



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων

Πτυχιακή Εργασία

**Δυναμική προσαρμογή περιεχομένου στις δυνατότητες τερματικής
συσκευής**

Εφαρμογή σε νοσοκομειακό περιβάλλον



Σπουδαστής: Αγκούτογλου Ρομπέρτο, Α.Μ. 2081
Επιβλέπων Καθηγητής: Παναγιωτάκης Σπύρος

Ηράκλειο, Μάιος 2010

Εισαγωγή:

Με τις εξελίξεις που είχαμε τα τελευταία χρόνια στο επίπεδο του Hardware των Η/Υ, αρχίζουμε και συνηθίζουμε να βλέπουμε την υπολογιστική δύναμη ενός Η/Υ 10ετίας, συρρικνωμένη σε μια συσκευή τύπου κινητού τηλεφώνου, μικρουπολογιστή με γραφίδα (PDA), ακόμα και ρολόι χειρός. Τελευταία, γίνεται μια προσπάθεια για ευκολία και αμεσότητα στην πρόσβαση στο διαδίκτυο, οπότε αυτές οι συσκευές είναι ότι καλύτερο θα μπορούσε να χρησιμοποιήσει κάποιος, ειδικά σε μέρη όπου δεν υπάρχουν άμεσα Η/Υ με δυνατότητα σύνδεσης στο διαδίκτυο. Κι αυτό εξαιτίας του γεγονότος ότι οι συσκευές αυτού του είδους, εκτός των δυνατοτήτων πρόσβασης στο διαδίκτυο, μπορούν να επεξεργαστούν, πλέον, HTML κώδικα, όπως και να χρησιμοποιούν εκδόσεις δημοφιλών πλοηγών (browsers) για κινητές συσκευές. Βέβαια, το μικρό μέγεθος απαιτεί θυσίες από πλευράς δυνατοτήτων εισόδου των πληροφοριών, παρόλα αυτά, είτε με γραφίδα, είτε με μίνι πληκτρολόγιο τύπου QWERTY, αυτό ξεπερνιέται. Παραμένει το θέμα όμως της σωστής απόδοσης της εξόδου της πληροφορίας, ανάλογα με τις δυνατότητες της κάθε συσκευής. Μικρές οθόνες, περιορισμένο εύρος χρωμάτων, διαφορετικά πρότυπα για απόδοση ιστοσελίδας, είναι τα τεχνολογικά εμπόδια που αντιμετωπίζει η διαδικτυακή κοινότητα προκειμένου να επιτρέψει στους χρήστες τέτοιων συσκευών, να αντλούν πληροφορίες απ' αυτήν.

Ο σκοπός της πτυχιακής αυτής, είναι να προτείνει ένα τρόπο σχετικά με το πώς θα μπορούσε ένας Server να αναγνωρίσει τον τύπο της τερματικής συσκευής ενός client, που ζητάει μια ιστοσελίδα, ώστε ανάλογα με τις δυνατότητές του client, να του αποστέλει το ανάλογο περιεχόμενο. Είναι το αποτέλεσμα μιας πολύπλευρης έρευνας, η οποία, προσπαθώντας να εγγυηθεί την ευρύτερη δυνατή απήχηση, προτείνει ένα νέο μοντέλο πολύπλευρης δράσης και παρέμβασης, που θέτει τους άξονες μιας μελλοντικής πορείας και ενός νέου μοντέλου ανάπτυξης. Ανεξαρτήτως της εκάστοτε προσωπικής προσεγγίσεως, η ανάπτυξη της ενδογενούς διαδικτυακής ανάδρασης, συμπαρασύρει και ενισχύει δράσεις και διαθέσεις για πραγματική και ενεργό συμμετοχή όλων των χρηστών, ανεξαρτήτως της συσκευής που κατέχουν, στο διαδικτυακό γίγνεσθαι.

Οργάνωση της εργασίας:

Η παρούσα εργασία, αποτελείται από την εισαγωγή και τρία μέρη:

Το πρώτο μέρος, είναι μια θεωρητική εισαγωγή στις μεθόδους και τις τεχνολογίες που υπεισέρχονται στην αντιμετώπιση ζητημάτων προσαρμογής περιεχομένου και οι οποίες χρησιμοποιήθηκαν, εν τέλει, στη μελέτη και την υλοποίηση της λύσης που προτείνεται στην παρούσα εργασία.

