLabel

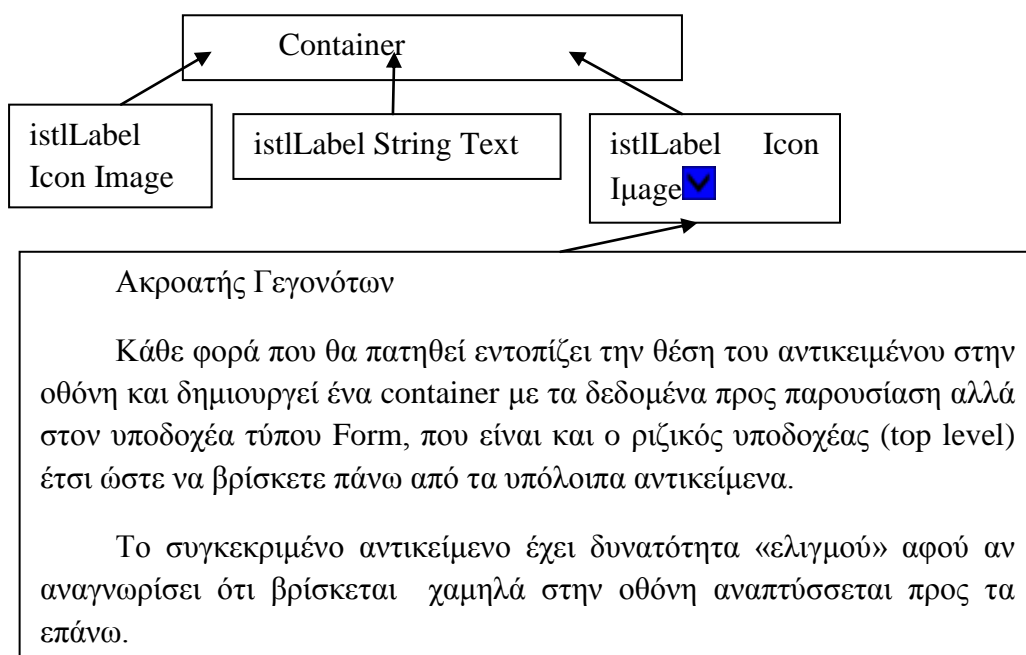
Το αντικείμενο `istlLabel` είναι ένα υβριδικό αντικείμενο που προέκυψε από την διασταύρωση του αντικειμένου `Κουμπιού (Button)` και `Ετικέτας (Label)`, για την ακρίβεια επεκτείνει το αντικείμενο `Label`.

Τύπος Επιστροφής Μεθόδου	Όνομα Μεθόδου	Χρήση
int	<code>getXcor()</code>	Θέση X συντεταγμένης

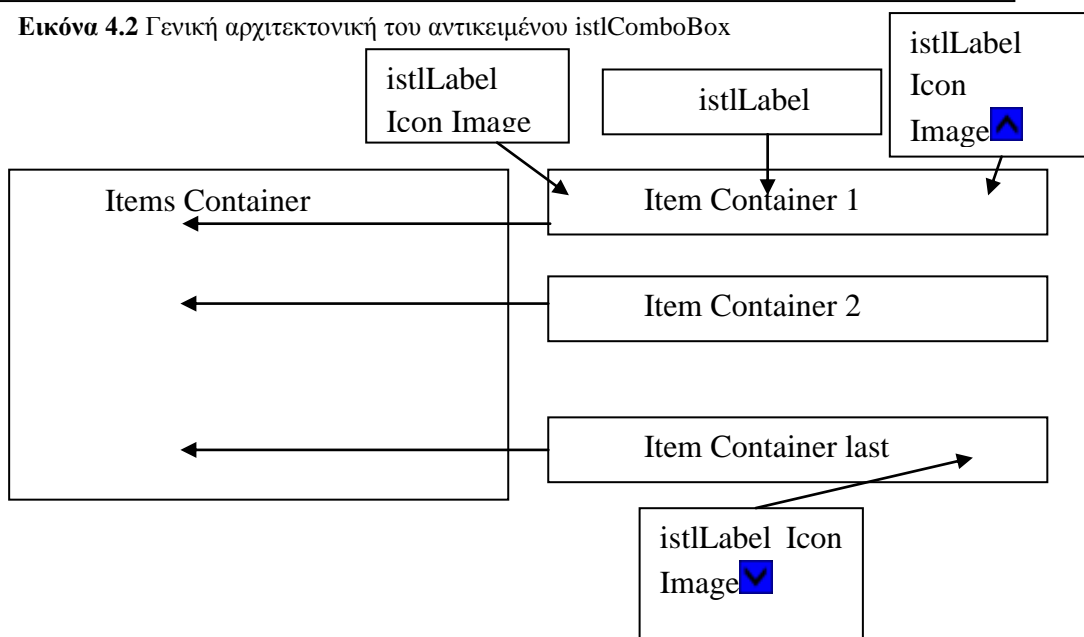
		στην οθόνη
int	getYcor()	Θέση Y συντεταγμένης στην οθόνη
void	setLabelid(int lid)	Εκχωρεί αναγνωριστική τιμή (Id) στο αντικείμενο
int	getLabelId()	Επιστρέφει την αναγνωριστική τιμή (Id) του αντικειμένου

IstlComboBox


Το συγκεκριμένο αντικείμενο επεκτείνει το αντικείμενο Container και υλοποιείται με την χρήση ήδη υπάρχων αντικειμένων.



Εικόνα 4.2 Γενική αρχιτεκτονική του αντικειμένου istlComboBox



Εικόνα 4.3 Γενική αρχιτεκτονική του αντικειμένου istlComboBox σε κατάσταση ανάπτυξης

Τα σχεδιαγράμματα (εικόνα 4.2 και 4.3) παρουσιάζουν την αρχιτεκτονική του `istlComboBox`. Στο πρώτο σχεδιάγραμμα (εικόνα 4.2) είναι το ίδιο το αντικείμενο, ενώ στο δεύτερο σχεδιάγραμμα (εικόνα 4.3) είναι η αρχιτεκτονική του ανεπτυγμένου τμήματος που εμφανίζεται μόλις πατηθεί το κουμπί βέλος .

Στη συγκεκριμένη έκδοση το πλήθος των επιλογών σε πτυσσόμενη κατάσταση είναι 4 το οποίο δεν μεταβάλλεται με κάποια μέθοδο. Παρόλα αυτά υπάρχει ράβδος ολίσθησης (scrollbar) στα δεξιά η οποία ενεργοποιείται όταν ανιχνεύσει περισσότερες των 4 επιλογών και με δυνατότητα ορίων.

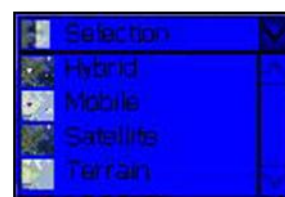
Για να υλοποιηθεί ένα `istlComboBox` χρησιμοποιείται μια κλάση προδιαγραφής, που περιέχει το αντικείμενο κάθε επιλογής. Η κλάση αυτή αναγνωρίζει αυτόματα αν η επιλογή θα είναι εικόνα, κείμενο ή και τα δύο. Παρέχοντας την δυνατότητα στο να υπάρχουν ταυτόχρονα επιλογές μόνο με εικόνα, μόνο κείμενο ή και τα δύο στο ίδιο `combobox`.

Χρήση

```
Vector mpt = new Vector();
istlComboModelItem cmbi = new
istlComboModelItem(st.getMapTypeHybridIco(), "Hybrid");
mpt.addElement(cmbi);
cmbi = new istlComboModelItem(st.getMapTypeMobileIco(), "Mobile");
mpt.addElement(cmbi);
cmbi = new
istlComboModelItem(st.getMapTypeSatteliteIco(), "Satellite");
mpt.addElement(cmbi);
cmbi = new istlComboModelItem(st.getMapTypeTerrainIco(), "Terrain");
mpt.addElement(cmbi);

maptypes = new istlComboBox(mpt);
maptypes.setDefaultIcon(st.getMapTypeIco());
```

Ο παραπάνω κώδικας δημιουργεί το `combobox` για την επιλογή τύπου χάρτη (εικόνα 4.4).



Εικόνα 4.4
`istlComboBox`

Δημιουργείται ένα `Vector` με ονομασία `mpt` το οποίο θα περιέχει αντικείμενα τύπου `istlComboModelItem`. Κάθε τέτοιο αντικείμενο θα αποτελέσει και μία επιλογή για το `combobox`. Στην συγκεκριμένη περίπτωση η κάθε επιλογή περιέχει μία εικόνα και έναν τίτλο. Κατόπιν υλοποιείται ένα αντικείμενο τύπου `istlComboBox` με ονομασία `maptypes` και με όρισμα αυτό το `Vector` (`new istlComboBox(mpt);`), ενώ δίνεται μια εικόνα που θα φαίνεται σαν προεπιλογή με την μέθοδο (`.setDefaultIcon(st.getMapTypeIco());`),

Κατόπιν μπορούμε με την χρήση ενός `actionlistener` σε ένα κουμπί να πάρουμε την επιλεγμένη τιμή του `maptypes istlcombobox` που δημιουργήσαμε, όπως φαίνεται στο παρακάτω κομμάτι κώδικα.

```

apply.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
matypes.getSelectedItem();
    }
});

```

Το συγκεκριμένο κομμάτι κώδικα προσθέτει ένα ActionListener στο κουμπί Apply της φόρμας παραμετροποίησης χάρτη. Κάθε φορά που πατιέται το κουμπί Apply το istlComboBox επιστρέφει την επιλεγμένη τιμή του με την μέθοδο .getSelectedItem();

IstlMenuItem

Το συγκεκριμένο αντικείμενο επεκτείνει το αντικείμενο Container και υλοποιείται με την χρήση ήδη υπάρχοντων αντικειμένων. Αρχιτεκτονικά είναι μια παραλλαγή του istlComboBox, έχοντας όμοιες ιδιότητες και χαρακτηριστικά.

Μια διαφορά με το istlComboBox βρίσκεται στην κλάση προδιαγραφής. Για να υλοποιηθεί ένα istlMenuItem χρησιμοποιείται μια κλάση προδιαγραφής, που περιέχει το αντικείμενο κάθε επιλογής. Η κλάση αυτή αναγνωρίζει αυτόματα αν η επιλογή θα είναι εικόνα, κείμενο ή και τα δύο. Παρέχοντας την δυνατότητα στο να υπάρχουν ταυτόχρονα επιλογές με εικόνα μόνο, κείμενο μόνο ή και τα δύο στο ίδιο menuItem, επιπρόσθετα για την υποστήριξη πολλαπλών επιπέδων μπορεί να δεχθεί και άλλο menuItem. Αυτό για να γίνει πρέπει να δημιουργηθεί το ένα istlMenuItem και να τοποθετηθεί κατόπιν ως επιλογή στο Vector που θα περιέχει τις επιλογές του κεντρικού menuItem. Τέτοιου τύπου είναι το αντικείμενο MenuItem που υπάρχει στο αντικείμενο istlMap και προσφέρει ανάλογες επιλογές (εικόνα 4.5). Παρακάτω θα εξετασθεί κομμάτι κώδικα για το πώς υλοποιήθηκε.



Εικόνα 4.5 istlMenuItem

Αυτό για να γίνει πρέπει να δημιουργηθεί το ένα istlMenuItem και να τοποθετηθεί κατόπιν ως επιλογή στο Vector που θα περιέχει τις επιλογές του κεντρικού menuItem. Τέτοιου τύπου είναι το αντικείμενο MenuItem που υπάρχει στο αντικείμενο istlMap και προσφέρει ανάλογες επιλογές (εικόνα 4.5). Παρακάτω θα εξετασθεί κομμάτι κώδικα για το πώς υλοποιήθηκε.

Χρήση

```

Vector mpt = new Vector();
istlMenuItem cmbi = new istlMenuItem("Hide");
mpt.addElement(cmbi);
cmbi = new istlMenuItem(st.getNotationAddIco(), "Add");
mpt.addElement(cmbi);
cmbi = new istlMenuItem(st.getNotationMoveIco(), "Move");
mpt.addElement(cmbi);
cmbi = new istlMenuItem(st.getNotationRemoveIco(), "Remove");
mpt.addElement(cmbi);
sub_menu = new istlMenuItem("Notations", mpt);
sub_menu.setDefaultIcon(st.getNotationPropertiesIco());

sub_menu.getMenuItemTitle(0).addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent evt) {
istlLabel current = (istlLabel) evt.getSource();
sub_menu.getMenuItems().setVisible(false);
menu.getMenuItems().setVisible(false);
t.getComponentForm().removeComponent(sub_menu.getMenuItems());
}
}

```

```

t.getComponentForm().removeComponent(menu.getMenuItems());
for(int k=0;k<t.getComponentForm().getComponentCount();k++)
{
t.getComponentForm().getComponentAt(k).repaint();
t.getComponentForm().getComponentAt(k).refreshTheme();
}
}
});

```

Η τεχνική υλοποίησης είναι η ίδια με του αντικειμένου `istlComboBox`. Δυστυχώς λόγω τεχνικού προβλήματος που παρέμεινε άλυτο μέχρι το τέλος, είναι απαραίτητο σε κάθε ακροατή γεγονός για την κάθε επιλογή να υπάρχει κομμάτι κώδικα που θα αναλαμβάνει να αποκρύπτει το συγκεκριμένο αντικείμενο (όπως φαίνεται παραπάνω).

```

Vector zmt = new Vector();
istlMenuItem cmb;
cmb = new istlMenuItem("Gps Config");
zmt.addElement(cmb);
cmb = new istlMenuItem(sub_menu);
cmb = new istlMenuItem("Exit");
zmt.addElement(cmb);
menu = new istlMenuItem("Options", zmt);

```

Κατόπιν όλο το μενού προστίθεται στο κεντρικό μενού με ονομασία `menu`.

IstlPopUp

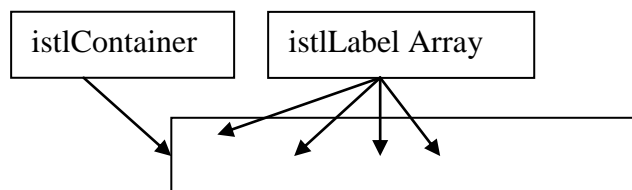
Αποτελεί απλουστευμένη μορφή του `menu item` και έχει βάθος μόνο ενός επιπέδου. Η υλοποίηση του είναι παρόμοια με το προηγούμενο αντικείμενο. Η εικόνα 4.6 δείχνει το `istlPopUp` που περιλαμβάνει επιλογές για τους εταίρους..



Εικόνα 4.6
`istlPopUp`

IstlRateItem

Το συγκεκριμένο αντικείμενο σε αντίθεση με τα προηγούμενα που προσπαθούν να εξομοιώσουν αντικείμενα από τον χώρο των υπολογιστών γραφείου αποτελεί ένα τελειώς καινούριο αντικείμενο και γι' αυτό έγινε προσπάθεια για μέγιστη ευελιξία και φορητότητα.

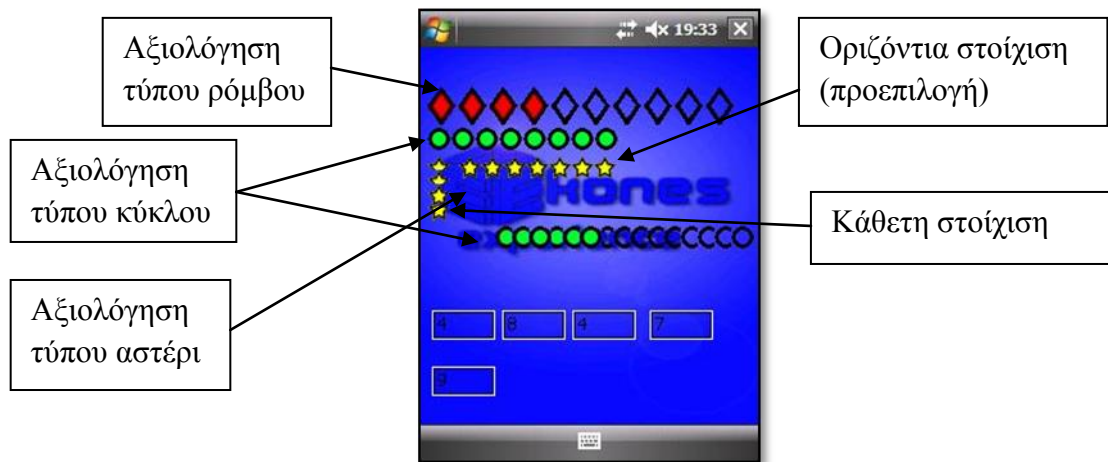


Εικόνα 4.7 Αρχιτεκτονική `istlRateItem`

Το σχεδιάγραμμα της εικόνας 4.7 δείχνει την βασική αρχιτεκτονική του αντικειμένου. Η λογική του είναι απλή, δημιουργείται ένας υποδοχέας τύπου `istlcontainer` με σύμβολα που αναπαριστούν βαθμολογία.

Υπάρχει κατασκευαστής που απαιτεί σαν όρισμα αριθμό αντικειμένων βαθμολογίας (πχ 5, 8 κλπ). Το προεπιλεγμένο σύμβολο αξιολόγησης είναι το αστέρι, ενώ παρέχει πληθώρα μεθόδων για σύμβολα χρήστη, αποστάσεις μεταξύ συμβόλων, οριζόντια ή κάθετη στοίχιση. Το αντικείμενο `istlRate` χρησιμοποιεί μεθόδους για αναγνώριση χαρακτηριστικών όπως μήκος οθόνης και προσαρμόζεται ανάλογα.

Η παρακάτω εικόνα (εικόνα 4.8) έχει τραβηχτεί από την υποστηρικτική εφαρμογή δοκιμών και ελέγχου που αναπτύχθηκε. Στην εικόνα αυτή είναι εμφανή όλα τα χαρακτηριστικά του αντικειμένου



Εικόνα 4.8 Βοηθητική εφαρμογή χρήσης istlRateItem

Στην προηγούμενη εικόνα είναι εμφανείς τα διαφορετικά σύμβολα αξιολόγησης χρήστη, η στοίχιση καθώς και οι αποστάσεις μεταξύ των συμβόλων.

Χρήση

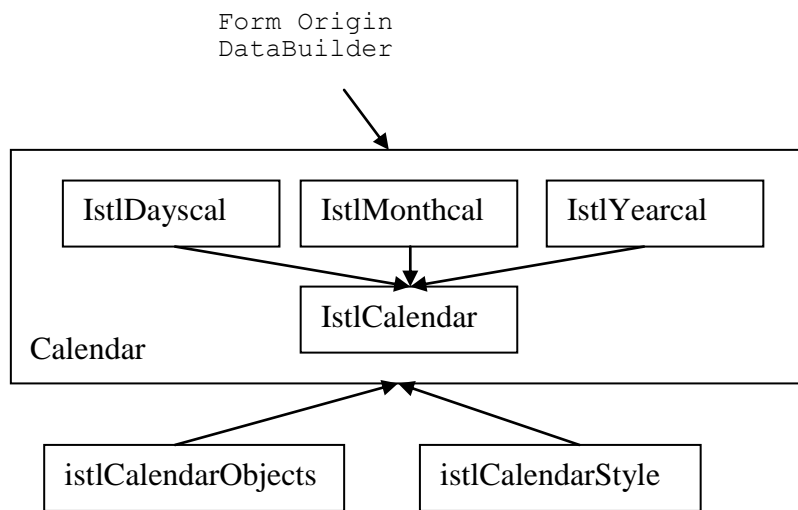
```
istlRate rt = new istlRate(5);
```

Με την χρήση του κατασκευαστή `istlRate(5)` υλοποιούμε ένα αντικείμενο `istlRate` 5 αστερών, το όρισμα αντιπροσωπεύει πόσα αστέρια θέλουμε να περιλαμβάνει το αντικείμενο βαθμολόγησης. Κατόπιν με την χρήση επιπρόσθετων μεθόδων μπορεί να μεταβληθεί το αντικείμενο `istlRate`.

Τύπος Επιστροφής Μεθόδου	Όνομα Μεθόδου	Χρήση
int	<code>getId()</code>	Επιστρέφει αναγνωριστική τιμή.
int	<code>getRate()</code>	Επιστρέφει την αξιολόγηση.
void	<code>setAlignmentVertical()</code>	Θέτει κάθετη στοίχιση
void	<code>setCompact(int compact)</code>	Θέτει επιθυμητές διαστάσεις χάρτη.
void	<code>setId(int id)</code>	Θέτει αναγνωριστικό κωδικό.
void	<code>setIstlItemSize(int AspectDimension)</code>	Θέτει το μέγεθος της εικόνας αξιολόγησης σε σχέση με το τρέχων.
void	<code>setIstlRateSize(com.sun.lwuit.geom.Dimension d)</code>	Θέτει το μέγεθος της εικόνας αξιολόγησης.
void	<code>setRate(int no)</code>	Θέτει αριθμό εικονιδίων αξιολόγησης.
void	<code>setRateEmptyIcon(com.sun.lwuit.Image empty)</code>	Θέτει εικόνα αξιολόγησης για άδεια κατάσταση.
void	<code>setRateFillIcon(com.sun.lwuit.Image fill)</code>	Θέτει εικόνα αξιολόγησης για επιλεγμένη κατάσταση.
void	<code>setRateStyle(com.sun.lwuit.plaf.Style ratestyle)</code>	Θέτει στυλ για το αντικείμενο <code>istlRate</code> .

Η αρχιτεκτονική του αντικειμένου Calendar

Το αντικείμενο calendar δημιουργήθηκε με σκοπό να προσφέρει χρονική αναπαράσταση του πακέτου στον χρήστη, καθώς και επιπρόσθετες λειτουργίες.



Εικόνα 4.9 Αρχιτεκτονική αντικειμένου Calendar

Το αντικείμενο calendar (εικόνα 4.10) αποτελείται από 3 υπό-αντικείμενα (istlDayscal, istlMonthcal, istlYearcal) και ένα κεντρικό αντικείμενο υποδοχέα (istlCalendar που επεκτείνει τον υποδοχέα form) (εικόνα 4.9). Το αντικείμενο istlCalendar έχει ένα κατασκευαστή με δύο ορίσματα. Το πρώτο όρισμα είναι τύπου form και αυτό γιατί αφού το istlCalendar είναι τύπου form, για να μπορέσει να λειτουργήσει και να συνδεθεί με την υπόλοιπη εφαρμογή χρειάζεται ένα όρισμα τέτοιου τύπου, που να δηλώνει την φόρμα προέλευσης, ενώ το δεύτερο τύπου DataBuilder περιέχει τα δεδομένα που έχουν ανακτηθεί από τον server του e-kones με πληροφορίες για το πακέτο.