Στο δεύτερο μέρος, περιγράφεται μια υποθετική υλοποίηση σε νοσοκομειακό περιβάλλον. Συγκεκριμένα, πρόκειται για μια εφαρμογή δυναμικής παρουσίασης στοιχείων ασθενών στο ιατρικό προσωπικό, ανάλογα με τις δυνατότητες της συσκευής που έχει ο πιστοποιημένος χρήστης, ο οποίος ζητάει πρόσβαση σ' αυτήν την πληροφορία.

Το τρίτο μέρος, παρουσιάζει τα συμπεράσματά μας από την προσέγγιση που υιοθετήσαμε, κάνει επισκόπηση των στόχων που επιτεύχθηκαν, και αναφέρει περαιτέρω παραδείγματα χρήσης και πιθανές μελλοντικές επεκτάσεις.

Περιεχόμενα

Μέρος 1^ο – Διαδικτυακές τεχνολογίες

Κεφάλαιο 1^ο – Γενικές πληροφορίες

1.1:Αρχιτεκτονική Server – Client: Μια σύντομη ανασκόπηση	
1.1.1 Ιστορική αναδρομή _____	σελ.9
1.1.2 Αντιστοίχιση στο διαδίκτυο _____	σελ.9
1.1.3 Τρόποι επικοινωνίας Client- Server _____	σελ.10
1.1.4 Το πρωτόκολλο HTTP _____	σελ.10
1.2:Glassfish server	
1.2.1:Βασικά στοιχεία _____	σελ.12
1.2.2:Λειτουργίες _____	σελ.12
1.3:Java Servlets	
1.3.1:Τι είναι ένας servlet _____	σελ.12
1.3.1:Πως λειτουργεί ένας servlet _____	σελ.13
1.3.1:Γιατί servlets; _____	σελ.14
1.4:Προσωπικές συσκευές με δυνατότητα ανάγνωσης ιστοσελίδων	
1.4.1:Η ανάγκη της αμεσότητας _____	σελ.15
1.4.2:Δυνατότητες των τωρινών συσκευών _____	σελ.16

Κεφάλαιο 2^ο – User-agents και WURFL

2.1:Τι είναι τα user-agents	
2.1.1:Βασικές πληροφορίες _____	σελ.18
2.2:Χρήσεις των user agents	
2.2.1:Εισαγωγικά στοιχεία _____	σελ.23
2.2.1:Η πρόταση CC/PP _____	σελ.23
2.3:WURFL	
2.3.1:Εισαγωγή στο WURFL _____	σελ.26
2.3.2:Οργάνωση και δομή του WURFL αρχείου _____	σελ.27
2.4:Πως χρησιμοποιούμε το WURFL	
2.4.1:Parsing _____	σελ.30
2.4.2:Άντληση στοιχείων _____	σελ.30
2.5:Συνδυασμός WURFL με τους user agents	
2.4.1:WURFL αντί UAProf _____	σελ.31
2.4.1: Δυναμική ανίχνευση του προφίλ UAProf ανάλογα με τον τύπο user agent της συσκευής-πελάτη _____	σελ.31
2.4.1:Παράδειγμα _____	σελ.31

Κεφάλαιο 3^ο – Δομικά στοιχεία μιας Ιστοσελίδας

3.1:HTML: ιστορική αναδρομή	
-----------------------------	--

3.1.1:ENQUIRE – Μια αρχή, με πολλές υποσχέσεις _____	σελ.34
3.1.2:Η ιδέα του υπερκειμένου _____	σελ.35
3.1.3:HTML, η γέννηση _____	σελ.35
3.2:Εκδόσεις της HTML	
3.2.1:HTML _____	σελ.36
3.2.2:ΧHTML _____	σελ.36
3.2.3:ΧHTML - MP _____	σελ.36
3.2.4:Δυναμική ΧHTML _____	σελ.37
3.3:Μορφή της HTML	
3.3.1:Τρόπος συγγραφής του κώδικα _____	σελ.37
3.3.2:Δόμηση του κώδικα _____	σελ.38
3.4:Τι είναι η XML	
3.4.1:Ανάγκη για ελεύθερη οργάνωση στοιχείων _____	σελ.39
3.4.2:Παράδειγμα _____	σελ.39
3.4.2:Δυνατότητες _____	σελ.
3.5:Τι είναι το XSLT	
3.5.1:στα χνάρια της XML _____	σελ.40
3.5.2:Παράδειγμα _____	σελ.41

Μέρος 2ο – Η εφαρμογή MedApp

Κεφάλαιο 4ο – Εισαγωγή

4.1:Εικονικό σενάριο	
4.1.1:HealthGuy General Hospital _____	σελ.45
4.1.2:Λειτουργία του νοσοκομείου _____	σελ.45
4.2:Ανάγκες	
4.2.1:Ιατροί _____	σελ.45
4.2.2:Διοικητικοί _____	σελ.46
4.3: Ταξινόμηση τερματικού εξοπλισμού	
4.3.1:H/Y, laptops, netbooks _____	σελ.46
4.3.2: PDAs και κονσόλες ψυχαγωγίας με δυνατότητα πρόσβασης στο διαδίκτυο _____	σελ.46
4.3.3:Συσκευές κινητής τηλεφωνίας _____	σελ.47
4.3.4:Server με συνεχή δυνατότητα επεξεργασίας αιτήσεων _____	σελ.47
4.4:Προτυποποίηση	
4.4.1:Κατάταξη των συσκευών σε κλάσεις δυνατοτήτων _____	σελ.47
4.4.1.1: Κλάση H/Y και laptop _____	σελ.48
4.4.1.2: Κλάση συσκευών τύπου PDA _____	σελ.48
4.4.1.3: Κλάση κινητών συσκευών με πολυμεσικές δυνατότητες ____	σελ.49
4.4.1.4: Κλάση κινητών συσκευών με περιορισμένες πολυμεσικές δυνατότητες _____	σελ.50
4.4.1.5: Κλάση κινητών συσκευών χωρίς πολυμεσικές δυνατό- τητες και με περιορισμούς στην προβολή των ιστοσελίδων ____	σελ.50

4.4.2:Προσαρμογή περιεχομένου, ανάλογα την κλάση κάθε τερματικής συσκευής.	51
4.4.2.1: Μορφή του περιεχομένου της ιστοσελίδας για τερματικές συσκευές πελάτη τύπου Η/Υ (desktop, laptop)	σελ.51
4.4.2.2: Μορφή του περιεχομένου της ιστοσελίδας για τερματικές συσκευές πελάτη τύπου PDA	σελ.52
4.4.2.3: Μορφή του περιεχομένου της ιστοσελίδας για κινητές συσκευές πελάτη με πολυμεσικές δυνατότητες	σελ.53
4.4.2.4:Μορφή του περιεχομένου της ιστοσελίδας για κινητές συσκευές πελάτη με περιορισμένες πολυμεσικές δυνατότητες	σελ.53
4.4.2.5:Μορφή του περιεχομένου της ιστοσελίδας για κινητές συσκευές πελάτη χωρίς πολυμεσικές δυνατότητες και με περιορισμούς στην προβολή ιστοσελίδων	σελ.54
4.4.3:Μορφή της ιστοσελίδας υποδοχής	σελ.54

Κεφάλαιο 5^ο – Εφαρμογή

5.1:Ζητούμενη παραγόμενη ιστοσελίδα

5.1.1:Γενικά	σελ.55
5.1.2:Γιατί το XML, XSLT	σελ.55

5.2:XML ασθενών

5.2.1:Οργάνωση	σελ.56
5.2.2:Κώδικας του XML	σελ.57

5.3:Εισαγωγική ιστοσελίδα

5.3.1:Κώδικας	σελ.59
5.3.2:Παρατηρήσεις	σελ.60
5.3.3:Επεξηγήσεις	σελ.60

5.4:XSLT για συσκευές τύπου Η/Υ

5.4.1:Κώδικας	σελ.61
5.4.2:Παρατηρήσεις	σελ.62

5.5:XSLT για συσκευές τύπου PDA

5.5.1:Κώδικας	σελ.63
5.5.2:Παρατηρήσεις	σελ.65

5.6:XSLT για κινητά με δυνατότητες πολυμέσων

5.6.1:Κώδικας	σελ.65
5.6.2:Παρατηρήσεις	σελ.67

5.7:XSLT για κινητά με περιορισμένες δυνατότητες πολυμέσων

5.7.1:Κώδικας	σελ.67
5.7.2:Παρατηρήσεις	σελ.69

5.8:XSLT για απλά κινητά

5.8.1:Κώδικας	σελ.69
5.8.2:Παρατηρήσεις	σελ.70

5.9:XSLT για αιματολογικό πίνακα

5.9.1:Κώδικας	σελ.70
5.9.2:Παρατηρήσεις	σελ.72

5.10:Java Servlet: Medservlet

5.10.1:Κώδικας	σελ.73
5.10.2:Τα βασικά της λειτουργίας του servlet	σελ.77
5.10.3:Το αντικείμενο out	σελ.78
5.10.4:Το αντικείμενο request	σελ.79
5.10.5:Το αντικείμενο response	σελ.81
5.10.6:Μέθοδοι μεταγλώττισης XML, ανάλογα με τις δυνατότητες της τερματικής συσκευής	σελ.81