Τα υπό-αντικείμενα istlDayscal, istlMonthcal, istlYearcal όπως υποδηλώνει και το όνομα τους έχουν συγκεκριμένη λειτουργία. Έτσι το αντικείμενο istlYearcal (εικόνα 4.11) επεκτείνει το αντικείμενο Container του LWUIT και η λειτουργία του είναι να σχεδιάζει τον τρέχων χρόνο και τα δεξιά και αριστερά κουμπιά ολίσθησης. Αντίστοιχη δομή έχει και το αντικείμενο istlMonthcal (εικόνα 4.12) μόνο που η δεξιά και αριστερή ολίσθηση γίνεται σε επίπεδο μήνα. Πιο σύνθετο από τα τρία είναι το αντικείμενο istlDayscal (εικόνα 4.13). Το συγκεκριμένο αντικείμενο επεκτείνει το αντικείμενο



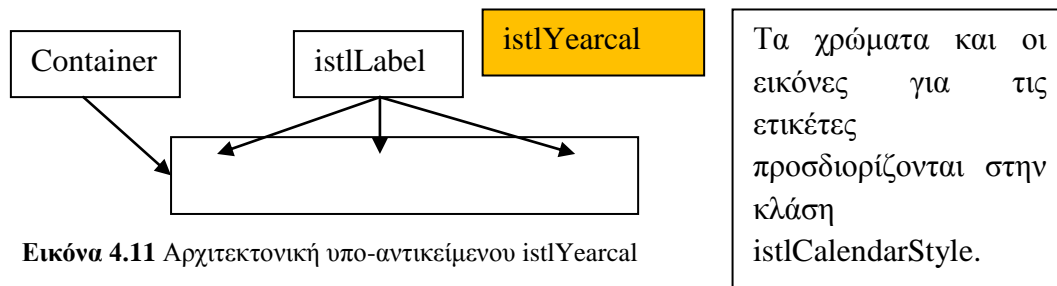
Εικόνα 4.10 Calendar

Container. Όπως και τα προηγούμενα έχει ένα κατασκευαστή με τρία ορίσματα (τρέχων μήνας, τρέχων χρόνος, Databuilder) αλλά ταυτόχρονα αποτελείται και υλοποιεί δύο ξεχωριστούς istlContainer. Ο ένας container με ονομασία cont_day_header αναλαμβάνει να σχεδιάσει την κεφαλίδα με τις ονομασίες των ημερών, ενώ ο δεύτερος με ονομασία cont_dayscont αναλαμβάνει να σχεδιάσει τις

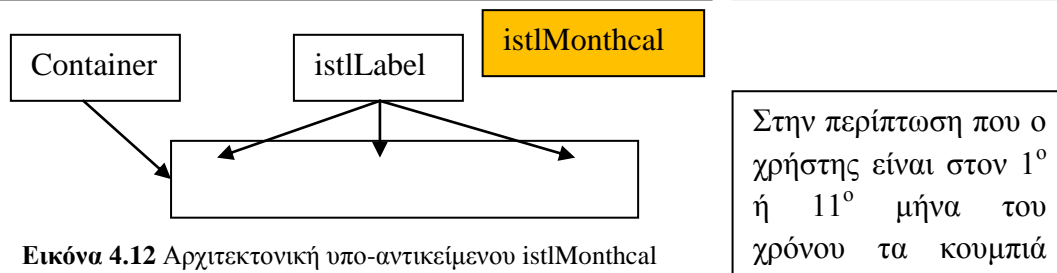
ημέρες ανάλογα με την αντιστοιχία μήνα και ημέρα εβδομάδας, επίσης προσθέτει ακροατή γεγονότος (ActionListener) σε κάθε ημέρα. Τέλος μαρκάρει με διαφορετικό χρώμα τις ημέρες του τρέχοντος πακέτου.

Από τα παραπάνω γίνεται ορατό ότι το αντικείμενο istlCalendar για τις ανάγκες της εφαρμογής είναι hardcoded. Παρόλα αυτά έχει υλοποιηθεί και μια πιο ελαφριά γενική έκδοση η οποία όμως δεν προσφέρει δυνατότητα προσθήκης ακροατή γεγονότων για την κάθε ημέρα.

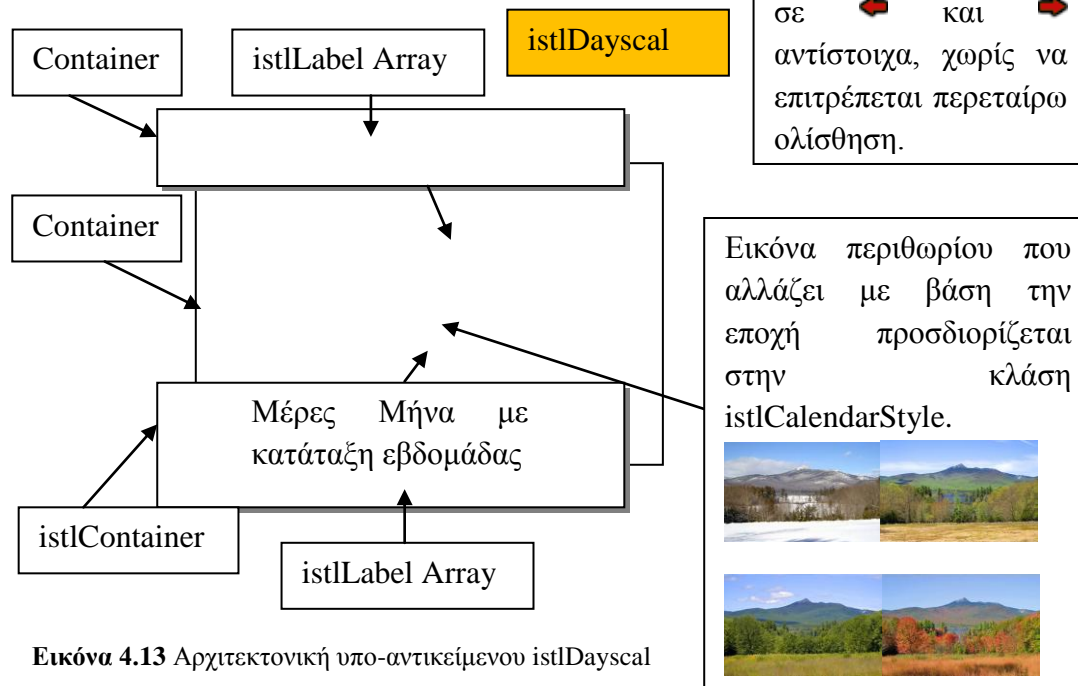
Το istlCalendarObjects διατηρεί πληροφορίες για τις σημειώσεις του χρήστη καθώς επίσης εκτελεί κάποιες βασικές λειτουργίες σχετικά με αυτές. Το istlCalendarStyle είναι κλάση που σχετίζεται με την γραφική παρουσίαση συστατικών των υπολοίπων όπως οι εικόνες.



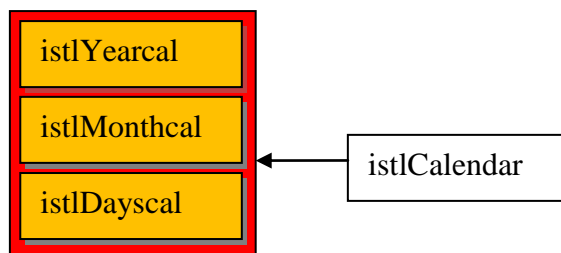
Εικόνα 4.11 Αρχιτεκτονική υπο-αντικείμενου istlYearcal



Εικόνα 4.12 Αρχιτεκτονική υπο-αντικείμενου istlMonthcal



Εικόνα 4.13 Αρχιτεκτονική υπο-αντικείμενου istlDayscal

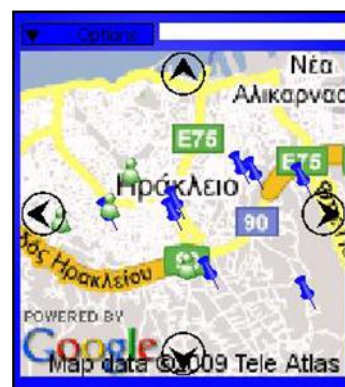


Εικόνα 4.14 Η συνολική αρχιτεκτονική του αντικειμένου Calendar από τα τρία υποαντικείμενα

Η αρχιτεκτονική του αντικειμένου istlMap

Το αντικείμενο istlMap προσφέρει χωρική αναπαράσταση του πακέτου στον χρήστη (εικόνα 4.15). Με τον όρο χωρική αναπαράσταση εννοούνται πληροφορίες όπως οι εταίροι του πακέτου, άλλοι χρήστες του πακέτου, καθώς και προσωπικές πληροφορίες. Για να έχουν όλα αυτά την χωρική έννοια εμφανίζονται σε χάρτη που προέρχεται από την υπηρεσία Static Google Maps. Επίσης εμφανίζονται και οι γεωγραφικές συντεταγμένες πλάτους και μήκους που βρίσκεται ο χρήστης, αλλά δεν αναπαρίστανται γραφικά.

Το αντικείμενο istlMap επεκτείνει το αντικείμενο form, και έχει δύο κατασκευαστές. Ο ένας με ορίσματα ένα αντικείμενο τύπου form το οποίο δηλώνει την φόρμα προέλευσης προέλευσης και ένα αντικείμενο τύπου Vector που περιέχει τα δεδομένα των εταίρων. Ο δεύτερος κατασκευαστής έχει επιπρόσθετα άλλο ένα αντικείμενο τύπου UserActivityData που περιέχει πληροφορίες χρήστη. Ο λόγος που υπάρχουν δύο διαφορετικοί κατασκευαστές είναι ένας περιορισμός της εικονικής μηχανής J9 ο οποίος θα αναλυθεί στο τέλος.

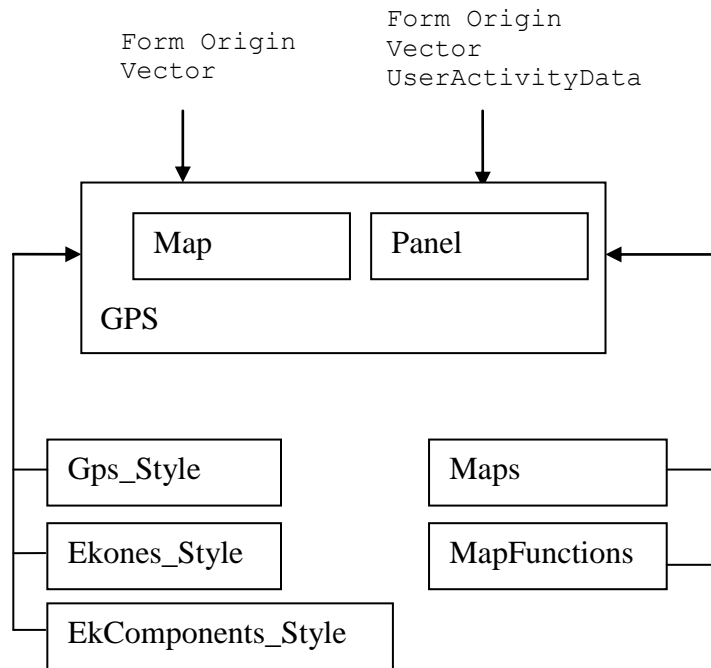


Εικόνα 4.15 istlMap

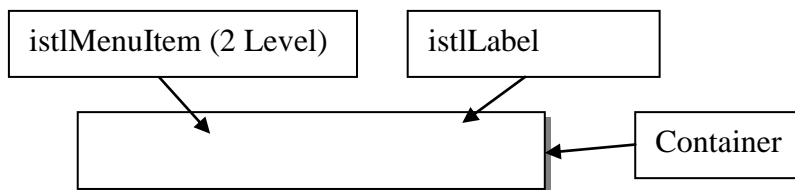
Το αντικείμενο αποτελείται από δύο υπό-αντικείμενα τύπου istlContainers (εικόνα 4.16), το ένα με ονομασία panel, βρίσκεται στο πάνω μέρος και περιλαμβάνει ένα istlMenuItem δύο επιπέδων καθώς και ένα istlLabel στο οποίο εμφανίζονται οι συντεταγμένες θέσης του χρήστη (εικόνα 4.17). Το άλλο με ονομασία map, αποτελεί τον υποδοχέα πάνω στο οποίο τοποθετείται ο χάρτης και όλα τα αντικείμενα σήμανσης (user markers, ekones users markers και e-kones) (εικόνα 4.18), τα οποία θα αναλυθούν στην συνέχεια.

Συνολικά το αντικείμενο istlMap αποτελεί ένα σύνθετο αντικείμενο λόγω των χαρακτηριστικών αλληλεπίδρασης που προσφέρει, όπως εισαγωγή, μετακίνηση και διαγραφή σήμανσης χρήστη, προβολή πληροφοριών για τους επιχειρηματικούς εταίρους του πακέτου καθώς και παρακολούθηση της κίνησης των άλλων χρηστών που είναι εγγεγραμμένη στο ίδιο πακέτο σε πραγματικό χρόνο πάνω στον χάρτη. Εκτός όμως από αυτές τις λειτουργίες, την πολυπλοκότητα του αντικειμένου αυξάνουν και οι αλγόριθμοι που εκτελούνται για να υποστηρίξουν αυτές τις

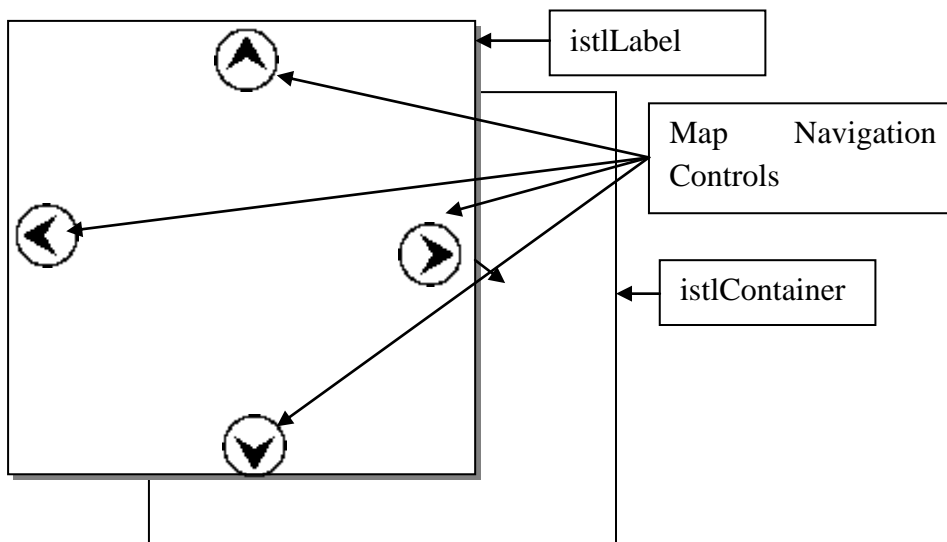
λειτουργίες όπως αλγόριθμος μαθηματικών συναρτήσεων χάρτη, 2 νήματα για την περιοδική ενημέρωση των θέσεων των άλλων χρηστών και την θέση του χρήστη.



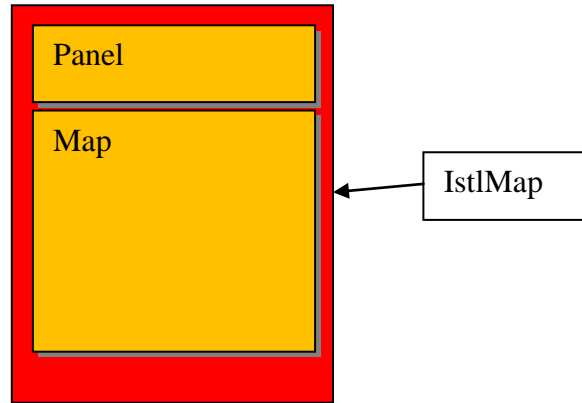
Εικόνα 4.16 Αρχιτεκτονική του αντικειμένου `istlMap`



Εικόνα 4.17 Αρχιτεκτονική υπο-αντικειμένου `panel`.



Εικόνα 4.18 Υποαντικείμενο `map`



Εικόνα 4.19 Γενική άποψη της αρχιτεκτονικής αντικειμένου istlMap

Αλληλεπίδραση με τον υποδοχέα MAP του αντικειμένου istlMap

Το αντικείμενο map είναι τύπου istlContainer, όπως αναφέρθηκε προηγούμενος το αντικείμενο istlContainer μπορεί να ανιχνεύσει γεγονότα «πέννας» και με κατάλληλες μεθόδους να επιστρέψει την συντεταγμένη στην οθόνη που παράχθηκε το γεγονός. Πάνω από το αντικείμενο map υπάρχει αντικείμενο τύπου istlLabel με ονομασία istlmapholder.

Το μόνο γεγονός που αντιλαμβάνεται και αξιοποιεί ο υποδοχέας map είναι πίεση και σύρσιμο της πέννας πάνω στην περιοχή του. Αντίστοιχα το γεγονός πίεσης δημιουργεί στην θέση που συνέβη αντικείμενο τύπου istlPopUp προσφέροντας αντίστοιχες επιλογές, ενώ το σύρσιμο προκαλεί την μετακίνηση ενός σημαδιού σε άλλη θέση πάνω στον χάρτη.

Σημάνσεις Πάνω Στον Χάρτη

Η εφαρμογή παρέχει 3 ειδών σημάνσεις για τον χάρτη (σημάνσεις χρήστη, σημάνσεις θέσεων άλλων χρηστών, σημάνσεις e-kones). Οι σημάνσεις αυτές δεν είναι στην πραγματικότητα δυναμικές, αλλά δημιουργούνται κατά την εκκίνηση του αντικειμένου GPS. Οι σημάνσεις τύπου e-kones που παρουσιάζουν τις θέσεις των επιχειρηματικών εταίρων του πακέτου δημιουργούνται με βάση το πλήθος τους όπως λαμβάνεται από το XML που έχει αναγνωσθεί από την βάση δεδομένων της υπηρεσίας e-kones. Κατόπιν εκχωρείται αντίστοιχος ακροατής γεγονότος που σχετίζεται με τον συγκεκριμένο εταίρο και τοποθετείται στην αντίστοιχη θέση στον χάρτη με βάση τις γεωγραφικές συντεταγμένες που είχε καταχωρήσει στην υπηρεσία του e-kones. Το πλήθος των σημάνσεων θέσης άλλων χρηστών που έχουν το ίδιο πακέτο με τον χρήστη υπολογίζεται και σε αυτή την περίπτωση από το αντίστοιχο XML που έχει ανακτηθεί από την βάση δεδομένων της υπηρεσίας e-kones με παρόμοιο τρόπο όπως και οι σημάνσεις εταίρων, επιπρόσθετα όμως κάθε 10 δευτερόλεπτα η εφαρμογή ενημερώνεται για την καινούρια θέση χρήστη. Το πλήθος των σημάνσεων χρήστη δεν μπορεί να προσδιοριστεί από την αρχή άλλα ούτε και να είναι άπειρο για λόγους απόδοσης, οπότε η εφαρμογή θεωρεί αντίστοιχο αριθμό σημάνσεων ίσο με το πλήθος των εταίρων. Στις σημάνσεις χρήστη εκχωρείται

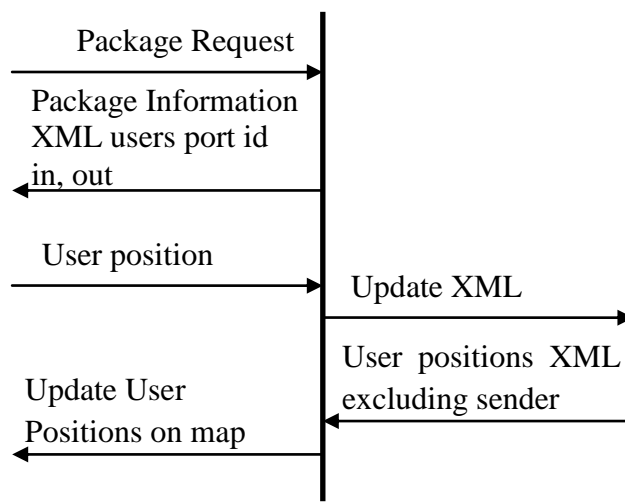
ακροατής γεγονότων που δημιουργεί αντικείμενο τύπου `istlPopUp` με αντίστοιχες επιλογές όταν ενεργοποιηθεί. Οι σημάνσεις χρήστη τοποθετούνται εκτός περιοχής οθόνης και αποκρύπτονται. Στην πραγματικότητα όταν ο χρήστης «δημιουργεί» μία προσωπική σήμανση, αυτή εμφανίζεται και τοποθετείται στο χάρτη σε θέση στην οθόνη όπου παρήχθηκε γεγονός πίεσης. Αντίστοιχα η «διαγραφή» αποτελεί απόκρυψη της σήμανσης και τοποθέτηση της εκτός οθόνης.

Κάθε τύπος σήμανσης είναι αντικείμενο τύπου `istlLabel` καθώς επίσης προσφέρει δικές του επιλογές και υπόκειται σε συγκεκριμένους περιορισμούς (πχ, δεν μπορεί να διαγραφεί σήμανση τύπου `e-kones`).

Νήματα

Το αντικείμενο `istlMap` ξεκινάει δύο νήματα κατά την εκκίνηση του. Το ένα νήμα είναι υπεύθυνο για την ενημέρωση της εφαρμογής με τις θέσεις των άλλων χρηστών που μοιράζονται το ίδιο πακέτο. Κάθε 10000 `millisecond` ενημερώνεται από τον διακοσμιτή της υπηρεσίας του `e-kones` για τις νέες θέσεις των χρηστών και τοποθετεί τις αντίστοιχες σημάνσεις στον χάρτη. Το δεύτερο νήμα κάθε 1000 `millisecond` ενημερώνεται από τον δορυφόρο για την γεωγραφική θέση του χρήστη την οποία αναρτά στο αντικείμενο `panel`.