5.11:WURFL Parser

5.11.1:Κώδικας	σελ.82
5.11.2:Βασική λειτουργία	σελ.86
5.11.3:Κοινοποιημένες μεταβλητές	σελ.86

Κεφάλαιο 6^ο – Αποτελέσματα

6.1:Παραγόμενες ιστοσελίδες

6.1.1:Εξομοίωση	σελ.88
6.1.2:Καταχώρηση των user agents	σελ.88

6.2:Επίτευξη ζητούμενου στόχου

6.2.1:Ιστοσελίδες για Η/Υ	σελ.92
6.2.2: Ιστοσελίδες για PDA	σελ.93
6.2.3: Ιστοσελίδες για κινητά τηλέφωνα με πολυμεσικές δυνατότητες	σελ.93
6.2.4: Ιστοσελίδες για κινητά τηλέφωνα με περιορισμένες πολυμεσικές δυνατότητες	σελ.94
6.2.5:Ιστοσελίδες για κινητά τηλέφωνα χωρίς ή με πολύ περιορισμένες πολυμεσικές δυνατότητες	σελ.95
6.2.6:Αιματολογικός πίνακας ασθενή	σελ.96

6.3:Βελτιώσεις

6.3.1:Τι μπορεί να γίνει έπειτα	σελ.97
6.3.2:Εισαγωγή νέων στοιχείων	σελ.98

Μέρος 3^ο - Σύνοψη

Κεφάλαιο 7^ο – Προβλήματα

7.1:Ελλείψεις του WURFL

7.1.1:Το αιώνιο πρόβλημα της δωρεάν, ανοιχτής κοινότητας	σελ.100
7.1.2:Capabilities	σελ.100
7.1.3:Patch	σελ.101

7.2:User agents και κακή χρήση αυτών

7.2.1: Robots, web crawlers, downloading tools, linkers, web filterers και spammers	σελ.101
7.2.2:Παρόμοιες κινητές συσκευές	σελ.101
7.2.3:Έλλειψη ασφάλειας	σελ.102
7.2.4:Συμπέρασμα	σελ.102

Κεφάλαιο 8^ο – Περαιτέρω παραδείγματα

8.1:Υπηρεσίες

8.1.1:Μετακινήσεις _____ σελ.103

8.1.2:Εξυπηρέτηση του πολίτη _____ σελ.103

8.1.3:Κοινοποίηση πολιτογραφικών στοιχείων _____ σελ.103

8.2:Ψυχαγωγία

8.2.1:Αθροιστικές προσομοιώσεις _____ σελ.104

8.2.2:«Κυνήγι θυσανρού» και λοιπά αστικά παιχνίδια _____ σελ.105

8.2.3:Προσωπικά παιχνίδια _____ σελ.106

8.3:Μηχανογράφηση _____ σελ.106

Παράρτημα

Βιβλιογραφία _____ σελ.108

ΜΕΡΟΣ 1^ο

ΔΙΑΔΙΚΤΥΑΚΕΣ ΤΕΧΝΟΛΟΓΙΕΣ

- Κεφάλαιο 1^ο: Γενικές πληροφορίες
- Κεφάλαιο 2^ο: User agents και WURFL
- Κεφάλαιο 3^ο: Δομικά στοιχεία μιας ιστοσελίδας

Κεφάλαιο 1^ο – Γενικές πληροφορίες

1.1: Αρχιτεκτονική Client - Server : Μια σύντομη ανασκόπηση

1.1.1: Ιστορική αναδρομή

Πολλά χρόνια πριν φτάσουμε στο σημείο που βρισκόμαστε σήμερα όσον αφορά την εξέλιξη της πληροφορικής, το να χρησιμοποιήσει κάποιος έναν Η/Υ δεν ήταν απλή υπόθεση. Οι υπολογιστές αυτοί, συνήθως είχαν το μέγεθος ενός μικρού κτηρίου, περιείχαν εκατοντάδες λυχνίες που χρέζανε αντικατάστασης συχνά – πυκνά, καταναλώναν μεγάλα ποσά ηλεκτρικής ισχύος και οι εργασίες που περαίωναν ήταν συγκεκριμένες και μη επεκτάσιμες.

Μια μεγάλη επανάσταση ήταν, ο προγραμματισμός των εργασιών που θα πραγματοποιούσε ο Η/Υ, χάρη στην αρχιτεκτονική του Φον Νιούμαν. Πλέον, ο χρήστης μπορούσε να εισάγει τις δικές του ακολουθίες υπολογισμού, τον λεγόμενο «κώδικα», και ο Η/Υ να επεξεργαστεί τα δεδομένα με τον τρόπο που πραγματικά επιθυμούσε ο χρήστης, χωρίς να χρειάζεται να ανακατασκευαστούν από την αρχή τα μηχανικά και τα ηλεκτρικά του στοιχεία.