Αξίζει να σημειωθεί ότι υπηρεσία για την ενημέρωση των χρηστών δεν έχει δημιουργηθεί έτσι οι θέσεις των χρηστών που εμφανίζονται στον χάρτη στην



Εικόνα 4.20 Πιθανή αρχιτεκτονική υπηρεσίας που θα υποστήριζε σε πραγματικό χρόνο ανταλλαγή μηνυμάτων θέσεις μεταξύ χρηστών.

δοκιμαστική έκδοση ανακτώνται από διακοσμιτή διαδικτύου είναι όμως σε αρχείο και δεν δημιουργούνται με δυναμικό τρόπο. Η δημιουργία μίας τέτοιας υπηρεσίας στα πλαίσια αυτής της πτυχιακής εργασίας δεν κρίθηκε απαραίτητη. Παρόλα αυτά μια πιθανή αρχιτεκτονική που θα μπορούσε να δημιουργηθεί παρουσιάζετε στο παρακάτω σχεδιάγραμμα (εικόνα 4.20).

Σε αυτό το θεωρητικό μοντέλο ανταλλαγής δεδομένων μεταξύ διακοσμιτή και χρήστη απαιτείται η ύπαρξη ενός τρίτου νήματος που στέλνει την θέση του χρήστη. Ο

διακοσμιτής μετά την επιλογή εκδρομικού πακέτου από τον χρήστη μαζί με τις πληροφορίες του επιλεγμένου πακέτου στέλνει δύο θέσεις «πόρτες» στις οποίες η εφαρμογή θα στέλνει και θα λαμβάνει τις πληροφορίες θέσης. Κατά την εκκίνηση του αντικειμένου istlMap τα νήματα ενημέρωσης θέσης χρήστη και άλλων χρηστών εκτελούνται. Είναι λογικό να απαιτούνται χρόνοι εκτέλεσης διαφορετικοί για αποστολή και λήψη καθώς και πρωτόκολλο επικοινωνίας για μεταφορά μηνυμάτων τα οποία θα μπορούν να ρυθμίζουν τους χρόνους εκτέλεσης, σε περιπτώσεις όπως υπερχειλίσις της προσωρινής μνήμης του διακοσμιτή για αποφυγή απώλειας δεδομένων.

Από την πλευρά του ο διακοσμιτής μόλις λάβει δεδομένα θέσης σε συγκεκριμένους χρόνους θα ανανεώνει το αρχείο των θέσεων. Η ανανέωση μπορεί να γίνεται είτε με χρονικό περιθώριο είτε με συνθήκη (πχ όλοι οι χρήστες του πακέτου να έχουν στείλει ενημέρωση θέσης κατά x φορές). Η είσοδος του διακοσμιτή χρειάζεται να έχει προσωρινή μνήμη έτσι ώστε να προλαβαίνει να κάνει εργασίες εισόδου/εξόδου με μικρή πιθανότητα σφάλματος. Ορθή θεωρείται και η δημιουργία πρωτόκολλου μηνυμάτων που θα ενημερώνουν τους χρήστες έτσι ώστε να μειώνουν την ταχύτητα μετάδοσης σε περιπτώσεις συμφόρησης.

Ένα δείγμα XML αρχείου που χρησιμοποιείται από την εφαρμογή για λήψη των θέσεων άλλων χρηστών χωρίς να λαμβάνεται υπόψη πρωτόκολλο ρύθμισης είναι το παρακάτω:

```
<users>
  <user id='1'>
    <username>Kotsalomistos</username>
    <last_name>Kotsalis</last_name>
    <first_name>Dimitris</first_name>
  </user>
  <user id='2'>
    <username>Gallactica</username>
    <last_name>Milolidakis</last_name>
    <first_name>Giannis</first_name>
  </user>
  <user id='3'>
    <username>Larisaios</username>
    <last_name>Vellis</last_name>
    <first_name>Giorgos</first_name>
  </user>
  <user id='4'>
    <username>Episimonas</username>
    <last_name>Plemenos</last_name>
    <first_name>Argiris</first_name>
  </user>
</users>
```

Το συγκεκριμένο θα μπορούσε να διαμορφωθεί κατάλληλα έτσι ώστε να περιλαμβάνει και μηνύματα εφαρμογής

```
<exchange>
<code>
```

```

    <transmission_rate>1</transmission_rate>
</code>
<users>
  <user id='1'>
    <username>Kotsalomistos</username>
    <last_name>Kotsalis</last_name>
    <first_name>Dimitris</first_name>
  </user>
  <user id='2'>
    <username>Gallactica</username>
    <last_name>Milolidakis</last_name>
    <first_name>Giannis</first_name>
  </user>
  <user id='3'>
    <username>Larisaios</username>
    <last_name>Vellis</last_name>
    <first_name>Giorgos</first_name>
  </user>
  <user id='4'>
    <username>Episimonas</username>
    <last_name>Plemenos</last_name>
    <first_name>Argiris</first_name>
  </user>
</users>
</exchange>

```

Το παραπάνω αρχείο XML έχει διαμορφωθεί κατάλληλα έτσι ώστε να υποστηρίζει ανταλλαγή μηνυμάτων συστήματος. Ο κόμβος `</code>` μπορεί υποθετικά να ενθυλακώνει μηνύματα συστήματος, αντίστοιχα το γνώρισμα `<transmission_rate>` `</transmission_rate>` που βρίσκεται μέσα στον κόμβο `<code></code>` θα μπορούσε να περιέχει πληροφορίες που να σχετίζονται με την ταχύτητα αποστολής. Τιμή 1 όπως στο παράδειγμα να σημαίνει κανονική ροή ενώ μεγαλύτερες να σημαίνουν συντελεστή μείωσης πχ τιμή 5 μπορεί να σημαίνει 50% του προηγούμενου χρόνου αποστολής, ενώ τιμή 0 εκτός λειτουργίας. Η εφαρμογή με βάση αυτές τις τιμές θα στέλνει την θέση του χρήστη στον διακοσμητή.

Αλγόριθμοι Μαθηματικών Υπολογισμών Χάρτη

Ένα από τα δυσκολότερα κομμάτια αυτής της εργασίας ήταν η μετατροπή των γεωγραφικών συντεταγμένων σε συντεταγμένες οθόνης, σε συνδυασμό με τον χάρτη από την υπηρεσία Google Maps και με δυνατότητα μεταβολής επιπέδου μεγέθυνσης, ανάλυσης αλλά και τύπου χάρτη.

Η υπηρεσία google maps προσφέρει χάρτη με βάση το κέντρο γεωγραφικών συντεταγμένων με επιθυμητή μεγέθυνση, ανάλυση και τύπο. Δεδομένης της ανάλυσης της οθόνης και τις γεωγραφικές θέσεις, είτε των εταίρων που είναι γνωστές από την υπηρεσία του e-kones, είτε την θέση του χρήστη την οποία παρέχει η συσκευή GPS πρέπει να υπολογιστεί η αντιστοιχία γεωγραφικών συντεταγμένων σε θέση στην οθόνη. Ο χώρος που τοποθετείται ο χάρτης στην συσκευή είναι ανάλυσης 232X232.

Το σύστημα εκτελεί πράξη υπολογισμού με βάση τις γεωγραφικές συντεταγμένες των εταίρων για υπολογισμό κέντρου και αιτείται από την υπηρεσία

Google Maps χάρτη με αυτό το κέντρο. Με πειραματικό τρόπο βρέθηκε μαθηματική σχέση μεταξύ ανάλυσης και zoom.

Ανάλυση	Μεγέθυνση	Ολίσθηση Μισής Οθόνης	Ολίσθηση Ολόκληρης οθόνης
128X128	17 (μέγιστη)	0.000689	0.001378
	8	$0.000689X2^9$	$0.001378X2^9$
	4	$0.000689X2^{13}$	$0.001378X2^{13}$
136X136	17 (μέγιστη)	0.000738	0.001466
	12	$0.000738X2^4$	$0.001466X2^4$
	8	$0.000738X2^9$	$0.001466X2^9$
	4	$0.000738X2^{13}$	$0.001466X2^{13}$
144X144	17 (μέγιστη)	0.000769	0.001538
	16	$0.000769X2^1$	$0.001538X2^1$
	12	$0.000769X2^4$	$0.001538X2^4$
	8	$0.000769X2^9$	$0.001538X2^9$
152X152	17 (μέγιστη)	0.000819	0.001638
	16	$0.000819X2^1$	$0.001638X2^1$
	12	$0.000819X2^4$	$0.001638X2^4$
160X160	17 (μέγιστη)	0.0008595	0.001719
	16	$0.0008595X2^1$	$0.001719X2^1$
	12	$0.0008595X2^4$	$0.001719X2^4$
168X168	17 (μέγιστη)	0.000903	0.001806
	16	$0.000903X2^1$	$0.001806X2^1$
	12	$0.000903X2^4$	$0.001806X2^4$
176X176	17 (μέγιστη)	0.000942	0.001884
	16	$0.000942X2^1$	$0.001884X2^1$
	12	$0.000942X2^4$	$0.001844X2^4$
184X184	17 (μέγιστη)	0.000988	0.001976
	12	$0.000988X2^4$	$0.001976X2^4$
	8	$0.000988X2^9$	$0.001976X2^9$
	4	$0.000988X2^{13}$	$0.001976X2^{13}$
192X192	17 (μέγιστη)	0.001031	0.002062
	8	$0.001031X2^9$	$0.002062X2^9$
	4	$0.001031X2^{13}$	$0.002062X2^{13}$
232X232	17 (μέγιστη)	0.00124	0.00248
	12	$0.00124X2^4$	$0.00248X2^4$

Στον παραπάνω πίνακα εμφανίζονται τα πειραματικά αποτελέσματα συσχέτισης μεταξύ ανάλυσης και μεγέθυνσης. Η πρώτη στήλη του πίνακα δηλώνει την ανάλυση του χάρτη, η δεύτερη το επίπεδο μεγέθυνσης, η τρίτη στήλη την ολίσθηση σε γεωγραφικές συντεταγμένες (μοίρες – degrees) που απαιτείται για τον επόμενο χάρτη με κέντρο την άκρη του τρέχοντος και η τέταρτη στήλη δηλώνει την ολίσθηση που απαιτείται για να πάρουμε χάρτη που βρίσκεται ακριβώς δίπλα από τον τρέχων. Όπως είναι προφανές η τιμή στην τέταρτη στήλη είναι η διπλάσια από την αντίστοιχη της τρίτης στήλης. Τα παραπάνω αποτελούν πειραματικά αποτελέσματα και ενδέχεται πιθανότητα σφάλματος.

Είναι άξιο παρατήρησης ότι για συγκεκριμένη ανάλυση υπάρχει μαθηματική σχέση μεταξύ συγκεκριμένου επιπέδου μεγέθυνσης και ανάλυσης. Όπως είναι όμως εμφανές δεν υπάρχει μαθηματική σχέση μεταξύ διαφορετικών αναλύσεων. Με αποτέλεσμα να υπάρχει πρόβλημα με τον υπολογισμό νέας θέσης στο χάρτη σε

περίπτωση που ο χρήστης μεταβάλλει την ανάλυση του χάρτη προς κατέβασμα από την υπηρεσία Google Maps. Άρα μπορεί να υπάρξει αλγόριθμος υπολογισμού θέσης στο χάρτη με βάση την ανάλυση που θα επιτρέπει και ολίσθηση κρατώντας τις θέσεις των σημάνσεων στον χάρτη.

```
double pxf;
    double long_dif,lat_dif;
    int zoom_dif = Math.abs(16-zoom);
    int zoommf=2;
    for(int x=0;x<zoom_dif;x++)
    {
        zoommf=zoommf*2;
    }
    cfs = 0.00248*zoommf;
    cfc = 0.00124*zoommf;
    pxf=cfs/232;
    long_dif=cen_long+cfc;
    lat_dif=cen_lat+cfc;
    mark_long = (int) ((long_dif-longt)/ pxf);
    mark_lat = (int) ((lat_dif-lat) / pxf);
```

Εκτός από τον περιορισμό μεταξύ μεταβλητής ανάλυσης και επίπεδο μεγέθυνσης εμφανίστηκαν προβλήματα μεταξύ τύπων χάρτη, αναλύσεων και επίπεδο μεγέθυνσης. Τα προβλήματα αυτά πιθανότατα οφειλόταν σε ασυμβατότητα μεταξύ LWUIT και J9, για αυτό το λόγω ο αλγόριθμος «κλειδώθηκε» για ανάλυση οθόνης 244X244, αίτηση εικόνας χάρτη ανάλυσης 144X144, μεγέθυνσης 12 και προφίλ τύπου mobile, που αποτελούσαν ασφαλείς ρυθμίσεις χωρίς προβλήματα. Πάνω από τον χάρτη υπάρχουν τέσσερα βέλη (ένα σε κάθε άκρη), τα οποία επιτρέπουν ολίσθηση προς την ανάλογη κατεύθυνση με την χρήση παρόμοιου αλγορίθμου. Ωστόσο υπάρχει μικρό πρόβλημα το οποίο δεν έχει λυθεί, με αποτέλεσμα οι σημάνσεις να τοποθετούνται με μικρό σφάλμα κατά την ολίσθηση.

```
    public double MapRightScroll(double clat)
    {
        int zoom_dif = Math.abs(17-zoom);
        int zoommf=1;
        for(int x=0;x<zoom_dif;x++)
        {
            zoommf=zoommf*2;
        }
        cfc = 0.00124*zoommf;
        cen_lat= clat+(cfc/2);
        return cen_lat;
    }

    public double MapLeftScroll(double clat)
    {
        int zoom_dif = Math.abs(17-zoom);
        int zoommf=1;
        for(int x=0;x<zoom_dif;x++)
        {
            zoommf=zoommf*2;
        }
        cfc = 0.00124*zoommf;
        cen_lat=clat-(cfc/2);
        return cen_lat;
```

```

    }
    public double MapUpScroll(double clong)
    {
        int zoom_dif = Math.abs(17-zoom);
        int zoommf=1;
        for(int x=0;x<zoom_dif;x++)
        {
            zoommf=zoommf*2;
        }
        cfc = 0.00124*zoommf;
        cen_long=clong+(cfc/2);
        return cen_long;
    }

    public double MapDownScroll(double clong)
    {
        int zoom_dif = Math.abs(17-zoom);
        int zoommf=1;
        for(int x=0;x<zoom_dif;x++)
        {
            zoommf=zoommf*2;
        }
        cfc = 0.00124*zoommf;
        cen_long=clong-(cfc/2);
        return cen_long;
    }
}

```

Ένας ακόμα μαθηματικός αλγόριθμος, είναι ο αλγόριθμος υπολογισμού απόστασης μεταξύ των εταίρων. Ο αλγόριθμος βασίζεται στον τύπο του Haversine

```

R = earth's radius (mean radius = 6,371km)
Δlat = lat2- lat1
Δlong = long2- long1
a = sin2(Δlat/2) + cos(lat1).cos(lat2).sin2(Δlong/2)
c = 2.atan2(√a, √(1-a))
d = R.c

```

Δημιουργήθηκε μέθοδος που να μπορεί να εκτελέσει τις απαιτούμενες πράξεις του αλγορίθμου. Η κλάση Math του midp δεν παρείχε μαθηματικές μεθόδους κατάλληλες για την εκτέλεση κάποιων απαιτούμενων πράξεων οπότε δημιουργήθηκε κλάση istlMath προσφέροντας τις μαθηματικές συναρτήσεις atan, atan2, toRad και το Round.

```

public double MarkerDistanceCalculation(double slat,double
slong,double dlat,double dlong)
{
    double R = 6371; // km
    double dLat = istlMath.toRad(slat-dlat);
    double dLon = istlMath.toRad(slong-dlong);
    double a = Math.sin(dLat/2) * Math.sin(dLat/2)
+Math.cos(istlMath.toRad(dlat)) * Math.cos(istlMath.toRad(slat))
*Math.sin(dLon/2) * Math.sin(dLon/2);
    double c = 2 * istlMath.atan2(Math.sqrt(a), Math.sqrt(1-
a));
    double d = R * c;

    return istlMath.Round(d);
}

```

Αντικείμενο Επικοινωνίας με Υπηρεσία Google Maps

Για να υποστηριχτεί η υπηρεσία Google Maps δημιουργήθηκε απαραίτητη κλάση με ονομασία Maps. Η κλάση αυτή έχει ένα «κενό» κατασκευαστή.

```
public Maps()  
{  
  
}
```

Με την χρήση κατάλληλων μεθόδων γίνεται σύνδεση με την υπηρεσία Static Google Maps και ανάκτηση χάρτη. Η κλάση περιέχει μέθοδο που συνδέεται με την υπηρεσία μέσω πρωτοκόλλου http. Για να γίνει επιτυχής σύνδεση και ανάκτηση χάρτη, χρειάζεται να είναι γνωστά ο σύνδεσμος καθώς και άλλοι παράμετροι, όπως επιθυμητή ανάλυση χάρτη, επίπεδο μεγέθυνσης, τύπος χάρτη. Τις λειτουργίες αυτές αναλαμβάνουν κατάλληλες μέθοδοι. Παράδειγμα σύνδεσης με την υπηρεσία Google Maps φαίνεται παρακάτω.

```
mp = new Maps();  
mp.setMapZoom(cfg.getZoom());  
mp.setMapXdimension(cfg.getResolution());  
mp.setMapYdimension(cfg.getResolution());  
mp.setMapProfile(cfg.getMapType());  
mp.setMapLon(rd[0]);  
mp.setMapLat(rd[1]);  
mp.setMapURL();
```

Στο παραπάνω κομμάτι κώδικα δημιουργείται ένα στιγμιότυπο της κλάσης Maps με ονομασία mp. Στο συγκεκριμένο παράδειγμα εμφανίζεται η κλάση προδιαγραφής GpsConfig από την οποία χρησιμοποιούνται τιμές ως ορίσματα μεθόδων ρύθμισης χαρακτηριστικών του «προς κατέβασμα χάρτη» που επιστρέφονται με μεθόδους όπως φαίνεται παρακάτω.

```
cfg = new GpsConfig();  
cfg.setMapType("mobile");  
cfg.setResolution(144);  
cfg.setZoom(12);
```

Η κλάση προδιαγραφής GpsConfig διατηρεί προσαρμοσμένες ρυθμίσεις χρήστη για τον χάρτη. (Στην τελική έκδοση όμως οι προσαρμοσμένες ρυθμίσεις χρήστη έχουν απενεργοποιηθεί, με αποτέλεσμα ο χρήστης να μην μπορεί να κάνει προσαρμοσμένες ρυθμίσεις χάρτη).

Οι ρυθμίσεις του χάρτη έχουν γίνει. Το τελευταίο βήμα είναι η σύνδεση με την υπηρεσία και η ανάκτηση του επιθυμητού χάρτη.

```
Image image;  
try {  
    image = Image.createImage(mp.mapConnect(), 0,  
mp.mapConnect().length);  
} catch (Exception ex) {  
    istlmapholder.setText(ex.getMessage());  
}
```

Πλέον έχει γίνει σύνδεση και ανάκτηση εικόνας χάρτη η οποία αποθηκεύεται σε εικόνα με ονομασία image. Κατόπιν η εφαρμογή θα τοποθετήσει αυτή την εικόνα στον αντίστοιχο υποδοχέα.

Οι μέθοδοι της κλάσης Maps φαίνονται παρακάτω.

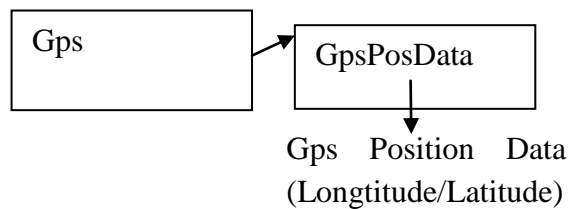
Τύπος Επιστροφής Μεθόδου	Όνομα Μεθόδου	Χρήση
com.sun.lwuit.Image	getMapImage()	Επιστρέφει τον χάρτη από την υπηρεσία google maps
java.lang.String	getMapUrl()	Επιστρέφει τον σύνδεσμο της υπηρεσία google maps
byte[]	mapConnect()	Συνδέεται με την υπηρεσία και ανακτά χάρτη.
void	setMapDimensions(com.sun.lwuit.geom.Rectangle d)	Θέτει επιθυμητές διαστάσεις χάρτη.
void	setMapFormat(java.lang.String format)	Θέτει τον τύπο του αρχείου (.png8, .png24 κλπ).
void	setMapKey(java.lang.String key)	Θέτει το κλειδί που απαιτείται από την υπηρεσία google maps
void	setMapLat(double lat)	Θέτει το επιθυμητό γεωγραφικό πλάτος για το κέντρο του επιθυμητού χάρτη.
void	setMapLon(double lon)	Θέτει το επιθυμητό γεωγραφικό μήκος για το κέντρο του επιθυμητού χάρτη.
Void	setMapProfile(java.lang.String profile)	Θέτει το επιθυμητό προφίλ χάρτη.
Void	setMapURL()	Προκαθορισμένη διεύθυνση υπηρεσίας.
Void	setMapURL(java.lang.String url)	Θέτει την διεύθυνση υπηρεσία google maps
Void	setMapXdimension(int x)	Θέτει την X επιθυμητή διάσταση εικόνας χάρτη.
Void	setMapYdimension(int y)	Θέτει την Y επιθυμητή διάσταση εικόνας χάρτη.
Void	setMapZoom(int zoom)	Θέτει το επιθυμητό επίπεδο μεγέθυνσης χάρτη.

Αντικείμενο GPS

Η εφαρμογή χρησιμοποιεί το πρωτόκολλο NMEA για την επικοινωνία με το ολοκληρωμένο κύκλωμα του GPS (SirfStarIII), η οποία γίνεται μέσω της κλάσης GPS που δημιουργήθηκε. Η εφαρμογή χρησιμοποιεί μία κλάση δεδομένων με ονομασία GpsPosData η οποία κρατάει τα δεδομένα και μία κλάση GPS που είναι υπεύθυνη για την σύνδεση με σύστημα GPS της συσκευής και την ανάκτηση των δεδομένων. Σε αυτό το σημείο πρέπει να σημειωθεί ότι μία πλήρης εφαρμογή GPS θα πρέπει να ελέγχει αν όχι όλα, τα περισσότερα «κλειδιά» μηνύματος που μεταδίδει το κύκλωμα GPS. Για λόγους απλότητας η εφαρμογή αυτή ελέγχει και προσπελάζει μόνο το κλειδί που σηματοδοτεί μήνυμα θέσης, και επομένως μόνο αυτό το μήνυμα. Παρόλα αυτά η συσκευή μεταδίδει περισσότερα μηνύματα.

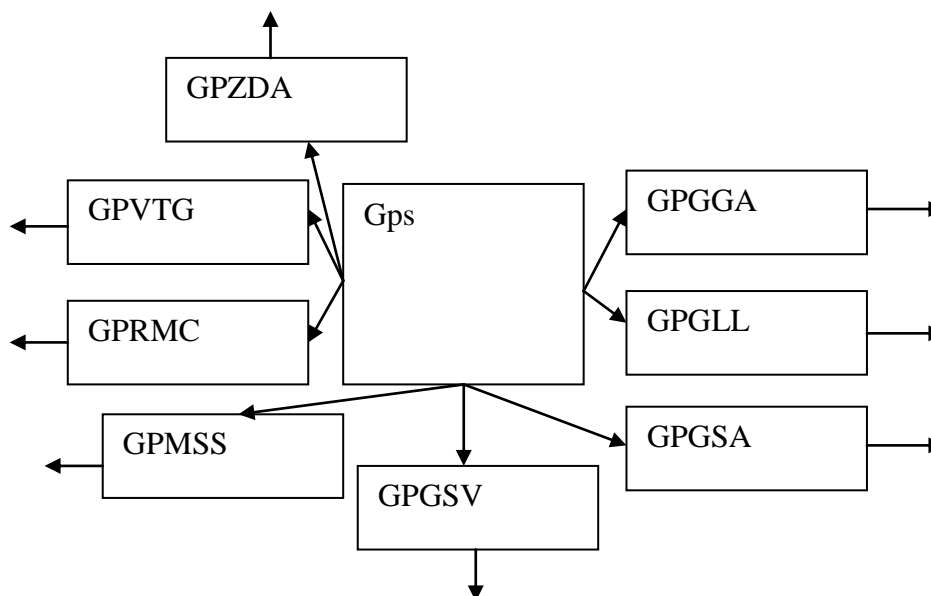
Αναγνωριστικό Κλειδί	Τύπος Μηνύματος
GGA	Time, position and fix type data.
GLL	Latitude, longitude, UTC time of position fix and status.
GSA	GPS receiver operating mode, satellites used in the position solution, and DOP values.
GSV	The number of GPS satellites in view satellite ID numbers, elevation, azimuth, and SNR values.
MSS	Signal-to-noise ratio, signal strength, frequency, and bit rate from a radio-beacon receiver.
RMC	Time, date, position, course and speed data.
VTG	Course and speed information relative to the ground.
ZDA	PPS timing message (synchronized to PPS).
150	OK to send message.

Η αρχιτεκτονική του αντικειμένου, η οποία είναι υπεύθυνη για την ανάγνωση μηνυμάτων GPS από την συσκευή φαίνεται στο παρακάτω σχεδιάγραμμα (εικόνα 4.21).



Εικόνα 4.21 Αρχιτεκτονική αντικειμένου που επικοινωνεί με το GPS

Μια εναλλακτική υλοποίηση που θα προσέφερε περισσότερη πληρότητα όσον αφορά την κάλυψη όλων των αναγνωριστικών κλειδιών θα μπορούσε να έχει την παρακάτω αρχιτεκτονική (εικόνα 4.22).



Εικόνα 4.22 Εναλλακτική αρχιτεκτονική για την επικοινωνία με το GPS που μπορεί να εκμεταλλευτεί όλους τους τύπους μηνυμάτων που δέχεται η συσκευή.

Η παραπάνω αρχιτεκτονική εκμεταλλεύεται πλήρως το κάθε αναγνωριστικό κλειδί με μία αντίστοιχη κλάση η οποία φέρει δεδομένα και μεθόδους για των

χειρισμό των μηνυμάτων με βάση το έγγραφο προδιαγραφής της SIRF για τα δεδομένα NMEA.

Για να λειτουργήσει το αντικείμενο GPS χρειάζεται πρώτα να υλοποιηθεί ένα στιγμιότυπο αυτού.

```
GPS gp = new GPS();
```

Μόλις δημιουργηθεί στιγμιότυπο κλάσης τύπου GPS αυτόματα συνδέεται με την θύρα com4, προσπαθώντας να προσπελάσει μηνύματα με αναγνωριστικό κλειδί \$GPGGA, τα οποία και περιέχουν πληροφορίες θέσης. Κατόπιν υλοποιεί ένα αντικείμενο τύπου GpsPosData και με κατάλληλες μεθόδους τοποθετεί τα δεδομένα. Η εφαρμογή μπορεί να ανακτήσει τα δεδομένα θέσης με τον παρακάτω τρόπο.

```
gp.RetreiveGpsData();
GPSPosData gpd =gp.getGPSPosData();
```

Πλέον η εφαρμογή μπορεί με κατάλληλες μεθόδους που παρέχει η κλάση GpsPosData να χειριστεί τα απαιτούμενα δεδομένα.

Τύπος Επιστροφής Μεθόδου	Όνομα Μεθόδου	Χρήση
java.lang.String	getGps()	Επιστρέφει μια πλήρη ανεπεξέργαστη γραμμή δεδομένων.
java.lang.String	getKey()	Επιστρέφει το αναγνωριστικό κλειδί του μηνύματος.
java.lang.String	getLat()	Επιστρέφει το γεωγραφικό Πλάτος.
java.lang.String	getLong()	Επιστρέφει το Γεωγραφικό Μήκος.
int	getNoOfSatellites()	Επιστρέφει αριθμό δορυφόρων που καλύπτουν την περιοχή.
double	getTime()	Επιστρέφει χρόνο.
boolean	isValid()	Εσωτερική μέθοδος ελέγχου για την ακεραιότητα των δεδομένων.
void	setGps(java.lang.String gps)	Θέτει μια πλήρη ανεπεξέργαστη γραμμή δεδομένων.
void	setKey(java.lang.String key)	Θέτει το αναγνωριστικό κλειδί του μηνύματος.
void	setLat(double Lat, java.lang.String Dir2)	Θέτει το γεωγραφικό Πλάτος και κατεύθυνση ορίζοντα.
void	setLong(double Long, java.lang.String Dir1)	Θέτει το γεωγραφικό Μήκος και κατεύθυνση ορίζοντα..
void	setNoOfSatellites(int n oOfSatellites)	Θέτει των αριθμό δορυφόρων που καλύπτουν την περιοχή.
void	setTime(double time)	Θέτει τον χρόνο
void	setValid(boolean valid)	Εσωτερική μέθοδος ελέγχου για την ακεραιότητα των δεδομένων.

Για να μπορέσει η εφαρμογή να έχει συνεχή ροή δεδομένων, έχει δημιουργηθεί ένα νήμα το οποίο εκτελείται κάθε 1000 millisecond επιστρέφοντας πληροφορίες θέσης.

Στην εφαρμογή που υλοποιήθηκε e-kones Experience δεν είναι ιδιαίτερα ευδιάκριτες οι δυνατότητες του συστήματος που υποστηρίζει λειτουργίες GPS. Έχει υλοποιηθεί μικρό πρόγραμμα για σκοπούς δοκιμών και αποσφαλμάτωσης στο οποίο φαίνονται με περισσότερη ακρίβεια αυτές οι λειτουργίες και θα παρουσιασθεί στο παράρτημα.

Διαχείριση δεδομένων Εφαρμογής

Η εφαρμογή λόγω της φύσης της (εφαρμογή χρήστη για την υπηρεσία –ekones) απαιτεί συνδεσιμότητα με διακοσμιτή μέσω του διαδικτύου, ενώ είναι αναγκαία και προσπέλαση του συστήματος αρχείων της συσκευής.

Προσπέλαση Του Συστήματος Αρχείων της Συσκευής

Η εφαρμογή απαιτεί την μεταφορά δεδομένων από φόρμα σε φόρμα, έτσι ώστε να μπορούν να υπάρξουν δεδομένα αλληλεπίδρασης χρήστη καθώς και δυνατότητα αποθήκευσης αυτών με τον τερματισμό της εφαρμογής. Όπως αναφέρθηκε ήδη η υλοποίηση της IBM (J9) δεν περιλαμβάνει κανένα προαιρετικό πακέτο. Για να μπορέσει όμως μία εφαρμογή να έχει πρόσβαση στο σύστημα αρχείων της συσκευής είναι απαραίτητη η υποστήριξη του JSR-75 (File Connection – PDA Optional Package).

Για εφαρμογές που απαιτούν αποθήκευση δεδομένων χωρίς υποστήριξη του JSR-75 το προφίλ Midp 2.0 προσφέρει το λεγόμενο “Record Store” (RMS – Record Management System). Το RMS είναι κατά κάποιο τρόπο μία εσωτερική βάση δεδομένων που κρατάει στο ίδιο το προφίλ πληροφορίες για την αποθήκευση των δεδομένων. Δημιουργήθηκε και χρησιμοποιείτε κυρίως για την αποθήκευση απλών δεδομένων όπως τα σκορ σε ένα παιχνίδι. Ο συγκεκριμένος τρόπος αποθήκευσης θα μπορούσε να προσφέρει αποθήκευση των επιλογών του χρήστη ως ένα βαθμό, δεν θα μπορούσε όμως να εξυπηρετήσει ανάγκες τύπου «προσωρινής μνήμης» που απαιτεί η συγκεκριμένη εφαρμογή. Για τον λόγο αυτόν κρίθηκε ακατάλληλη η χρήση του.

Η παραπάνω έλλειψη οδήγησε στην δημιουργία «βαρέων» κατασκευαστών σε κλάσης, ώστε να μπορούν να μεταφέρουν δεδομένα από τη μία φόρμα στην άλλη και να προσφέρουν τουλάχιστον την απαιτούμενη διατήρηση των δεδομένων κατά την εκτέλεση της εφαρμογής, με φυσικό αντίκτυπο στην απόδοση. Ακόμα όμως και αυτό το τέχνασμα με το μειονέκτημα που επέφερε δεν κατάφερε να λύσει πλήρως το πρόβλημα της διατήρησης των δεδομένων, έτσι τα δεδομένα διατηρούνται μεταξύ τμημάτων της εφαρμογής για να μην επιβαρυνεται περισσότερο σε απόδοση η ήδη επιβαρυνόμενη εφαρμογή.

Προσπέλαση Δεδομένων από την Υπηρεσία του E-kones

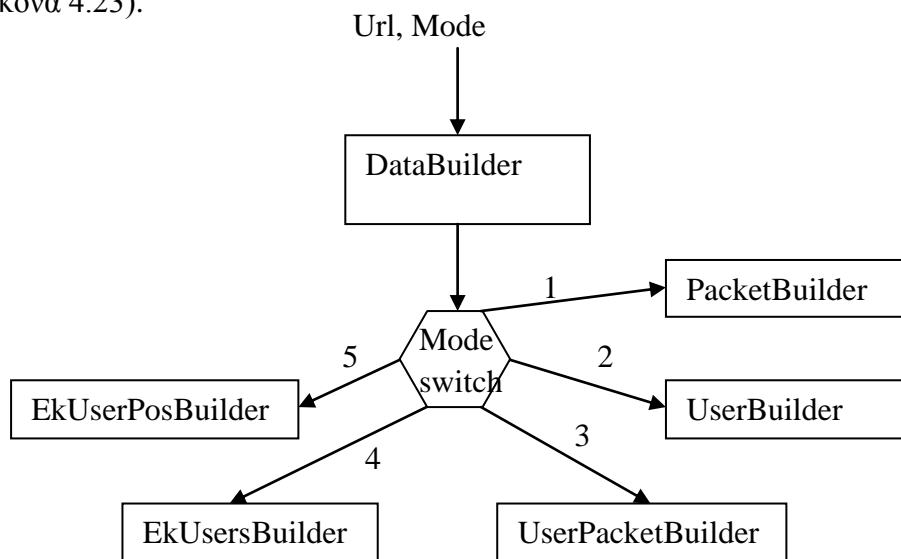
Στο κομμάτι της επικοινωνίας με διακοσμιτή δικτύου το προφίλ Midp προσέφερε τα απαραίτητα πρωτόκολλα (http). Η ανταλλαγή αρχείων από τον διακοσμιτή με την εφαρμογή έγινε με την χρήση αρχείου τύπου XML (όπως αναφέρθηκε προηγουμένως). Για την προσπέλαση των δεδομένων δημιουργήθηκε

ειδικό πρόγραμμα που θα διαβάζει αυτά τα δεδομένα (το οποίο θα παρουσιασθεί στο παράρτημα).

Η εφαρμογή χρησιμοποιεί τις παρακάτω κατηγορίες δεδομένων.

- Πακέτα που είναι εγγεγραμμένος ο χρήστης.
- Δεδομένα συγκεκριμένου (επιλεγμένου) πακέτου.
- Δεδομένα Θέσεων και Πληροφορίες Για άλλους Χρήστες.
- Προσωπικές Πληροφορίες Χρήστη.
- Δεδομένα Ρυθμίσεων (δεν υλοποιήθηκαν πλήρως, βλέπε προηγούμενο).

Η αρχιτεκτονική για τα δεδομένα παρουσιάζεται στην παρακάτω εικόνα (εικόνα 4.23).



Εικόνα 4.23 Αρχιτεκτονική αντικείμενου δημιουργίας δεδομένων για την

Από το παραπάνω σχεδιάγραμμα γίνεται προφανές ότι υπάρχει μία κλάση με έναν κατασκευαστή `DataBuilder` ο οποίος απαιτεί ένα όρισμα που δηλώνει την διεύθυνση του διακοσμιτή στην οποία που βρίσκεται το αρχείο, καθώς και ένα δεύτερο όρισμα που δηλώνει τον τρόπο λειτουργίας (`mode`). Ανάλογα με τον επιλεγμένο τρόπο λειτουργίας ακολουθεί η εκτέλεση αντίστοιχων μεθόδων.

Ο κατασκευαστής δημιουργίας δεδομένων.

```
public DataBuilder(String url,int mode)
{
    this.url=url;
    if(mode==1)
    {
        PacketBuilder();
    }
    else if(mode==2)
    {
        UserBuilder();
    }
    else if(mode==3)
    {
        UsersPacketBuilder();
    }
}
```



```

else if(mode==4)
{
    EkUsersBuilder();
}
else if(mode==5)
{
    EkUserPosBuilder();
}
}

```

Όπως φαίνεται παραπάνω, ο κατασκευαστής δεν παρέχει καμία ασφάλεια για επιλογή λειτουργίας διαφορετική από τις προκαθορισμένες, αλλά ούτε και για περιπτώσεις που ο χρήστης τοποθετήσει λάθος διεύθυνση διαδικτύου..

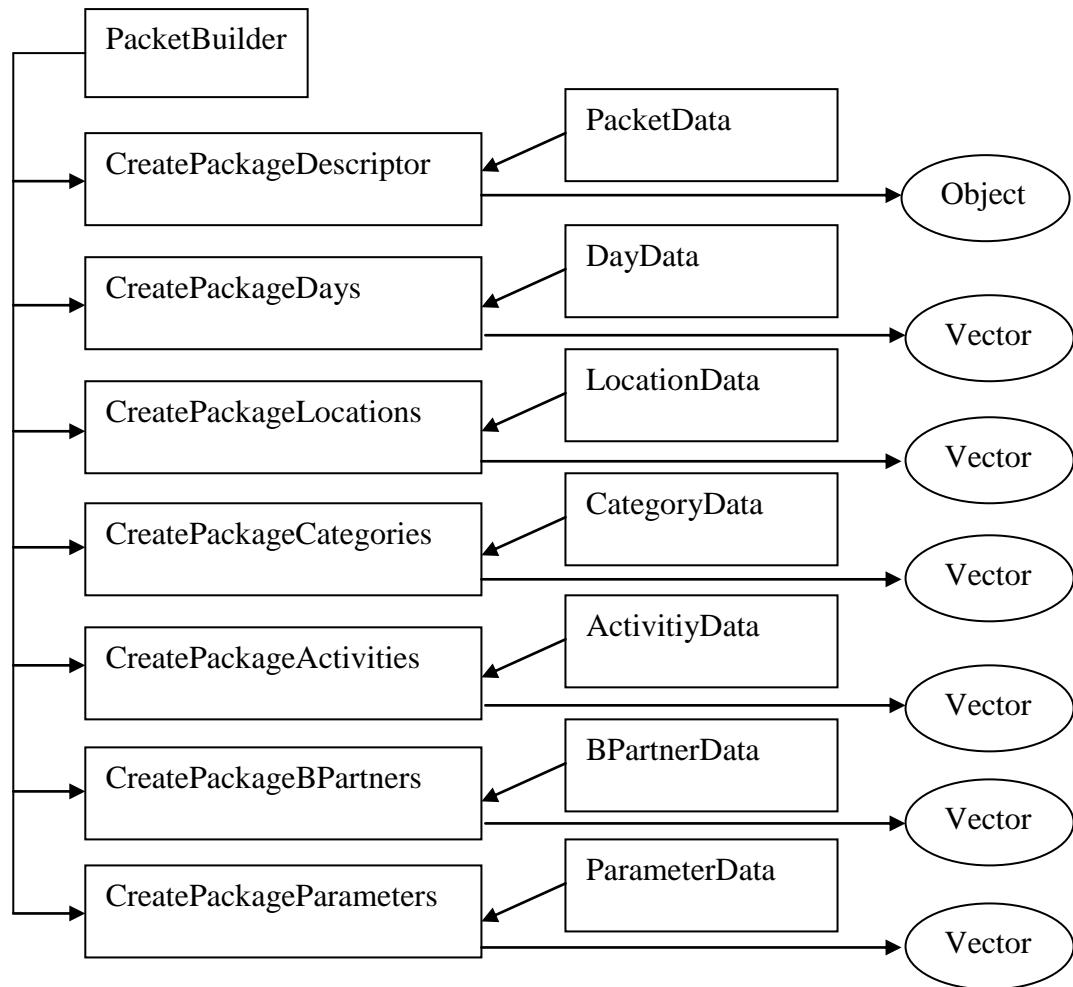
Επιλογή 1, αυτός ο τρόπος λειτουργίας ενεργοποιεί μεθόδους για το κτίσιμο των κλάσεων με πληροφορίες για το επιλεγμένο πακέτο. Ο κώδικας εκτέλεσης είναι ο ακόλουθος.

```

public void PacketBuilder()
{
    Days = new Vector();
    xmcon = new XMLConnect(url);
    CreatePackageDescriptor();
    CreatePackageDays();
    CreatePackageLocations();
    CreatePackageCategories();
    CreatePackageActivities();
    CreatePackageBPartners();
    CreatePackageParameters();
}

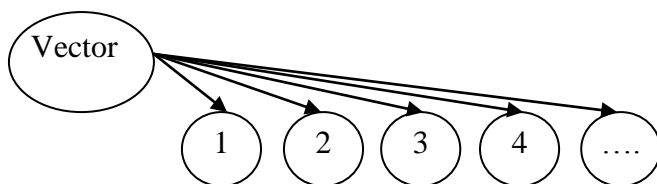
```

Η αρχιτεκτονική φαίνεται στο παρακάτω σχεδιάγραμμα (εικόνα 4.24).



Εικόνα 4.24 Αρχιτεκτονική δημιουργίας κλάσεων δεδομένων για ένα πακέτο του e-kones

Η κάθε μέθοδος που εκτελείται από την μέθοδο PacketBuilder διαβάζει το XML και τοποθετεί τα δεδομένα με την χρήση αντίστοιχης κλάσης προδιαγραφής. Η κάθε μέθοδος με το τέλος του διαβάσματος παράγει μία μορφή δεδομένων που περιέχει τα δεδομένα. Όλες οι μέθοδοι δημιουργούν δομή δεδομένων τύπου Vector, εκτός από την CreatePackageDescriptor η οποία χρησιμοποιεί την κλάση προδιαγραφής της και υλοποιεί αυτήν. Αυτό είναι λογικό αφού ένα πακέτο έχει μόνο μία περιγραφή, ενώ μπορεί να έχει πολλές δραστηριότητες, εταιρους κλπ.



Εικόνα 4.25 Μοντέλο δεδομένων πακέτου

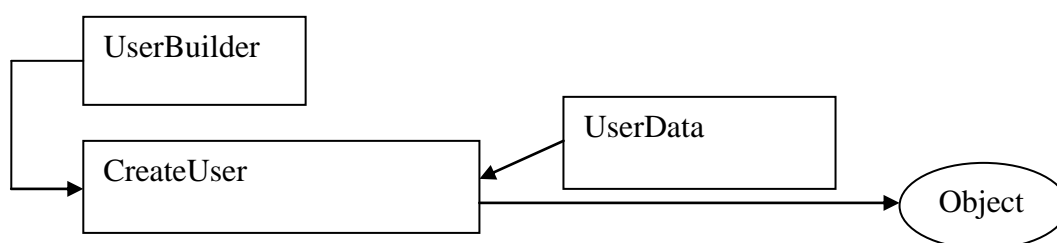
Τα τελικά δεδομένα θα έχουν την παρακάτω μορφή (εικόνα 4.25). Κάθε στοιχείο στην δομή δεδομένων vector μπορεί να είναι τύπου activitydata, bpartnerdata, categorydata κλπ, αντίστοιχα, η δομή δεδομένων

ανάλογα με τα περιεχόμενα της έχει την αντίστοιχη ονομασία. Για παράδειγμα η δομή δεδομένων που έχει σαν στοιχεία τύπου ActivityData και έχει δημιουργηθεί από την μέθοδο CreatePackageActivities, ονομάζεται Activities.

Επιλογή 2, αυτός ο τρόπος λειτουργίας ενεργοποιεί μέθοδο για το κτίσιμο των κλάσεων με πληροφορίες για τον χρήστη της εφαρμογής. Ο κώδικας εκτέλεσης είναι ο ακόλουθος.

```
public void UserBuilder ()
{
    xmcon = new XMLConnect (url);
    CreateUser ();
}
```

Η αρχιτεκτονική φαίνεται παρακάτω (εικόνα 4.26).