Σιγά-σιγά, παρατηρώντας ότι θα μπορούσαν να διαμοιράσουν την μεγάλη (για τότε!) επεξεργαστική ισχύ ενός Η/Υ, ώστε να εξυπηρετούνται ταυτόχρονα περισσότεροι του ενός χρήστες, οι κατασκευαστές προτείνανε και υλοποίησαν μοντέλα Η/Υ, που δέχονταν δεδομένα από τερματικά και επιστρέφανε σ' αυτά τα αποτελέσματα.

Τα τερματικά αυτά, τα λεγόμενα «κουτά τερματικά» (dummy terminals), ήταν στην ουσία μια οθόνη και ένα πληκτρολόγιο. Η αντίστοιχη κεντρική μονάδα που έχουν σήμερα οι προσωπικοί υπολογιστές, ήταν ο υπερυπολογιστής (mainframe). Κάθε τερματικό ζητούσε επεξεργαστική ισχύ από το mainframe, στέλνοντας του τα δεδομένα, κι αυτό αφού τα επεξεργάζονταν, γυρνούσε το αποτέλεσμα.

Το μοντέλο αυτό ονομάστηκε «πελάτης - εξυπηρετητής» (Client – Server), που το τερματικό, ο «πελάτης», ζητούσε να εξυπηρετηθεί με υπολογιστική ισχύ από τον κεντρικό Η/Υ, τον «εξυπηρετητή».

1.1.2: Αντιστοίχιση στο διαδίκτυο

Η κληρονομιά του μοντέλου αυτού, έρχεται σήμερα στη μορφή που λειτουργεί η προβολή ιστοσελίδων του διαδικτύου, και όχι μόνον. Ένας κεντρικός υπολογιστής, όχι απαραίτητα υπερυπολογιστής, που ονομάζεται Server, είναι μονίμως συνδεδεμένος στο διαδίκτυο και συνήθως, είναι υπεύθυνος να δίνει το υλικό ιστοσελίδων, σε όποιον τις ζητάει. Όταν τώρα, ένας απλός χρήστης που κατέχει έναν προσωπικό Η/Υ

με δυνατότητα σύνδεσης στο διαδίκτυο, θέλει να λάβει μια ιστοσελίδα από τον «εξυπηρετητή», ζητάει στην ουσία σαν «πελάτης» να του εμφανιστεί η ιστοσελίδα στο πρόγραμμα πλοήγησής του (browser). Τώρα πια, οι προσωπικοί Η/Υ, έχουν μεγάλες δυνατότητες, πιο μεγάλες ακόμα και από τα τεράστια μηχανήματα του '50 και '60 (και '70), οπότε συνήθως, ο εξυπηρετητής αποστέλλει στον πελάτη τον κώδικα της ιστοσελίδας, για να τον επεξεργαστεί τοπικά ο προσωπικός Η/Υ και ο πλοηγός να αποδώσει το τελικό αποτέλεσμα, χωρίς την μεσολάβηση του εξυπηρετητή. Ο εξυπηρετητής από την πλευρά του, πέραν από τον φόρτο εργασίας που έχει, να «σερβίρει» το υλικό του σε εκατομμύρια χρήστες που το ζητούν καθημερινά, και συνήθως ταυτόχρονα, πραγματοποιεί και κάποιες μικροεργασίες, όπως έλεγχος πιστοποίησης (authentication) και ασφάλειας, διορθώσεις κ.α.

1.1.3: Τρόποι επικοινωνίας Client - Server

Ο πελάτης και ο εξυπηρετητής, για να επιτελέσουν τις εργασίες από κοινού, πρέπει να θέσουν ένα κοινό κανόνα επικοινωνίας μεταξύ τους, που θα αφορά τη μετάδοση των δεδομένων, την σημασιολογία και τον τύπο των δεδομένων, καθώς και τις παραμέτρους για την εφαρμογή και προβολή των πληροφοριών που θα παρουσιαστούν στον τελικό χρήστη.

Σ' αυτή τη περίπτωση, δράση αναλαμβάνουν τα λεγόμενα πρωτόκολλα επικοινωνίας, τα οποία ποικίλουν ανάλογα με την εργασία που θέλουμε να επιτελέσουν. Τα πιο γνωστά, διαδικτυακά πρωτόκολλα, που χρησιμοποιούνται στο διαδίκτυο σήμερα, είναι τα SMTP, SNTP, SSH, DNS, POP, DHCP, FTP, TCP/IP, ARP και HTTP.