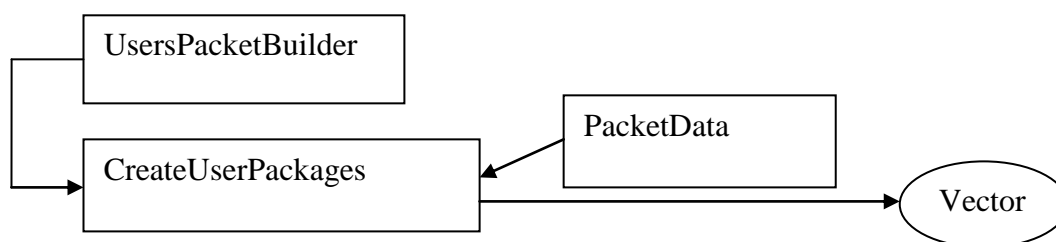
Εικόνα 4.26 Αρχιτεκτονική δημιουργίας κλάσεων δεδομένων για τον χρήστη του e-kones exp.

Η μέθοδος UserBuilder περιλαμβάνει μία μέθοδο που αναλαμβάνει να διαβάσει το αντίστοιχο XML και να τοποθετήσει τα δεδομένα σύμφωνα με την κλάση προδιαγραφής. Στο τέλος του διαβάσματος η μέθοδος παράγει ένα αντικείμενο τύπου UserData που περιέχει τις πληροφορίες χρήστη.

Επιλογή 3, αυτός ο τρόπος λειτουργίας ενεργοποιεί μέθοδο για το κτίσιμο των κλάσεων με πληροφορίες των πακέτων στα οποία έχει εγγραφεί ο χρήστης. Ο κώδικας εκτέλεσης είναι ο ακόλουθος.

```
public void UsersPacketBuilder ()
{
    xmcon = new XMLConnect (url);
    CreateUserPackages ();
}
```

Η αρχιτεκτονική φαίνεται παρακάτω (εικόνα 4.27).



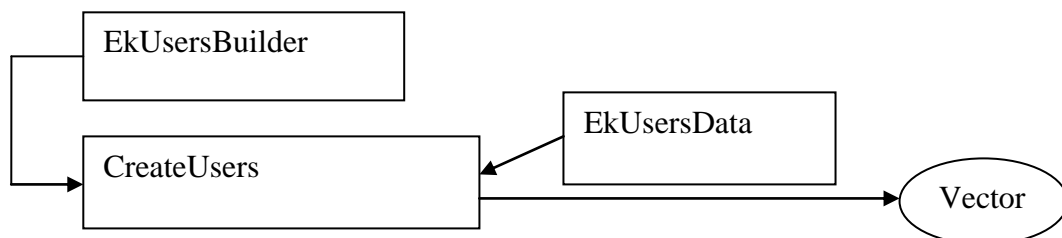
Εικόνα 4.27 Αρχιτεκτονική δημιουργίας κλάσεων δεδομένων τα πακέτα που είναι εγγεγραμμένος ο χρήστης της εφαρμογής στο e-kones

Η μέθοδος UsersPacketBuilder περιλαμβάνει μία μέθοδο που αναλαμβάνει να διαβάσει το αντίστοιχο XML και να τοποθετήσει τα δεδομένα σύμφωνα με την κλάση προδιαγραφής. Στο τέλος του διαβάσματος η μέθοδος παράγει ένα αντικείμενο τύπου Vector με στοιχεία αντικείμενα τύπου PacketData τα οποία φέρουν πληροφορίες για το κάθε πακέτο στο οποίο έχει εγγραφεί ο χρήστης.

Επιλογή 4, αυτός ο τρόπος λειτουργίας ενεργοποιεί μέθοδο για το κτίσιμο των κλάσεων με πληροφορίες των πακέτων στα οποία έχει εγγραφεί ο χρήστης. Ο κώδικας εκτέλεσης είναι ο ακόλουθος.

```
public void EkUsersBuilder()
{
    xmcon = new XMLConnect(url);
    CreateUsers();
}
```

Η αρχιτεκτονική φαίνεται παρακάτω (εικόνα 4.28).



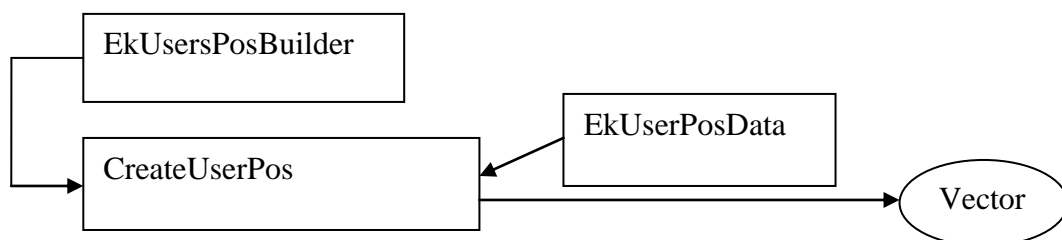
Εικόνα 4.28 Αρχιτεκτονική δημιουργίας κλάσεων δεδομένων για χρήστες ενός πακέτου του e-kones

Η μέθοδος EkUsersBuilder περιλαμβάνει μία μέθοδο που αναλαμβάνει να διαβάσει το αντίστοιχο XML και να τοποθετήσει τα δεδομένα σύμφωνα με την κλάση προδιαγραφής. Στο τέλος του διαβάσματος η μέθοδος παράγει ένα αντικείμενο τύπου Vector με στοιχεία αντικείμενα τύπου EkUsersData τα οποία φέρουν πληροφορίες για τον κάθε χρήστη που είναι εγγεγραμμένος στο ίδιο πακέτο.

Επιλογή 5, αυτός ο τρόπος λειτουργίας ενεργοποιεί μέθοδο για το κτίσιμο των κλάσεων με πληροφορίες των θέσεων των άλλων χρηστών οι οποίοι είναι εγγεγραμμένοι στο ίδιο πακέτο. Ο κώδικας εκτέλεσης είναι ο ακόλουθος.

```
public void EkUserPosBuilder()
{
    xmcon = new XMLConnect(url);
    CreateUserPos();
}
```

Η αρχιτεκτονική φαίνεται παρακάτω (εικόνα 4.29).



Εικόνα 4.29 Αρχιτεκτονική δημιουργίας κλάσεων δεδομένων τις θέσεις χρηστών ενός πακέτου του e-kones

Η μέθοδος `EkUsersPosBuilder` περιλαμβάνει μία μέθοδο που αναλαμβάνει να διαβάσει το αντίστοιχο XML και να τοποθετήσει τα δεδομένα σύμφωνα με την κλάση προδιαγραφής. Στο τέλος του διαβάσματος η μέθοδος παράγει ένα αντικείμενο τύπου `Vector` με στοιχεία αντικείμενα τύπου `EkUserPosData` τα οποία φέρουν πληροφορίες για την γεωγραφική θέση του κάθε χρήστη.

Κατά την εκτέλεση της εφαρμογής το αντικείμενο `DataBuilder` λειτουργεί ως εξής.

```
DataBuilder upkb;  
upkb=new  
DataBuilder("http://hwm.armedassault.info/personal/XMLPackageServlet1  
.xml",1);
```

Δημιουργείται ένα αντικείμενο τύπου `DataBuilder` με όνομα `upkb` και ορίσματα την διεύθυνση διαδικτύου με τον διακοσμητή και τρόπο λειτουργίας. Ο τρόπος λειτουργίας είναι «1». Οπότε θα εκτελεστεί η εσωτερική διεργασία για την δημιουργία δομών δεδομένων σχετικά με το επιλεγμένο πακέτο. Μπορεί κατόπιν η εφαρμογή να προσπελάσει τα δεδομένα με τον ακόλουθο τρόπο.

```
Vector v = new Vector();  
v = upkb.getPackageDays();
```

Στο παραπάνω παράδειγμα η μέθοδος `getPackageDays()` θα επιστρέψει ένα αντικείμενο τύπου `Vector` που θα έχει σαν στοιχεία αντικείμενα τύπου `DayData` με πληροφορίες για την κάθε ημέρα του εκδρομικού πακέτου. Μπορεί να γίνει προσπέλαση του κάθε στοιχείου της δομής `Vector` με μεθόδους που παρέχει η Java για την συγκεκριμένη δομή, όπως φαίνεται παρακάτω

```
for(int k=0;k<v.size();k++)  
{  
    DayData dt = new DayData();  
    dt=v.elementAt(k);  
    String str = dt.getDescription();  
    System.out.println(str);  
}
```

Στο παραπάνω παράδειγμα διαβάζονται όλα τα δεδομένα που περιέχει η δομή `Vector` με ονομασία `v`, κατόπιν κάθε στοιχείο τοποθετείται σε ένα στιγμιότυπο μία κλάσης τύπου `DayData` με ονομασία `dt` και με μία μέθοδο ανακτάται η περιγραφή της συγκεκριμένης μέρας η οποία τελικός τοποθετείται σε μία μεταβλητή τύπου `String` και προβάλλεται στην έξοδο συστήματος.

Ακολουθούν οι μέθοδοι που παρέχει η κλάση DataBuilder

Τύπος Επιστροφής Μεθόδου	Όνομα Μεθόδου	Χρήση
void	EkUserPosBuilder()	Κεντρική μέθοδος, καλεί μεθόδους ανάγνωσης XML δεδομένων για την γεωγραφική θέση χρηστών.
void	EkUsersBuilder()	Κεντρική μέθοδος, καλεί μεθόδους ανάγνωσης XML δεδομένων για τους χρήστες εγγεγραμμένους στο ίδιο πακέτο
java.util.Vector	getEkonesUsersData()	Επιστρέφει δεδομένα χρηστών εγγεγραμμένους στο ίδιο πακέτο.
EkUserPosData	getEkonesUsersPositionData()	Επιστρέφει γεωγραφική θέση χρηστών εγγεγραμμένους στο ίδιο πακέτο.
java.util.Vector	getPackageActivities()	Επιστρέφει της δραστηριότητες του πακέτου.
java.util.Vector	getPackageCategories()	Επιστρέφει τις κατηγορίες του πακέτου.
java.util.Vector	getPackageDays()	Επιστρέφει τι ημέρες του πακέτου.
java.util.Vector	getPackageLocations()	Επιστρέφει τις τοποθεσίες του πακέτου.
java.util.Vector	getPackageParameters()	Επιστρέφει τις παραμέτρους του πακέτου
java.util.Vector	getPackagePartners()	Επιστρέφει τους εταίρους του πακέτου.
PacketData	getPacketDescriptor()	Επιστρέφει την γενική περιγραφή του πακέτου.
UserData	getUserData()	Επιστρέφει τα δεδομένα του χρήστη.
java.util.Vector	getUserPackages()	Επιστρέφει τα πακέτα που είναι εγγεγραμμένος ο χρήστης
void	PacketBuilder()	Κεντρική μέθοδος, καλεί μεθόδους ανάγνωσης XML δεδομένων για την δημιουργία του τρέχοντος πακέτου.
void	UserBuilder()	Κεντρική μέθοδος, καλεί μεθόδους ανάγνωσης XML δεδομένων για πληροφορίες χρήστη.
void	UsersPacketBuilder()	Κεντρική μέθοδος, καλεί μεθόδους ανάγνωσης XML δεδομένων για τα πακέτα που είναι εγγεγραμμένος ο χρήστης.

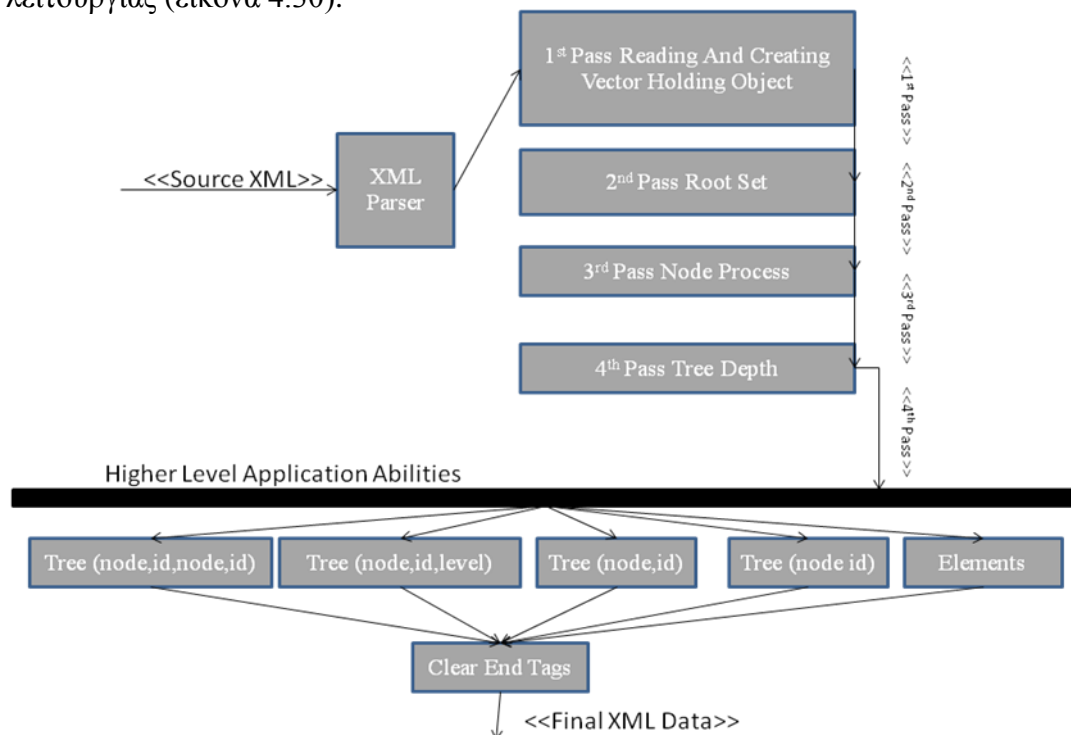
Οι κλάσεις προδιαγραφής που υπάρχουν είναι οι ακόλουθες.

Όνομα Κλάσης Προδιαγραφής	Λειτουργία
ActivityData	Περιέχει δεδομένα για τις δραστηριότητες.
BPartnerData	Περιέχει δεδομένα για τους εταίρους.
CategoryData	Περιέχει δεδομένα για τις κατηγορίες του πακέτου.
ConfigMapData	Περιέχει δεδομένα για τον χάρτη (Δεν χρησιμοποιείται).
DayData	Περιέχει δεδομένα για τις ημέρες του πακέτου.
EkUserPosData	Περιέχει δεδομένα για τις γεωγραφικές θέσεις των χρηστών.
EkUsersData	Περιέχει πληροφορίες για χρήστες εγγεγραμμένους στο ίδιο πακέτο.
LocationData	Περιέχει δεδομένα για τις τοποθεσίες του πακέτου.
PacketData	Περιέχει πληροφορίες για τα πακέτα που είναι εγγεγραμμένος ο χρήστης.
ParameterData	Περιέχει πληροφορίες παραμέτρων του πακέτου.
UserData	Περιέχει πληροφορίες χρήστη.

Η κάθε κλάση προδιαγραφής δεδομένων προσφέρει αντίστοιχες μεθόδους, με βάση τα δεδομένα που τηρεί. Όλες οι μέθοδοι που χρησιμοποιούν αυτές οι κλάσεις είναι τύπου set/get και δεν περιέχουν μεθόδους υπολογισμών.

XML Parser

Ένα βασικό συστατικό αυτής της εφαρμογής είναι ο XML Parser που αναπτύχθηκε γι' αυτήν. Ο συγκεκριμένος parser σε αυτή την έκδοση μπορεί να διαβάζει αρχείο XML οποιασδήποτε μορφής και έχει καλή αντοχή σε σφάλματα. Ο τρόπος που διαβάζει είναι σειριακός ξεκινώντας από την αρχή του εγγράφου μέχρι το τέλος. Μετά την ανάγνωση προσπαθεί να δημιουργήσει μία δενδρική δομή στα δεδομένα ώστε να είναι εύκολη η προσπέλαση τους από τα υψηλότερα στρώματα της εφαρμογή χρησιμοποιώντας δομή δεδομένων τύπου Vector . Λόγω του τρόπου που διαβάζει όμως δεν ενδείκνυται για μεγάλα αρχεία. Για την υλοποίηση του XML Parser δημιουργήθηκε επιπρόσθετο υποστηρικτικό πρόγραμμα που θα παρουσιασθεί στο παράρτημα. Ακολουθεί παρουσίαση της αρχιτεκτονικής και του τρόπου λειτουργίας (εικόνα 4.30).



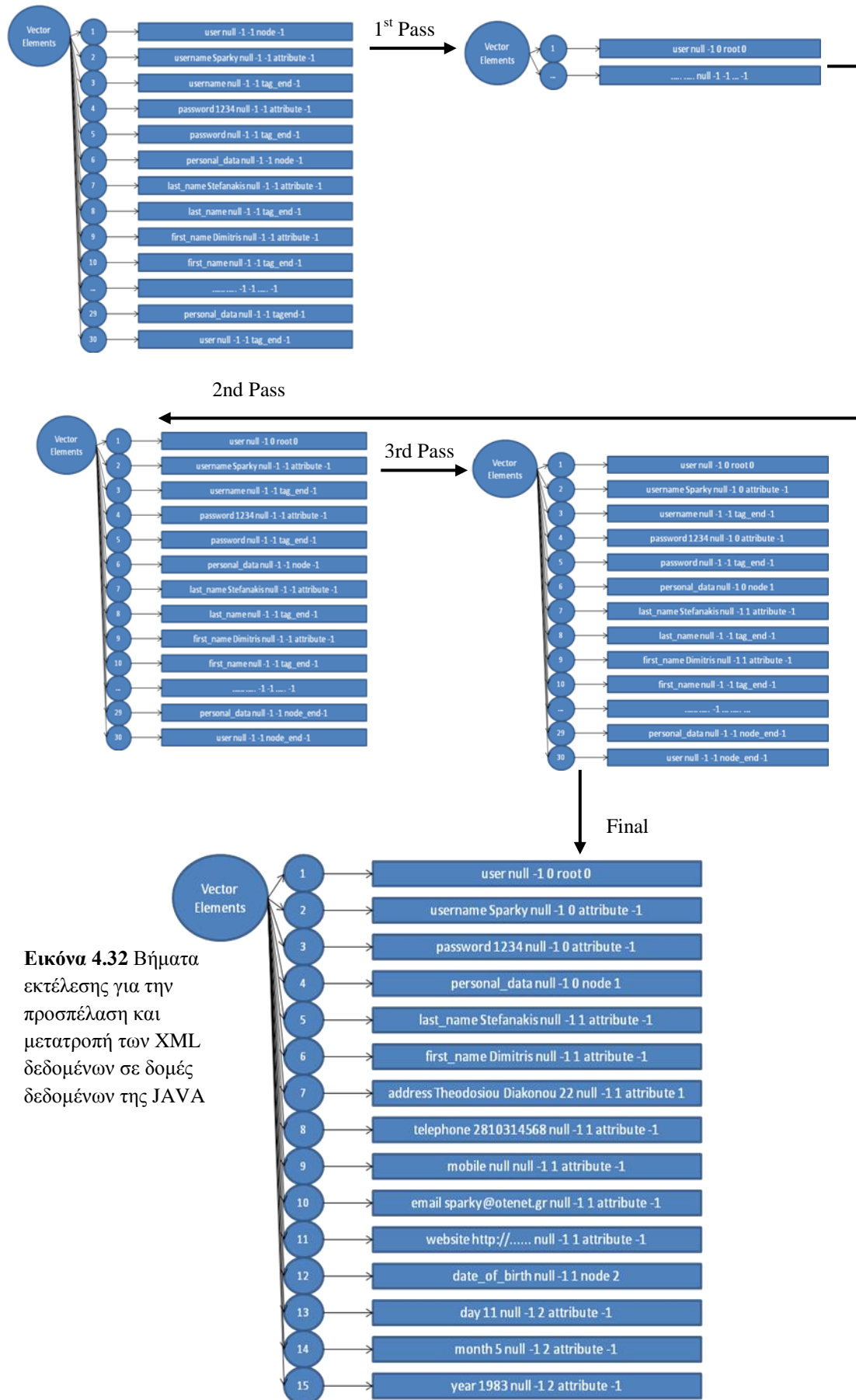
Εικόνα 4.30 Αρχιτεκτονική XML Parser

```

<user>
  <username>Sparky</username>
  <password>1234</password>
  <personal_data>
    <last_name>Stefanakis</last_name>
    <first_name>Dimitris</first_name>
    <address>Theodosiou Diakonou 22</address>
    <telephone>2810314568</telephone>
    <mobile>>null</mobile>
    <email>sparky@otenet.gr</email>
    <website>http://ofp.gamepark.cz/_hosted/mmstudios</website>
    <date_of_birth>
      <day>11</day>
      <month>5</month>
      <year>1983</year>
    </date_of_birth>
  </personal_data>
</user>

```

Εικόνα 4.31 Δείγμα XML αρχείου



Εικόνα 4.32 Βήματα εκτέλεσης για την προσπέλαση και μετατροπή των XML δεδομένων σε δομές δεδομένων της JAVA

Παράδειγμα χρήσης του Parser από την εφαρμογή

```
xmcon = new XMLConnect(url);
```

Υλοποιείται ένα αντικείμενο τύπου XMLConnect με όρισμα τη διεύθυνση διαδικτύου που βρίσκεται το XML.