Στο τελευταίο, ιδιαίτερη μνεία γίνεται στην παρούσα εργασία, καθώς θα διαχειριστούμε αιτήματα μέσω αυτού του πρωτοκόλλου, για την επίτευξη του στόχου μας.

1.1.4: Το πρωτόκολλο HTTP

Το πρωτόκολλο μεταφοράς κειμένου ιστοσελίδων (HyperText Transfer Protocol - HTTP), βρίσκεται στο 7^ο επίπεδο της διαστρωμάτωσης OSI, της γενικής αρχιτεκτονικής επικοινωνίας δύο ηλεκτρονικών συσκευών μέσω δικτύου. Βασίζεται στην αποστολή αιτήσεων, από πλευράς πελάτη και απαντήσεων, από πλευράς εξυπηρετητή. Οι HTTP αιτήσεις που καταθέτει ο πελάτης πηγάζουν από κατάλληλη εφαρμογή που μπορεί να ονομαστεί και user agent και αποτελεί αναπόσπαστο κομμάτι των διάφορων πλοηγών. Οι απαντήσεις δίνονται από τον εξυπηρετητή που μπορεί να ονομαστεί και origin server.

Ο πελάτης, με την αίτηση του, ζητάει από τον εξυπηρετητή πρόσβαση στην πληροφορία που αυτός περιέχει, προσδιορίζοντάς την μέσω των URIs (Uniform Resource Identifiers). Τα URIs στέλνονται υπό τη

μορφή ροής χαρακτήρων, και μεταφράζονται στον εξυπηρετητή, όπου αναγνωρίζει το τι ζητάει ο πελάτης, βάση του καθιερωμένου συντακτικού, κανόνων κι εντολών, που περιλαμβάνει το πρωτόκολλο HTTP, και τα οποία περιγράφονται στο RFC 1945 [21]. Πιο συγκεκριμένα, τα HTTP μηνύματα που μπορεί να στείλει ένας πελάτης σε έναν εξυπηρετητή είναι:

- GET – Ζητάει ο πελάτης, πρόσβαση σε κάποιο συγκεκριμένο υλικό που διαθέτει αποθηκευμένο τοπικά ο εξυπηρετητής. Για παράδειγμα, η μορφή μιας HTTP αίτησης προς το URI www.servername.gr/images/logo.gif, παίρνει τη μορφή:

```
GET /images/logo.gif HTTP/1.1  
Host: www.servername.gr
```

όπου www.servername.gr είναι το συμφωνημένο όνομα του εξυπηρετητή, [/images/logo.gif](http://www.servername.gr/images/logo.gif) είναι το μονοπάτι στο δίσκο του εξυπηρετητή προς την πληροφορία που αιτείται ο client, και HTTP/1.1 είναι η έκδοση του πρωτοκόλλου HTTP που αναγνωρίζει ο πελάτης.

- HEAD – Ίδια με την εντολή GET, αλλά ζητάει μόνο την επικεφαλίδα του πακέτου πληροφορίας, για χρήση σε παραγωγή πληροφοριών για την πληροφορία, τα γνωστά MetaData.
- POST – ο πελάτης υποβάλλει πληροφορίες που δίνει συνήθως ο χρήστης, για επεξεργασία και χρήση από τον εξυπηρετητή.
- DELETE – Διαγράφει από τον εξυπηρετητή το υλικό που αναφέρουμε στην αίτηση. Ο εξυπηρετητής πρέπει να είναι προσεχτικός, στο ποιος αιτείται τη διαγραφή και τι σκοπούς επιτελεί η πράξη αυτή.

Ο εξυπηρετητής, ανάλογα με το τι ζητάει ο πελάτης, επιστρέφει στον πλοηγό του πελάτη, ο οποίος αποτελεί την εφαρμογή προβολής της πληροφορίας, τις αιτηθείσες πληροφορίες που μπορούν να περιέχουν και πολυμεσικό περιεχόμενο [5]. Ο Η/Υ του πελάτη, πρέπει να είναι σε θέση να αναπαράγει το πολυμεσικό υλικό, αλλιώς υποβαθμίζεται η ποιότητα της μεταδιδόμενης πληροφορίας.

Το φαινόμενο αυτό, δηλαδή να αποστέλλεται βαριά πολυμεσική πληροφορία, σε πελάτες με Η/Υ χαμηλών δυνατοτήτων, είναι αρκετά συνηθισμένο, αφού το HTTP ως πρωτόκολλο δεν περιλαμβάνει αντίστοιχους μηχανισμούς αποτροπής, αφήνοντας στους προγραμματιστές των διαδικτυακών εφαρμογών την ανάλογη πρόνοια.