Το παρακάτω τμήμα κώδικα αποτελεί κομμάτι της εφαρμογής. Στην εφαρμογή ο ρόλος του είναι να υλοποιήσει κλάσεις τύπου ActivityData και να γεμίσει ένα Vector με αυτές. Αυτό που θα παρουσιασθεί στο σημείο αυτό είναι οι μέθοδοι που μπορεί να διαβαστεί πλέον το XML αφού έχει ήδη δεχθεί την επεξεργασία που αναφέρθηκε προηγουμένως. Να σημειωθεί ότι η εφαρμογή χρησιμοποιεί τα Vector δεδομένα που παράγονται από αυτό το κομμάτι κώδικα.

```
private void CreatePackageActivities()
{
    .....
    xmpl = xmcon.getXMLTree("activities","null",0);
    int[] activities_id;
    int activities_count=0;

    while (x<xmpl.size())
    {
        xm = (XMLData) xmpl.elementAt(x);
        if(xm.getTag().compareTo("activity")==0)
        {
            activities_count++;
        }
        x++;
    }
    .....
    for(i=0;i<activities_count;i++)
    {
        xmpl =
xmcon.getXMLTree("activity",String.valueOf(activities_id[i]));
        ad = new ActivityData();
        x=0;
        while (x<xmpl.size())
        {
            xm = (XMLData) xmpl.elementAt(x);
            ad.setActivityId(activities_id[i]);
            if(xm.getTag().compareTo("description")==0)
            {
                ad.setActivityDescription(xm.getContent());
            }
            if(xm.getTag().compareTo("departure")==0)
            {
                k=0;
                if(xm.isNode())
                {
                    xmpl2 = xmcon.getXMLTree(xm.getNodeId());
                    while (k<xmpl2.size())
                    {
                        xm2 = (XMLData) xmpl2.elementAt(k);
                        ad.setActivityDeparture(Integer.parseInt(xm2.getContent()));
                        k++;
                    }
                }
            }
            .....
            .....
            Activities.addElement(ad);
        }
    }
}
```

Η μέθοδος επιστρέφει ένα αντικείμενο τύπου Vector με δενδρική δομή και με αντικείμενο ρίζας ονόματος "activities"

Η μέθοδος xm.getTag() επιστρέφει την ονομασία του XML tag/attribute για το συγκεκριμένο.

Η μέθοδος xm.isNode() ελέγχει αν το συγκεκριμένο tag είναι attribute ή Node

Η μέθοδος xm.getContent() επιστρέφει το περιεχόμενο ανάμεσα σε δύο attributes.

Μέθοδος παρόμοια με την πρώτη η οποία επιστρέφει ένα υποδέντρο αλλά όχι με βάση String αλλά με βάση id. Ο Parser κατά την διάρκεια του Parsing δίνει NodeIds σε κόμβους και RootIds σε κόμβους και attributes. Η βασική διαφορά είναι ότι ένας κόμβος είναι κόμβος γιατί έχει NodeId και πατέρα αυτών που αναγράφεται στο RootId πεδίο του. Ένα attribute όμως έχει πάντα -1 σαν NodeId και Πάντα το id του κόμβου που τον περιέχει στο RootId πεδίο του. Στην ουσία αυτή η κατάταξη είναι που μας επιτρέπει να έχουμε λειτουργία δένδρου.

Αυτό που είναι εμφανές από το προηγούμενο παράδειγμα είναι ότι σε αυτό το επίπεδο με την χρήση απλών μεθόδων μπορεί να γίνει ανάκτηση των δεδομένων που χρειάζονται από το XML και μόνο αυτών.

Τύπος Επιστροφής Μεθόδου	Όνομα Μεθόδου	Χρήση
java.util.Vector	getTree(int root)	Επιστρέφει δενδρική δομή με βάση αναγνωριστικό κωδικό ρίζας.
java.util.Vector	getTree(java.lang.String tag_name , java.lang.String id)	Επιστρέφει δενδρική δομή με βάση ονόματος επιθυμητού κόμβου ρίζας και αναγνωριστικού κωδικού.
java.util.Vector	getTree(java.lang.String tag_name , java.lang.String id, int level)	Επιστρέφει δενδρική δομή με βάση ονόματος επιθυμητού κόμβου ρίζας, αναγνωριστικού κωδικού και βάθος.
java.util.Vector	getTree(java.lang.String tag_name , java.lang.String id, java.lang.String where, java.lang.String where_id)	Επιστρέφει δενδρική δομή με βάση ονόματος επιθυμητού κόμβου ρίζας, αναγνωριστικού κωδικού και ονόματος δένδρου πατέρα και κωδικό..
java.lang.String	getId()	Επιστρέφει τον κωδικό.
int	getNodeId()	Επιστρέφει τον κωδικό κόμβου.
int	getRootId()	Επιστρέφει τον κωδικό της ρίζας.
java.lang.String	getTag()	Επιστρέφει τον τίτλο της ετικέτας.
java.lang.String	getType()	Επιστραφεί τον τύπο (node, attribute, root).
boolean	isAttribute()	Ελέγχει αν είναι γνώρισμα.
boolean	isContentNull()	Ελέγχει αν έχει άδειο περιεχόμενο.
boolean	isNode()	Ελέγχει αν είναι κόμβος.

ΚΕΦΑΛΑΙΟ 5

Σενάριο Χρήσης της εφαρμογής E-kones Experience



Παρακάτω θα παρουσιασθεί ένα σενάριο χρήσης της εφαρμογής, παρουσιάζοντας πως μπορεί ο χρήστης να συνδεθεί με την υπηρεσία του e-kones, να επιλέξει και να ανακτήσει ένα πακέτο που είναι εγγεγραμμένος καθώς και τις λειτουργίες που του προφέρονται από την εφαρμογή για να «διαδράσει» με το πακέτο αυτό, την στιγμή που το βιώνει.

Σύνδεση με την υπηρεσία του e-kones



Εικόνα 5.1 φόρμα εισόδου.

Κατά την εκκίνηση ο χρήστης καλείται να δώσει όνομα χρήστη και κωδικό πρόσβασης (εικόνα 5.1). Η εφαρμογή ελέγχει αν ο χρήστης υπάρχει στην υπηρεσία (έχει εγγραφεί τουλάχιστον σε ένα πακέτο) και αποστέλλει στην εφαρμογή το αρχείο με τα πακέτα στα οποία έχει εγγραφεί ο χρήστης.

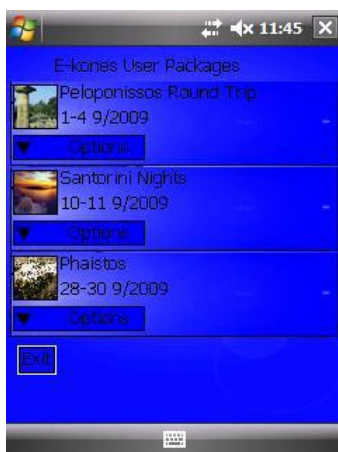


Εικόνα 5.2 Κεντρική φόρμα επιλογών.

Σε περίπτωση επιτυχής εισόδου (αν ο χρήστης υπάρχει στην βάση των χρηστών του E-kones) εμφανίζεται η κεντρική φόρμα επιλογών (εικόνα 5.2). Πλέον η εφαρμογή γνωρίζει τον χρήστη όποτε υπάρχει μήνυμα καλοσωρήματος πάνω ψηλά μαζί με την ώρα. Εντούτοις μόνο τρεις επιλογές είναι ενεργές (UserInfo, Currency Calculator, Select Package) ενώ η εφαρμογή προτρέπει τον χρήστη να επιλέξει πακέτο.

Επιλογή Πακέτου

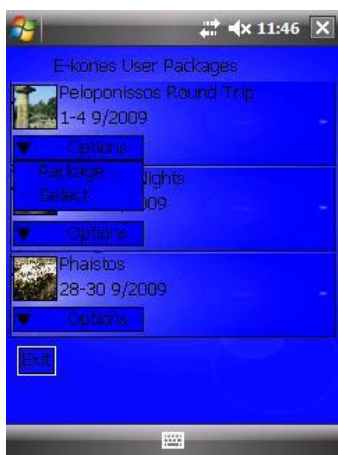
Για να επιλέξει ο χρήστης πακέτο, επιλέγει το εικονίδιο του πακέτου από το κεντρικό μενού.



Εικόνα 5.3 φόρμα επιλογής πακέτου.

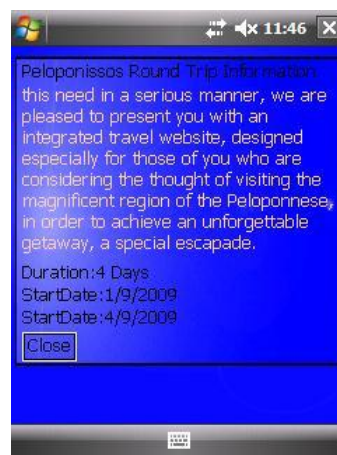
Η επόμενη διεπαφή (φόρμα επιλογής πακέτου) παρουσιάζει όλα τα πακέτα στα οποία έχει εγγραφεί ο χρήστης (εικόνα 5.3). Οι πληροφορίες που εμφανίζονται για το κάθε πακέτο είναι αντίστοιχο εικονίδιο, τίτλος και ημερομηνία αρχής και λήξης του πακέτου. Κάθε πακέτο έχει ένα `istlMenuItem` που προσφέρει δύο επιλογές (Package Information και Select) (εικόνα 5.4).

Η επιλογή Package Information παρέχει πληροφορίες για το αντίστοιχο πακέτο, όπως φαίνεται



Εικόνα 5.4 φόρμα επιλογής πακέτου. Επιλογές `stlMenuItem`

στην δεξιά εικόνα (εικόνα 5.5). Οι πληροφορίες περιλαμβάνουν τον τίτλο του πακέτου πάνω ψηλά. Κατόπιν μια συνοπτική περιγραφή για το πακέτο, την διάρκεια σε ημέρες και πιο συγκεκριμένα ημερομηνία έναρξης και τερματισμού. Το κουμπί



close τερματίζει αυτό το παράθυρο και επιστρέφει

Εικόνα 5.5 φόρμα πληροφοριών πακέτου.

στην φόρμα επιλογής πακέτου (εικόνα 5.3). Η επιλογή Select, του `istlMenuItem` επιλέγει το συγκεκριμένο πακέτο και τερματίζει την φόρμα επιλογής πακέτου.




Εικόνα 5.6 κεντρική φόρμα

Πλέον όλες οι επιλογές στην κεντρική φόρμα έχουν ενεργοποιηθεί ενώ η εφαρμογή πάνω ψηλά ενημερώνει τον χρήστη για το επιλεγμένο πακέτο (εικόνα 5.6). Από αυτές τις επιλογές η μόνη με διακοσμητικό χαρακτήρα που δεν έχει υλοποιηθεί είναι η Currency calculator. Οι υπόλοιπες θα αναλυθούν παρακάτω ξεκινώντας από τις απλούστερες.

Προσωπικές Πληροφορίες Χρήστη




Εικόνα 5.7 φόρμα
πληροφοριών χρήστη

Αν πατηθεί το κουμπί User Info  εμφανίζεται η φόρμα πληροφοριών χρήστη (εικόνα 5.7). Σε αυτή την φόρμα παρουσιάζονται προσωπικές πληροφορίες του χρήστη όπως όνομα, επίθετο, διεύθυνση, τηλέφωνο, κλπ, όπως ο ίδιος ο χρήστης τις είχε καταχωρίσει στην υπηρεσία του e-kones. Το κουμπί close τερματίζει την φόρμα πληροφοριών χρήστη και επιστρέφει στην κεντρική φόρμα.

Πληροφορίες Άλλων Χρηστών Εγγεγραμμένοι στο ίδιο Πακέτο



Εικόνα 5.8 φόρμα
πληροφοριών χρηστών


Το κουμπί Users Info  εμφανίζει την φόρμα πληροφοριών άλλων χρηστών που είναι εγγεγραμμένοι στο ίδιο πακέτο (εικόνα 5.8). Η υπηρεσία του e-kones στέλνει την λίστα με τους χρήστες αυτούς αμέσως μετά την επιλογή πακέτου, γι' αυτό το λόγο άλλωστε η συγκεκριμένη επιλογή είναι απενεργοποιημένη μέχρι να επιλεγεί εκδρομικό πακέτο.

Σε αυτήν την περίπτωση ο διακοσμιτής του E-kones για λόγους προσωπικών δεδομένων στέλνει μόνο ονοματεπώνυμο και username στην εφαρμογή.

Χρήση Του Ημερολογίου



Εικόνα 5.9 Ημερολόγιο

Όπως αναφέρθηκε προηγουμένως το αντικείμενο του ημερολογίου προσφέρει χρονική αναπαράσταση του πακέτου (εικόνα 5.9). Η εκκίνηση της φόρμας ημερολογίου γίνεται από το κουμπί Calendar .

Μόλις ξεκινήσει το αντικείμενο ημερολογίου από προεπιλογή δείχνει τον τρέχων χρόνο και μήνα. Όπως αναλύθηκε προηγουμένως (Αρχιτεκτονική Αντικειμένου Ημερολογίου) αποτελείται από 3 υπό αντικείμενα.

Υπο-Αντικείμενο Χρόνου

Το υποαντικείμενο χρόνου παρουσιάζει τον τρέχων χρόνο και προσφέρει δύο κουμπιά ολίσθησης. Το δεξί/αριστερό βέλος ολισθαίνει κατά ένα χρόνο πριν/μετά αντίστοιχα από τον τρέχων χρόνο που φαίνεται στο κέντρο (εικόνα 5.10).



Εικόνα 5.10 Υπο-αντικείμενο χρόνου

Υπο-Αντικείμενο Μήνα

Το υποαντικείμενο μήνα παρουσιάζει τον τρέχων μήνα και προσφέρει δύο κουμπιά ολίσθησης. Το δεξί/αριστερό βέλος ολισθαίνει κατά ένα μήνα πριν/μετά αντίστοιχα από τον τρέχων μήνα που φαίνεται στο κέντρο (εικόνα 5.11). Σε περίπτωση που ο μήνας γίνει Ιανουάριος ή Δεκέμβριος το αντίστοιχο κουμπί ολίσθησης γίνεται κόκκινο μη επιτρέποντας περαιτέρω ολίσθηση.



Εικόνα 5.11 Υπο-αντικείμενο μήνα

Υπο-Αντικείμενο Ημέρες Μήνα

Το κεντρικό σημείο του ημερολογίου είναι το υποαντικείμενο ημερών. Το οποίο μεταβάλλεται κάθε φορά που γίνεται ολίσθηση στον χρόνο/μήνα παρουσιάζοντας τις αντίστοιχες μέρες του χρόνου/μήνα (εικόνα 5.12).

Πάνω ψηλά βρίσκονται τα αρχικά των ημερών ξεκινώντας από την Κυριακή και καταλήγοντας στο Σάββατο. Οι ημέρες κατανέμονται μέσα στο αντικείμενο ημερών με βάση τις αντίστοιχες ημερομηνίες/μέρες του χρόνου/μήνα.



Εικόνα 5.12 Υπο-αντικείμενο ημέρες μήνα / χρόνου

Η εικόνα στο περιθώριο εκφράζει την εποχή. Συνολικά υπάρχουν 4 εικόνες για κάθε εποχή. Οι εικόνες αυτές αποτελούν παράδειγμα theme και μπορούν να μεταβληθούν εύκολα. Η κάθε ημερομηνία προσφέρει αλληλεπίδραση με τον χρήστη. Στο αυτό το σενάριο χρήσης θα επιλεγεί η 22 Ιουλίου.



Εικόνα 5.13 Φόρμα επισκόπησης ημέρας.



Εικόνα 5.14 Φόρμα εισαγωγής σημείωσης χρήστη.



Εικόνα 5.15 Φόρμα εισαγωγής σημείωσης χρήστη.

εμφανής η έλλειψη κουμπιού Μεταβολής και Διαγραφής Σημείωσης, τα οποία λόγω χρόνου και αξιολόγησης σημαντικότητας δεν δημιουργήθηκαν.

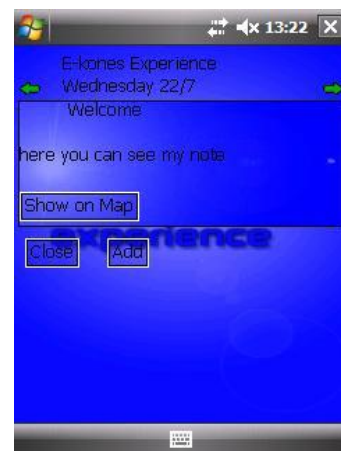
Η επιλογή της ημέρας θα εκκινήσει την φόρμα επισκόπησης ημέρας. Σε αυτήν την φόρμα εμφανίζονται οι σημειώσεις που πιθανόν έχει προσθέσει ο χρήστης ή η ημέρα του αν ο χρήστης βρίσκεται σε κάποια από τις ημερομηνίες του πακέτου. Στην αριστερή εικόνα η φόρμα ημέρας εμφανίζεται κενή αφού δεν αποτελεί ημέρα πακέτου αλλά ούτε έχει προσωπική σημείωση.

Η φόρμα επισκόπησης ημέρας αποτελείται από τον τίτλο που βρίσκεται πάνω ψηλά. Σε αυτή την περίπτωση τον γενικό τίτλο e-kones Experience και την τρέχουσα ημέρα και ημερομηνία που βρίσκεται από κάτω από τον τίτλο. Δεξιά και αριστερά υπάρχουν τα βέλη ολίσθησης ημερομηνίας (δεν έχουν υλοποιηθεί). Ακριβώς κάτω από την Ημερομηνία υπάρχει ο κεντρικός υποδοχέας ο οποίος είναι άδειος, προσφέροντας μόνο τα κουμπιά close και add (εικόνα 5.13).

Το πάτημα του κουμπιού add προκαλεί εκκίνηση της φόρμας εισαγωγής προσωπικής σημείωσης όπως φαίνεται στην εικόνα 5.14. Σε αυτή την φόρμα ο χρήστης μπορεί να προσθέσει την προσωπική του σημείωση για την συγκεκριμένη ημέρα. Η φόρμα προσωπικών σημειώσεων προσφέρει εισαγωγή τίτλου, κείμενο σημείωσης, καθώς και την δυνατότητα αντιστοίχισης αυτής της σημείωσης πάνω στον χάρτη με την επιλογή (Mark On Map).

Μόλις ο χρήστης προσθέσει την προσωπική του σημείωση, το κουμπί Apply καταχωρεί αυτή την σημείωση, και το κουμπί close τερματίζει την φόρμα εισαγωγής προσωπικής σημείωσης και επιστρέφει στην φόρμα επισκόπησης ημέρας. Στο παράδειγμα, προστέθηκε η σημείωση που φαίνεται στην εικόνα 5.15.

Η σημείωση του χρήστη είναι εμφανής στην φόρμα επισκόπησης ημέρας (εικόνα 5.16). Να σημειωθεί σε αυτό το σημείο ότι είναι



Εικόνα 5.16 Ενημερωμένη φόρμα επισκόπησης ημέρας



Εικόνα 5.17 Ενημέρωση ημερολογίου με ημέρα που περιέχει σημείωση.

Ο χρήστης μπορεί να πατήσει το κουμπί Add ώστε να προσθέσει και άλλη σημείωση (δεν υπάρχει περιορισμός για το πλήθος σημειώσεων χρήστη ανά ημέρα) ή να πατήσει το κουμπί Close και να επιστρέψει στον ημερολόγιο. Στο σενάριο αυτό δεν θεωρείται επιθυμητή η εισαγωγή επιπρόσθετης σημείωσης, οπότε γίνεται επιλογή του κουμπιού Close και επιστροφή στο Ημερολόγιο.

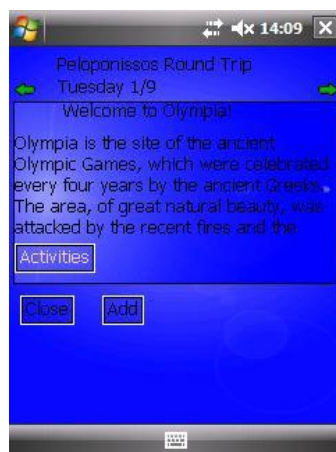
Αυτό που είναι εμφανές στο ημερολόγιο (όπως φαίνεται στην αριστερή εικόνα) είναι ότι η ημερομηνία με προσωπική σημείωση έχει σημειωθεί με κόκκινο χρώμα (εικόνα 5.17). Γενικώς σε αυτή την εφαρμογή υπάρχουν δύο κύρια χρώματα. Το κόκκινο εκφράζει σημειώσεις του χρήστη ενώ το μπλε του E-kones.



Εικόνα 5.18 Σημειωμένες από την εφαρμογή οι ημέρες του πακέτου

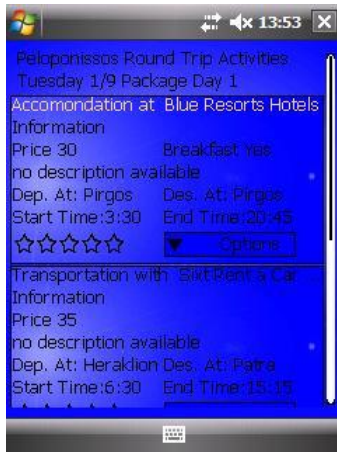
Για την συνέχεια αυτού του σεναρίου θα γίνει ολίσθηση κατά δύο μήνες μέχρι τον μήνα Σεπτέμβριο όπου είναι το επιλεγμένο πακέτο. (Σημείωση. Η μετάβαση μέχρι τον επιθυμητό μήνα του πακέτου μπορεί να φαίνεται κάπως παράξενη. Για την ακρίβεια δεν είναι, δεδομένου ότι το ημερολόγιο κατά την εκκίνηση εμφανίζει τον τρέχων μήνα και χρόνο και δεδομένου ότι ο χρήστης θα χρησιμοποιεί την εφαρμογή κατά την διάρκεια του Πακέτου δηλαδή μήνα Σεπτέμβριο θα έχει πάντα αυτό τον μήνα κατά την εκκίνηση ως προεπιλογή. Συνεπώς δεν θα χρειάζεται να μεταβαίνει στον επιθυμητό μήνα). Στην εικόνα 5.18 παρατηρούνται τα εξής:

1. Οι ημέρες του πακέτου έχουν σημειωθεί με μπλε χρώμα (όπως προαναφέρθηκε).
2. Η εικόνα στο περιθώριο έχει αλλάξει αντικατοπτρίζοντας την αντίστοιχη εποχή.



Εικόνα 5.19 Φόρμα δραστηριοτήτων.

Επιλέγοντας την Τρίτη 1 Σεπτεμβρίου ανοίγει πλέον η γνώριμη φόρμα επισκόπησης ημέρας. Μπορεί να παρατηρηθεί η διαφορά από την κενή ημέρα. Ο Τίτλος παρέχει πληροφορίες για την ονομασία του πακέτου, αντί τον γενικό τίτλο e-kones Experience, ενώ στον κεντρικό υποδοχέα υπάρχει η περιγραφή της ημέρας, καθώς και ένα επιπρόσθετο κουμπί με ονομασία Activities. Το πάτημα αυτού του κουμπιού εκκινεί την φόρμα δραστηριοτήτων επιτρέποντας στον χρήστη να δει πληροφορίες για τις δραστηριότητες της ημέρας.