Οι τελευταίοι, συνηθισμένοι να απευθύνονται σε πελάτες με τερματικό εξοπλισμό desktop ή laptop, με ισχυρή επεξεργαστική ισχύ δηλαδή, συνήθως αμελούν τέτοιου είδους διαχωρισμό των πελατών. Τελευταία, όμως, με την έλευση και ραγδαία εξάπλωση των κινητών συσκευών που έχουν πρόσβαση στο διαδίκτυο, το φαινόμενο της λανθασμένης προβολής περιεχομένου είναι πολύ συνηθισμένο. Στην παρούσα εργασία επιχειρούμε να εμβαθύνουμε σε ένα τρόπο αντιμετώπισης αυτού του προβλήματος.

1.2:GlassFish Server:

1.2.1: Βασικά στοιχεία

Η Sun Microsystems, στις 4 Μαΐου του 2006 έφτιαξε τον ομώνυμο web Server με χρήση της γλώσσας προγραμματισμού Java. Ο κώδικας είναι ανοιχτός, και προστατεύεται από δύο άδειες, την CDDL [29] και την GPL [28]. Στη δεύτερη περιλαμβάνεται το classpath exception, που σημαίνει ότι μπορούμε να συνδυάσουμε την εργασία μας με την εφαρμογή αυτή, χωρίς νομικούς περιορισμούς δικαιωμάτων όσον αφορά το εκτελέσιμο αυτής.

Ο GlassFish Server δίνεται δωρεάν για μεταφόρτωση από την κεντρική του ιστοσελίδα (www.glassfish.org). Επιπλέον, στις προαιρετικές επιλογές της εφαρμογής περιέχεται το επίσης δωρεάν περιβάλλον εργασίας NetBeans, ένα περιβάλλον συγγραφής κι ανάπτυξης σε γλώσσα Java.

1.2.2: Λειτουργίες

Ο Server αυτός, μπορεί να διαχειρίζεται HTTP αιτήσεις από απομακρυσμένους διαδικτυακούς χρήστες, όπως και να μεταγλωττίζει JSP σελίδες σε HTML, και να χειρίζεται νέους τύπους ιστοσελίδων και πληροφοριών, όπως ανακοινώνονται από το W3C, ενώ αναλαμβάνει και τις διεργασίες ασφαλείας, όπου και όποτε απαιτούνται.

Παράλληλα, μπορεί να διαχειριστεί ειδικές λειτουργίες της γλώσσας Java όσον αφορά τη δυναμική παράδοση και παρουσίαση μιας ιστοσελίδας στον πελάτη. Εμπεριέχει και λειτουργίες από το δημοφιλή Server Tomcat, σχετικές με την δυναμική διαχείριση των HTTP αιτήσεων.

1.3:Java Servlets:

1.3.1: Τι είναι ένας servlet

Ένας servlet είναι μία κλάση της γλώσσας Java που μπορεί να διαχειρίζεται αιτήσεις HTTP, μιας και συμμορφώνεται στο γενικό πρωτόκολλο Java Servlet API.

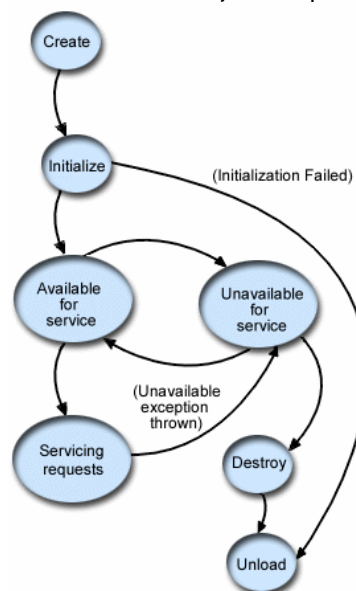
Παρόμοιες τακτικές ακολουθούνται και σε διαφορετικές τεχνολογίες, όπως η ASP.NET και γενικότερα η CGI (Common Gateway Interface) [1]. Η τελευταία αναφέρεται στη διεπαφή επικοινωνίας πελάτη-εξυπηρετητή για την ανταλλαγή δεδομένων μέσω HTTP και την παραγωγή από αυτά μιας ιστοσελίδας στον πλοηγό του πελάτη. Ένας servlet χρησιμοποιείται σε συνδυασμό με την γλώσσα Java, όπως και ιστοσελίδες τύπου JSP, για δυναμική διαχείριση περιεχομένου, τόσο από πλευράς εξυπηρετητή, όσο και από πλευράς πελάτη.