Εικόνα 5.19 Φόρμα δραστηριοτήτων.



Εικόνα 5.20 Φόρμα δραστηριοτήτων.

Στην αριστερή εικόνα (εικόνα 5.19) εμφανίζεται η φόρμα δραστηριοτήτων ημέρας. Στο επάνω μέρος υπάρχει ο τίτλος του πακέτου, ακριβώς από κάτω Ημερομηνία καθώς και ποια ημέρα του πακέτου αποτελεί η συγκεκριμένη. Στον κεντρικό υποδοχέα η κάθε δραστηριότητα ενθυλακώνεται μέσα σε ένα δικό της υποδοχέα. Στην περίπτωση όπως αυτή που οι δραστηριότητες είναι αρκετές με αποτέλεσμα να μην χωράνε κατά ύψος στην οθόνη εμφανίζεται δεξιά αντικείμενο ολίσθησης που επιτρέπει στον χρήστη να κινηθεί χαμηλότερα στην οθόνη (εικόνα 5.20). Ο κάθε υποδοχέας δραστηριότητας περιλαμβάνει τις εξής πληροφορίες.

1. Τίτλος δραστηριότητας.
2. Πληροφορίες σχετικές με την δραστηριότητα όπως τιμή, τόπος άφιξης, αναχώρησης, ώρα άφιξης/αναχώρησης.
3. Το πολύ σημαντικό αντικείμενο `istlRateItem` που επιτρέπει στον χρήστη να αξιολογήσει την συγκεκριμένη δραστηριότητα / πάροχο.
4. Τέλος ένα `istlMenuItem` που προσφέρει περισσότερες επιλογές



Εικόνα 5.21 `IstlMenuItem` για την συγκεκριμένη δραστηριότητα.

Σε αυτό το σενάριο χρήσης θα γίνει η επιλογή του τελευταίου εταίρου. Πατώντας στο `istlMenuItem` με ονομασία `Options` εμφανίζεται ένα μενού επιλογών με επιλογές `Business Partner Information`, `Location Information`, `Note`, `Show On Map`, `Submit` (εικόνα 5.21).



Εικόνα 5.22 Φόρμα Πληροφοριών εταίρου.

Η επιλογή `Business Partner Information` εμφανίζει την φόρμα πληροφοριών εταίρου (εικόνα 5.22). Από αυτήν την φόρμα ο χρήστης μπορεί να πάρει πληροφορίες για τον συγκεκριμένο εταίρο όπως τηλέφωνο, διεύθυνση κλπ. Το κουμπί `Close`



Εικόνα 5.23 Φόρμα Πληροφοριών περιοχής.

κλείνει την φόρμα και επιστρέφει στην φόρμα δραστηριοτήτων. Η επιλογή Location Information εμφανίζει την φόρμα πληροφοριών περιοχής (εικόνα 5.23), από όπου ο χρήστης μπορεί να δει πληροφορίες για την περιοχή που βρίσκεται ο εταίρος.



Εικόνα 5.24 Φόρμα εισαγωγής σημείωσης.



Εικόνα 5.26 Φόρμα δραστηριοτήτων με αξιολόγηση.



Εικόνα 5.27 Αξιολόγηση και υποβολή αυτής στην εφαρμογή.

Η επιλογή Note ανοίγει την φόρμα σημειώσεων χρήστη (εικόνα 5.24), η συγκεκριμένη φόρμα σημειώσεων είναι διαφορετική από αυτήν που παρουσιάστηκε προηγουμένως. Το κουμπί Mark On Map δεν προσφέρεται, αφού δεν έχει νόημα να σημειωθεί στον χάρτη η θέση μίας δραστηριότητας του E-kones, αφού είναι ήδη μαρκαρισμένη. Καθώς και η δυνατότητα εισαγωγής τίτλου, αφού εννοείται ότι η σημείωση που θα εισαχθεί αναφέρεται στην συγκεκριμένη δραστηριότητα.

Για τις ανάγκες του σεναρίου χρήσης θα γραφτεί το κείμενο “quality food” που θα εκφράζει την ευχαρίστηση του χρήστη για το φαγητό του συγκεκριμένου εταίρου με αυτήν την σημείωση, όπως φαίνεται στην εικόνα δεξιά (εικόνα 5.25). Κατόπιν το κουμπί Apply θα ενημερώσει την εφαρμογή και το κουμπί Close θα



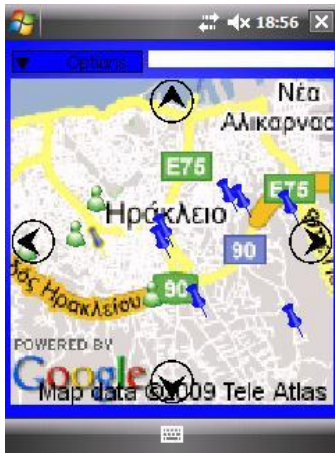
Εικόνα 5.25 Συμπληρωμένη φόρμα εισαγωγής σημείωσης.

τερματίσει την φόρμα εισαγωγής σημειώσεων και θα επιστρέψει στην φόρμα δραστηριοτήτων.

Το τελευταίο βήμα είναι η αξιολόγηση, αφού ήδη ο χρήστης έβαλε μία σημείωση μπορεί να προσθέσει μια συνολική αξιολόγηση σε αυτόν το εταίρο. Την δυνατότητα αυτήν την προσφέρει το `istlRateItem` που αναλύθηκε προηγουμένως. Ο χρήστης μπορεί να αξιολογήσει τον εταίρο από 1 έως 5 αστέρια, σε αυτό το σενάριο χρήσης αξιολογείται με 5 αστέρια (εικόνα 5.26).

Κατόπιν για να ενημερωθεί η εφαρμογή για αυτήν την αξιολόγηση, ο χρήστης από το μενού Options, πατάει την επιλογή Submit (εικόνα 5.27).

Η τελευταία πολύ σημαντική επιλογή για αυτό το σενάριο είναι η επιλογή Show On Map, με την οποία ο χρήστης μπορεί πλέον να δει στον χάρτη την θέση του



Εικόνα 5.28 Εμφάνιση χάρτη με σημειωμένο τον επιλεγμένο εταίρο.

Από το μενού Option, η επιλογή Close κλείνει το αντικείμενο istlMap και επιστρέφει στην φόρμα δραστηριοτήτων. Θα ολοκληρωθεί το σενάριο χρήσης του ημερολογίου με το πάτημα του κουμπιού close που βρίσκεται στο κάτω μέρος.

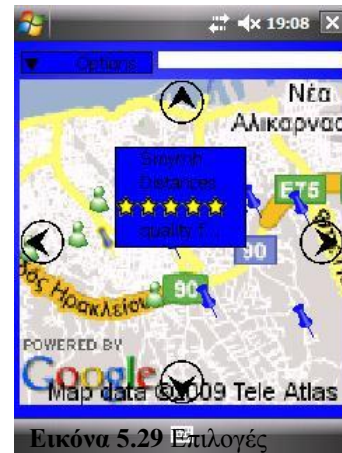


Εικόνα 5.30 Φόρμα επισκόπησης ημέρας με δραστηριότητα πακέτου και προσωπική σημείωση.

πριν παρουσιασθεί το σενάριο χρήσης του αντικειμένου istlMap για λόγους εμπλουτισμού αυτού του σεναρίου θα προστεθεί και μία προσωπική σημείωση στην συγκεκριμένη ημέρα πατώντας το κουμπί Add για εισαγωγή σημείωσης σε κενή ημέρα με τον τρόπο που αναλύθηκε πριν.

Ο χρήστης έβαλε μία σημείωση με τίτλο “holidays” και κείμενο “so far so good”. Αυτό που είναι εμφανές είναι ότι η σημείωση του χρήστη μπήκε κάτω από την περιγραφή της ημέρας του πακέτου (εικόνα 5.30), όπως αναφέρθηκε και πριν ο χρήστης μπορεί να εισάγει όσες προσωπικές σημειώσεις επιθυμεί σε μία ημέρα, αν όμως η ημέρα αυτή είναι ημέρα πακέτου τότε οι σημειώσεις του χρήστη μπαίνουν από κάτω όπως σε αυτό το παράδειγμα. Το κουμπί Close κλείνει την φόρμα επισκόπησης ημέρας και επιστρέφει στο αντικείμενο Ημερολόγιο, από εκεί πατώντας στο μενού, με ονομασία Options γίνεται επιλογή της μοναδικής επιλογής Exit, όπου και τερματίζει το αντικείμενο Ημερολογίου και επιστρέφει στην Κεντρική φόρμα επιλογών (εικόνα 5.31).

συγκεκριμένου εταίρου καθώς και άλλες πληροφορίες που θα αναλυθούν στην συνέχεια. Το πάτημα της επιλογής Show On Map, ανοίγει το αντικείμενο istlMap (εικόνα 5.28). Στον χάρτη φαίνεται ο συγκεκριμένος εταίρος με χαρακτηριστική σήμανση για να διαφέρει από τους άλλους . Ο χρήστης μπορεί να πατήσει πάνω σε αυτόν τον εταίρο και να δει την αξιολόγηση που έκανε, το σημείωμα του καθώς και άλλες πληροφορίες που θα αναλυθούν στην συνέχεια (εικόνα 5.29).



Εικόνα 5.29 Επιλογές istlPopUp για επιλεγμένο εταίρο στον χάρτη.



Εικόνα 5.31 Τερματισμός του ημερολογίου από την επιλογή Exit του istlMenuItem.

Χρήση Του Χάρτη



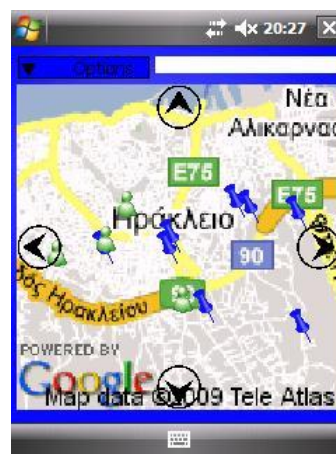
Εικόνα 5.31 Κεντρική φόρμα επιλογών.

Από το κεντρικό μενού το κουμπί GPS ενεργοποιεί χάρτη (εικόνα 5.31). Αυτό που θα παρατηρήσει ο χρήστης κατά την είσοδο του στο αντικείμενο του Χάρτη είναι ότι αντί να βλέπει την θέση του στον χάρτη, βλέπει χάρτη με κέντρο ένα νοητό τετράγωνο από τα τέσσερα σημεία που δημιουργούν οι τέσσερις πιο ακραίες θέσεις των εταιρών του πακέτου. Ενώ η γεωγραφική θέση του χρήστη φαίνεται πάνω στο επάνω μέρος.


Παρατηρούνται τα εξής (εικόνα 5.32).


1. Το μενού με ονομασία Options το οποίο προσφέρει επιλογές που θα εξεταστούν στην συνέχεια.
2. Τον κεντρικό υποδοχέα με τον χάρτη, καθώς και τα κουμπιά ολίσθησης χάρτη (βέλη που βρίσκονται στις άκρες της οθόνης).
3. Άξιο παρατήρησης είναι τα σημάδια (markers). Κατά την εκκίνηση τα σημάδια που προσφέρονται είναι μόνο των χρηστών και του E-kones (Μπλε χρώμα) (activities/Business Partner).


Όπως αναφέρθηκε και προηγούμενος το E-kones Experience προσφέρει 2 χρωματισμούς, ο κόκκινος για κινήσεις του χρήστη και ο μπλε για χαρακτηριστικά του E-kones.



Εικόνα 5.32 Χάρτης

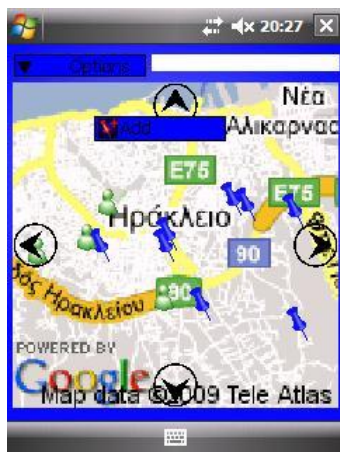
 Σήμανση (marker) E-kones. Η συγκεκριμένο σήμανση τοποθετείται από την εφαρμογή σε θέση στον χάρτη αντίστοιχη με αυτή που έχει δηλώσει ο εταίρος στην βάση του E-kones. Αντιπροσωπεύει τον συγκεκριμένο εταίρο χωρικά. Πρέπει να σημειωθεί ότι έχει δημιουργηθεί αλγόριθμος μετάφρασης της θέσης degrees σε θέση στην οθόνη, ο οποίος όμως έχει αρκετό ποσοστό σφάλματος.

 Σήμανση (marker). User Marker. Η σήμανση αυτή τοποθετείται από τον χρήστη σε θέση της αρεσκείας του είτε για να σημειώσει μια θέση μέσω του ημερολόγιου είτε μέσω του αντικειμένου χάρτη.

 Σήμανση (marker) Ekones Users Marker. Η συγκεκριμένη σήμανση τοποθετείται από την εφαρμογή στην θέση που έχουν ενημερώσει οι υπόλοιποι χρήστες που (είναι εγγεγραμμένοι στο ίδιο πακέτο) την υπηρεσία του E-kones ότι βρίσκονται. Για την ακρίβεια κατά την εκκίνηση κανένας χρήστης δεν πρέπει να

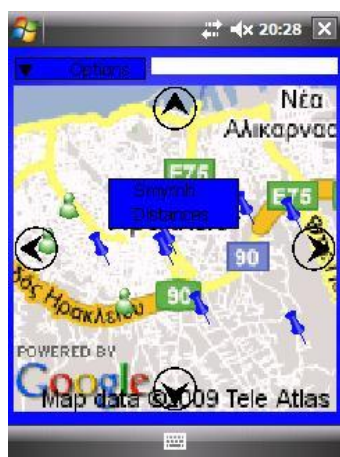
βρίσκεται στον χάρτη, αλλά να προστίθενται μετά κάθε φορά που ενημερώνεται η εφαρμογή.

Να σημειωθεί ότι για την ορθή λειτουργία αυτού του συστατικού απαιτείται υπηρεσία διακοσμητή καθώς και πρωτόκολλο λειτουργίας που δεν έχει υλοποιηθεί. Στο συγκεκριμένο παράδειγμα η θέση ενός χρήστη παρέχεται από διακοσμητή διαδικτύου, η οποία όμως είναι προκαθορισμένη και όχι σε πραγματικό χρόνο.



Εικόνα 5.33 istlPopUp ελεύθερης περιοχής.

Note (εικόνα 5.34). Η επιλογή move μετακινεί σήμανση σε μια νέα θέση στον χάρτη με απλό drag and drop. Η επιλογή Remove αφαιρεί τη σήμανση ενώ η επιλογή Note επιτρέπει στον χρήστη να προσθέσει προσωπική σημείωση.

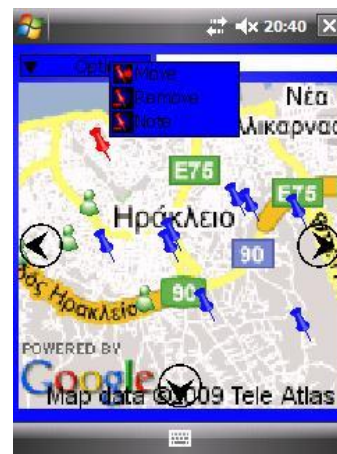


Εικόνα 5.35 istlPopUp εταιρίου.

Στην εικόνα 5.35 φαίνονται οι επιλογές που εμφανίζονται αν πατηθεί ένα E-kones Marker. Αυτές είναι οι Business Partner Name και Distances. Η πρώτη επιλογή μπορεί να πατηθεί και να παρουσιάσει πληροφορίες για τον συγκεκριμένο πάροχο. (εικόνα 5.36). Η επιλογή Distances εμφανίζει τις αποστάσεις σε

Οποιοδήποτε πάτημα πάνω σε μία ελεύθερη περιοχή του χάρτη ενεργοποιεί το istlPopUp προσφέροντας την επιλογή add (εικόνα 5.33) η οποία αν πατηθεί θα δημιουργηθεί ένα σημάδι χρήστη στην αντίστοιχη θέση που πατήθηκε (user Marker –κόκκινο χρώμα).

Αν πατηθεί η σήμανση που δημιουργήθηκε προηγουμένως εμφανίζεται ένα άλλο istlPopUp συγκεκριμένο για τα user markers προσφέροντας τις επιλογές move,remove και



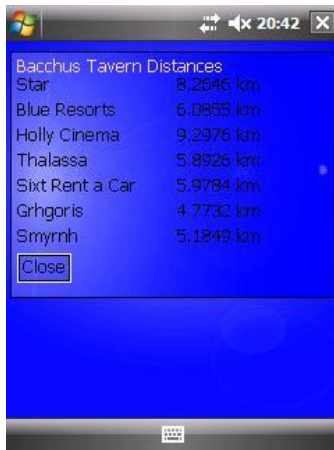
Εικόνα 5.34 istlPopUp σήμανσης χρήστη.

Να σημειωθεί ότι κάθε αντικείμενο του χάρτη προσφέρει istlPopUp με διαφορετικές επιλογές. Στο προηγούμενο παρουσιάστηκε istlPopUp με επιλογές για κενή περιοχή και user marker. Παρακάτω θα εξεταστούν οι επιλογές E-kones Users Markers, και E-kones Marker.



Εικόνα 5.36 Φόρμα πληροφοριών πάροχου.

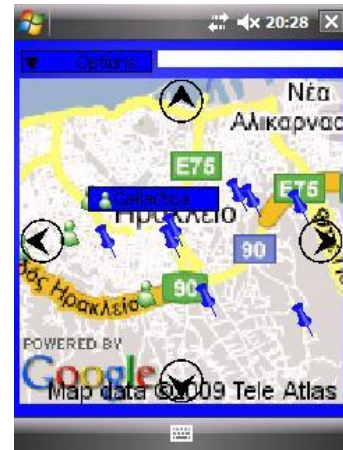
km μεταξύ του επιλεγμένου πάροχου και των υπολοίπων (εικόνα 5.37).



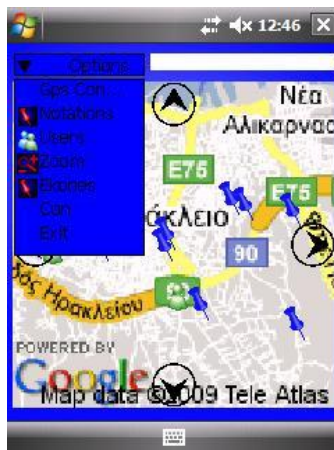
Εικόνα 5.37 Φόρμα πληροφοριών αποστάσεων μεταξύ πάροχων.

Η επιλογή ενός εικονιδίου χρήστη εμφανίζει το όνομα του συγκεκριμένου χρήστη (εικόνα 5.38).

Το τελευταίο συστατικό του αντικειμένου χάρτη είναι το `istlMenuItem` το οποίο προσφέρει πληθώρα επιλογών και επιπρόσθετα σε κάποιες επιλογές προσφέρει και δεύτερο επίπεδο επιλογών (εικόνα 5.39).



Εικόνα 5.38 `istlPopUp` άλλου χρήστη.

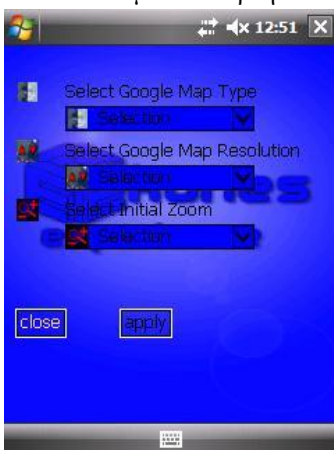


Εικόνα 5.39 `istlMenuItem` του αντικειμένου Χάρτη.

Οι επιλογές πρώτου επιπέδου είναι Gps Config, Notations, Users, Zoom, Ekones, Con, Exit. Θα εξεταστεί η κάθε μία ξεχωριστά.

Η επιλογή Con προσπαθεί να αναγκάσει τον χάρτη να εμφανιστεί σε περιπτώσεις που δεν συμβαίνει κατά την εκκίνηση. Η επιλογή Exit προκαλεί τον τερματισμό του αντικειμένου Χάρτη και επιστρέφει στην κεντρική φόρμα επιλογών.

Η επιλογή Gps Config ανοίγει την φόρμα παραμετροποίησης του χάρτη (εικόνα 5.40).



Εικόνα 5.40 Φόρμα παραμετροποίησης χάρτη.

Στο σημείο αυτό πρέπει να σημειωθεί ότι οι επιλογές δεν έχουν λειτουργικό αντίκρισμα στην τελική έκδοση της εφαρμογής καθώς αφαιρέθηκαν λόγω προβλημάτων συμβατότητας. Παρόλα αυτά παρουσιάζεται αυτό το παράθυρο κυρίως για την προβολή του αντικειμένου `istlComboBox`.



Εικόνα 5.41 Φόρμα παραμετροποίησης χάρτη. `istlComboBox` χάρτη.

Συνοπτικά το πρώτο combobox προσφέρει δυνατότητα επιλογής χάρτη σε Hybrid, Terrain, Satellite, Mobile που είναι όλοι οι δυνατοί χάρτες που προσφέρει η

υπηρεσία Static Google Maps (εικόνα 5.41).



Εικόνα 5.42 Φόρμα παραμετροποίησης χάρτη. istlComboBox ανάλυσης.

Το δεύτερο combo box προσφέρει δυνατότητα επιλογής ανάλυσης χάρτη από 64X64 μέχρι 240X240 (εικόνα 5.42). Το τρίτο combo box προσφέρει δυνατότητα επιλογής μεγέθυνσης χάρτη από 0 έως 17 (εικόνα 5.42). Αφού γίνουν οι απαραίτητες επιλογές το κουμπί apply τις ενεργοποιεί, ενώ το κουμπί close κλείνει την φόρμα παραμετροποίησης και επιστρέφει στο αντικείμενο χάρτη.



Εικόνα 5.43 Φόρμα παραμετροποίησης χάρτη. istlComboBox επίπεδου μεγέθυνσης.

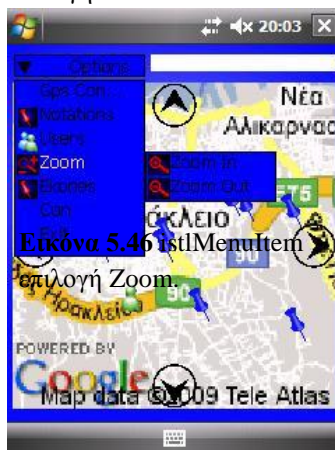


Εικόνα 5.44 istlMenuItem επιλογή Notations.

Η επιλογή Notations έχει δεύτερο επίπεδο το οποίο προσφέρει επιλογές παρόμοιες με το αντίστοιχο istlPopupMenu του User Marker (εικόνα 5.44). Η μόνο καινούρια επιλογή είναι η Hide με την οποία ο χρήστης μπορεί να αποκρύψει όλα τα σημάδια τύπου User Marker από την οθόνη.



Εικόνα 5.45 istlMenuItem επιλογή Users.



Εικόνα 5.46 istlMenuItem επιλογή Zoom.

Η επιλογή Users έχει δεύτερο επίπεδο το οποίο προσφέρει τις επιλογές Show All και Show Friends (εικόνα 5.45). Η επιλογή Zoom έχει δεύτερο επίπεδο το οποίο προσφέρει τις επιλογές Zoom In και Zoom Out (εικόνα 5.46). Η επιλογή Ekones έχει δεύτερο

επίπεδο το οποίο προσφέρει την επιλογή Hide Markers με την οποία μπορούμε να αποκρύψουμε όλα τα σημάδια τύπου E-kones Markers (εικόνα 5.47).



Εικόνα 5.47 istlMenuItem επιλογή Ekones.

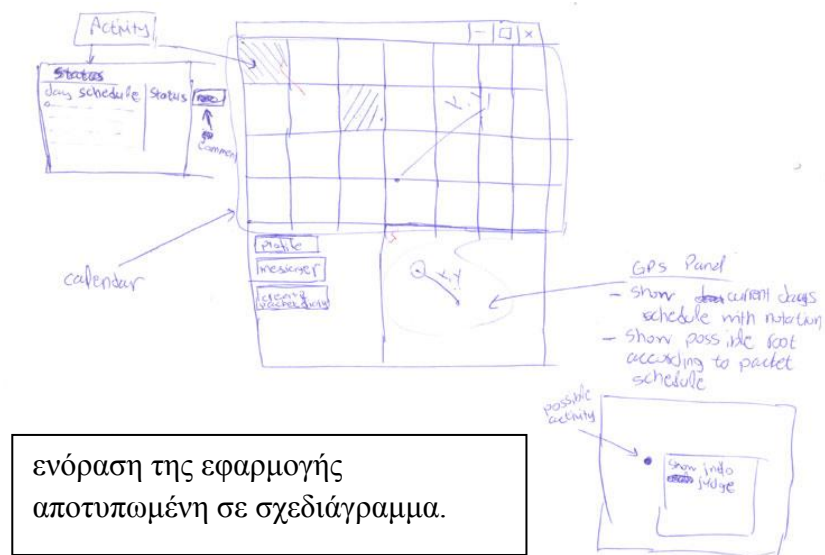
ΚΕΦΑΛΑΙΟ 6

Κύκλοι Ανάπτυξης και Προβλήματα.

Η ανάπτυξη της εφαρμογής e-kones experience διήρκησε 4 μήνες, και πέρασε από πολλά στάδια ανάπτυξης πριν την ολοκλήρωσή της. Σε πρώτη φάση κάθε αντικείμενο αναπτύχθηκε σαν πρωτότυπο και ανεξάρτητο από τα υπόλοιπα. Σε δεύτερη φάση συνδέθηκαν μεταξύ τους και κτίστηκε το κεντρικό μενού ενώ γινόνταν απαραίτητες διορθώσεις όπου ήταν αναγκαίο. Ταυτόχρονα με την σύνδεση των αντικειμένων δημιουργούνταν και οι απαραίτητες κλάσεις μεταφοράς δεδομένων μέσα στην εφαρμογή. Αφού είχε ολοκληρωθεί το δέσιμο των συστατικών δημιουργήθηκε ο XML Parser για προσπέλαση δεδομένων XML από την υπηρεσία καθώς και οι απαραίτητες κλάσεις προδιαγραφής δεδομένων. Στο τέλος αναπτύχθηκαν τα αντικείμενα διεπαφής.

Παράλληλα με την ανάπτυξη δημιουργήθηκαν και δευτερεύοντα εργαλεία δοκιμών και ελέγχου (XML Parser tool, Google Maps Connection Tool, GPS tool).

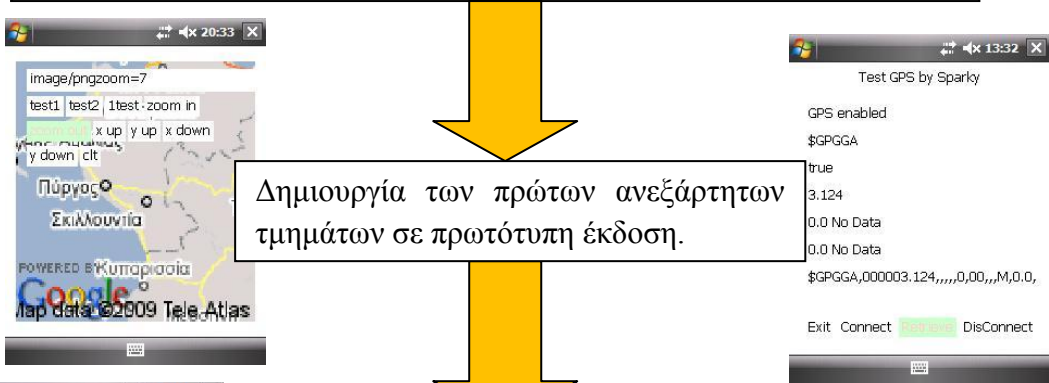
Κατά την διάρκεια την ανάπτυξης εμφανίστηκαν αρκετά προβλήματα κυρίως λόγω ασυμβατότητας μεταξύ (J9 – LWUIT), κάποια μπόρεσαν να λυθούν κάποια όμως εξακολουθούν να υφίστανται



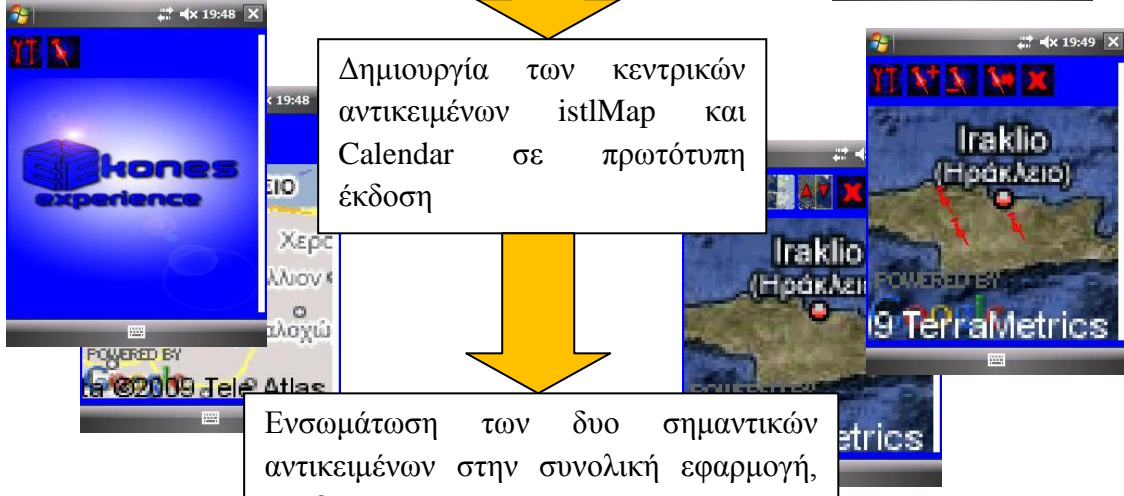
και στην τελική πρωτότυπη έκδοση. Αρκετά από αυτά ήδη παρουσιάστηκαν σε προηγούμενες σελίδες, τα κυριότερα που εξακολουθούν να υπάρχουν είναι

1. Άρνηση της συσκευής να φορτώσει χάρτες του google maps συγκεκριμένης ανάλυσης, επιπέδου μεγέθυνσης και τύπου.
2. Μικρά προβλήματα με τους αλγόριθμους μετάφρασης των γεωγραφικών συντεταγμένων σε συντεταγμένες οθόνης.
3. Γενικό πρόβλημα απόδοσης λόγω μεγάλου όγκου δεδομένων στην μνήμη που μεταφέρονται από φόρμα σε φόρμα.
4. Έλλειψη υποστήριξης του JSR-75, έτσι ώστε να μην μπορούν να δημιουργηθούν προσωρινά αρχεία για την εφαρμογή, με αποτέλεσμα το προηγούμενο.

Εκκίνηση πρώτου κύκλου ανάπτυξης, ενόραση της εφαρμογής, προσπάθεια δημιουργίας των πρώτων βασικών συστατικών.



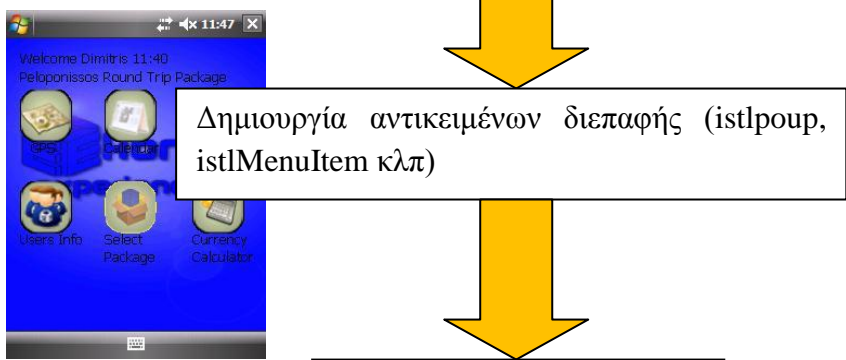
Δημιουργία των πρώτων ανεξάρτητων τμημάτων σε πρωτότυπη έκδοση.



Δημιουργία των κεντρικών αντικειμένων istlMap και Calendar σε πρωτότυπη έκδοση

Ενσωμάτωση των δυο σημαντικών αντικειμένων στην συνολική εφαρμογή, και δημιουργία του XML Parser

Δημιουργία Κεντρικού μενού εφαρμογής, δευτερεύοντες φόρμες, έλεγχοι και διορθώσεις όπου είναι αναγκαίο



Δημιουργία αντικειμένων διεπαφής (istlrouop, istlMenuItem κλπ)

Τελική πρωτότυπη έκδοση

Σύνοψη

Η εφαρμογή που δημιουργήθηκε αν και πρωτότυπη με σοβαρές ελλείψεις κατάφερε να διεισδύσει σε ένα βαθμό στην τεχνολογία που αφορά φορητές συσκευές και στο πως αυτές μπορούν να αλληλεπιδράσουν με τον χρήστη. Η ενασχόληση με την επιλεγμένη γλώσσα προγραμματισμού δείχνει τα πλεονεκτήματα που προσφέρει στον προγραμματιστή, παρόλα τα προβλήματα που προέκυψαν λόγω των περιορισμών της εικονικής μηχανής που χρησιμοποιήθηκε αλλά και των σοβαρών προβλημάτων συμβατότητας με την γραφική βιβλιοθήκη LWUIT.

Η εφαρμογή σε αυτή την πρωτότυπη έκδοση εκμεταλλεύτηκε στο μέγιστο επιτρεπτό βαθμό τις δυνατότητες της συσκευής, όπως μεγάλη οθόνη, οθόνη αφής, σύστημα GPS. Αντικείμενα όπως το `istlMap` προσφέρουν αλληλεπίδραση με την οθόνη αφής αφού ο χρήστης μπορεί να προσθέσει, να διαγράψει αλλά και να μεταφέρει σε όποιο σημείο της οθόνης θέλει δική του σήμανση. Αντικείμενα όπως τα `istlMenuItem`, `istlComboBox`, `istlPopUp`, παρουσιάζουν την ευκολία που προσφέρει η Java στην δημιουργία αντικειμένων χρήστη, με σκοπό δημιουργία διεπαφής πιο κοντά σε πρότυπα εφαρμογών για επιτραπέζιους υπολογιστές. Νέοι καινούργιοι τύποι αντικειμένων όπως το `istlRate` δίνουν ακόμα μεγαλύτερη ελευθερία στον προγραμματιστή και φέρνουν την έννοια της προσωποποίησης πιο κοντά στην πραγματικότητα.

Παρόλα αυτά απαιτείται να διανυθεί δρόμος μέχρι να μπορέσει η J2ME να φτάσει σε σημείο όπου ο προγραμματισμός εφαρμογών θα γίνεται με τον ίδιο απλό τρόπο όπως στους επιτραπέζιους υπολογιστές. Η ψαλίδα μεταξύ δωρεάν και εμπορικών εικονικών μηχανών όσον αφορά τα ποιοτικά χαρακτηριστικά εξακολουθεί να υπάρχει, παρά τις αξιόλογες προσπάθειες για δωρεάν εικονικές μηχανές υψηλών δυνατοτήτων, δεν φαντάζει να γίνεται στο πολύ κοντινό μέλλον. Ωστόσο αυτά το ενδιαφέρον της παγκόσμιας κοινότητας είναι πλέον στις φορητές συσκευές, εταιρίες όπως η Google, youtube αλλά και άλλες ήδη δημιουργούν και προσφέρουν δικές τους τεχνολογίες σε πολύ γρήγορο ρυθμό. Εφαρμογές τρίτων κατασκευαστών δίνουν περισσότερες δυνατότητες γι' αυτό που ονομάζεται προσωποποίηση συσκευής, ενώ και η ίδιες οι εταιρίες όπως η Nokia, Sony-Ericson ενσωματώνουν περισσότερα από τα χαρακτηριστικά της J2ME στις συσκευές τους.

ΠΑΡΑΡΤΗΜΑ

Βοηθητική Εφαρμογή XML Parser

Κατά την διάρκεια ανάπτυξης του αλγόριθμου προσπέλασης XML δεδομένων δημιουργήθηκε πρόγραμμα ελέγχου και αποσφαλμάτωσης του αλγόριθμου. Κατόπιν κτίστηκε η κάθε κλάση προδιαγραφής δεδομένων που σχετιζόταν με το αντίστοιχο XML ή τμήμα αυτού, ελέγχθηκε η σωστή προσπέλαση και ενσωματώθηκε στην εφαρμογή. Εδώ θα εξεταστεί συνοπτικά η εφαρμογή αυτή.

Κουμπι σύνδεσης με την διεύθυνση διαδικτύου που βρίσκεται το XML.

Συνολικός αριθμός γραμμών που έχουν προσπελαστεί.

Τρόπος υποβολής ερωτήματος (1 εκ των 4). Μόλις πατηθεί το κουμπι connect δημιουργείτε το συγκεκριμένο istlComboBox με περιεχόμενο μόνο τους κόμβους

Επόμενη/Προηγούμενη γραμμή δομής δεδομένων (Vector) που κρατάει το XML.

Τρέχων γραμμή

Πληροφορίες XML γραμμής.

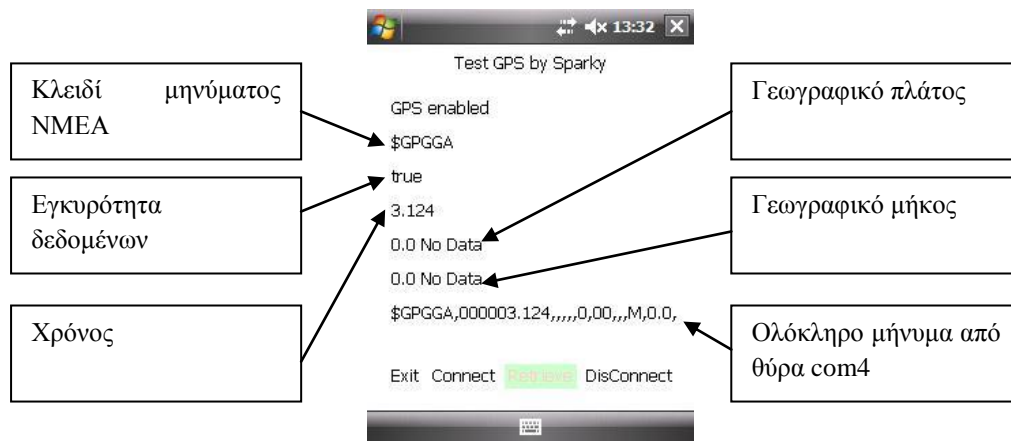
- Ετικέτα γραμμής
- Τύπος (Rot, Node, attribute)
- Περιεχόμενο (an attribute)
- Αναγνωριστικό κλειδί.
- Αναγνωριστικό κλειδί κόμβου
- Αναγνωριστικό κλειδί πατέρα.

Έχει επιλεγθεί ο κόμβος με nodeid 1, πλέον έχει αλλάξει ο συνολικός αριθμός των γραμμών σε 5, που είναι ο αντίστοιχος.

Lines	Line	Tag	Type	Content	Id	Node Id	Root Id
566	0	package	Root		1	0	0
5	0	year	attribute	2009	null	-1	1

Βοηθητική Εφαρμογή GPS

Η βοηθητική εφαρμογή GPS λειτουργεί αυτόνομα και εμφανίζει την γεωγραφική θέση του χρήστη (μήκος/ πλάτος) καθώς και άλλες πληροφορίες. Η εφαρμογή χρησιμοποιεί προσωρινή μνήμη (buffer) μεγέθους 512 bytes, για να διαβάσει δεδομένα από την θύρα com4. Κατόπιν ελέγχει για μήνυμα που αρχίζει από το κλειδί \$GPGGA και τερματίζει με τους χαρακτήρες διαφυγής <CR><LF>. Το τελικό βήμα είναι να κατακερματίσει το μήνυμα σε αντίστοιχα δεδομένα όπως γεωγραφικό μήκος, πλάτος, αριθμός δορυφόρων κλπ, και να γεμίσει με δεδομένα αντίστοιχη κλάση προδιαγραφής.



BIBΛΙΟΓΡΑΦΙΑ

What are KVM and CVM in J2ME?

<http://www.ucertify.com/article/what-are-kvm-and-cvm-in-j2me.html>

J2ME Core Concepts by Eric Giguère

<http://www.developer.com/java/j2me/article.php/1378971>

A Brief Look at Java 2 Micro Edition by Michael Nygard

<http://www.wirelessdevnet.com/channels/java/features/j2me.html>

Types of configurations in J2ME By [Ivan Lim](#)

<http://www.java-samples.com/showtutorial.php?tutorialid=440>

Articles by Eric Giguere

<http://www.ericgiguere.com/articles/index.html>

Dive into the J2ME configuration layer by [Puneet Gupta](#)

http://articles.techrepublic.com.com/5100-10878_11-1046647.html

J2ME: Introduction, Configurations and Profiles by Urs Steiner University of Zürich

http://www.google.gr/url?sa=t&source=web&ct=res&cd=1&url=http%3A%2F%2Fwww.ifi.uzh.ch%2F~riedl%2Flectures%2FJava2001-j2me.pdf&ei=yQdCSvDEG83J_ga_yq2dDQ&usg=AFQjCNGVbmLoYM9Kf0ToN2ML9Mu0puzICA&sig2=B-1MfU4H_iYmz1rVJTb2nQ

Think small with J2ME by Soma Ghosh

<http://www.ibm.com/developerworks/library/wi-j2me/>

The K virtual machine (KVM)

<http://java.sun.com/products/kvm/wp/KVMwp.pdf>

Sun Central Webpage with links to white papers and data sheets for the Java 2 Platform, Micro Edition (J2ME) and Java Card technologies

<http://java.sun.com/javame/overview/techpapers/index.jsp>

Java Community Process Website <http://jcp.org/en/home/index>

Introduction to the Java 2 Micro Edition (J2ME) Platform By Prentice Hall

<http://www.developer.com/ws/article.php/1475521>

J2ME Programming Models Technical Information Paper – v1.0 19 October 2004 by Kevin Horowitz

<http://www.google.gr/url?sa=t&source=web&ct=res&cd=4&url=ftp%3A%2F%2Fftp.software.ibm.com%2Fsoftware%2Fpervasive%2Finfo%2Fproducts%2Fwctme%2FJ2ME-Platform.pdf&ei=dA1CSoucDtaNsAbx2P3GCA&usg=AFQjCNGBuplYdu61Q85LJmtKpqcaml5jsg&sig2=Fbobi3i2qUFbN62nBHU2JA>

Overview of Java 2 Platform, Micro Edition (J2ME™) (Chapter 2)

<http://www.google.gr/url?sa=t&source=web&ct=res&cd=4&url=http%3A%2F%2Fwww.informit.com%2Fcontent%2Fimages%2F0321197984%2Fsamplechapter%2Friggsch02.pdf&ei=rAICSvKsJpGPsAbzyPDPCA&usq=AFQjCNFHH-oIHhf8EnO1sl2q9CjiAo61zw&sig2=U81A5DwQfGmbcEHglUEbDw>

Java In Small things Manning 2002

J2ME: The Complete Reference McGraw-Hill/Osborne 2003

Introduction to Mobile Devices

http://www.webopedia.com/quick_ref/mobile_OS.asp

Mobile operating system from Wikipedia

http://en.wikipedia.org/wiki/Mobile_platform

<http://www-01.ibm.com/software/wireless/weme/>

<http://www.nsicom.com/Default.aspx?tabid=138>

<http://www.handango.com/catalog/ProductDetails.jsp?storeId=2218&platformId=30&productId=48776>

<http://www.handango.com/catalog/ProductDetails.jsp?storeId=2218&productId=163028>

http://www2s.biglobe.ne.jp/~dat/java/project/jvm/index_en.html

<https://phoneme.dev.java.net/>

Mobile & Embedded Community on Java.net

http://java.sun.com/javame/reference/docs/mobile_embedded_community.pdf

SIRF NMEA Protocol Manual

http://www.usglobalsat.com/downloads/NMEA_commands.pdf

SIRF StartIII GPS Chipset

<http://www.sirf.com/products/GSC3LPPProductInsert.pdf>

NMEA Protocol

http://en.wikipedia.org/wiki/NMEA_0183

Google Static Maps API

<http://code.google.com/apis/maps/documentation/staticmaps/>