Δύο είναι οι βιβλιοθήκες που περιέχουν τις μεθόδους για την χρήση του servlet, η javax.servlet.HTTP και η javax.servlet γενικότερα. Στο NetBeans, τα πρόσθετα που δίνονται στην έκδοση J2EE, περιέχουν αυτές τις βιβλιοθήκες, όπως και εργαλεία δημιουργίας και διαχείρισης ολόκληρης της διαδικτυακής εφαρμογής που υλοποιείται με τον Server Glassfish.

1.3.2: Πώς λειτουργεί ένας servlet

Ένας servlet για να δράσει, πρέπει να δημιουργηθεί ένα instance αυτού. Το Σχήμα 1, μας δίνει μια ιδέα του κύκλου ζωής ενός servlet. Αν δεν υπάρχει instance του servlet, ο εξυπηρετητής:

1. Φορτώνει την κλάση του servlet.
2. Δημιουργεί ένα instance αυτού (create).
3. Ξεκινά τον servlet εκτελώντας την init μέθοδο (initialize).



Σχήμα 1. Σχηματικό διάγραμμα ροής ενός Servlet.

Έπειτα, καλεί τη μέθοδο service (available for service), δηλώνοντας παράλληλα τα αντικείμενα request και response. Τα αντικείμενα αυτά, είναι αιτήσεις του πρωτοκόλλου HTTP.

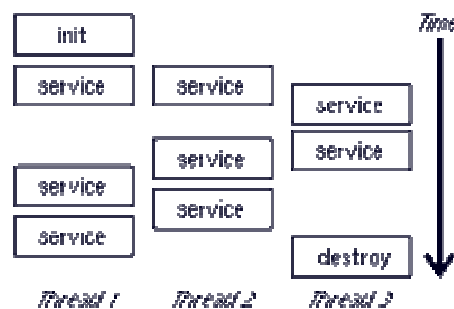
Το αντικείμενο request περιέχει την αίτηση του πελάτη προς τον εξυπηρετητή, ενώ το αντικείμενο response δημιουργείται από τον servlet ώστε να σταλεί ως απάντηση από τον εξυπηρετητή προς τον πελάτη.

Ένας servlet μένει στην μνήμη για όση ώρα χρειαστεί να επεξεργαστεί όλες τις αιτήσεις που διαχειρίζεται ο εξυπηρετητής. Όσο ο εξυπηρετητής διαχειρίζεται αιτήσεις από τους πελάτες (servicing requests), το instance παραμένει και διαχειρίζεται τις λειτουργίες που είναι προγραμματισμένο να κάνει.

Αν για κάποιο λόγο θέλουμε να τερματίσουμε τη λειτουργία του servlet ή δεν μπορεί ο servlet να επεξεργάζεται αιτήματα του HTTP πρωτοκόλλου πια (unavailable for service), ο εξυπηρετητής πρώτα σιγουρεύει ότι δεν υπάρχουν άλλες αιτήσεις προς επεξεργασία, και ακολούθως τερματίζει τη λειτουργία του servlet με χρήση της μεθόδου destroy. Ο garbage collector καθαρίζει την μνήμη από τις όποιες αναφορές προς τον servlet, και η διαδικασία μπορεί να επαναεκκινηθεί κατά βούληση.

1.3.3: Γιατί servlets;

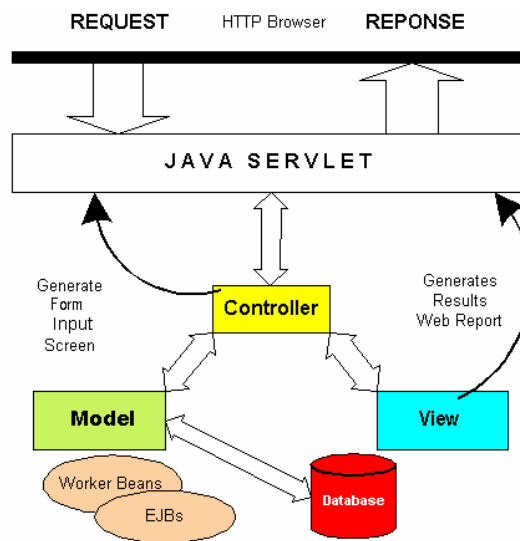
Οι servlets, σε συνδυασμό με τις ιστοσελίδες τύπου JSP, έχουν αμεσότητα και πιο γρήγορη επεξεργασία αιτήσεων από τα παραδοσιακά CGI. Ενώ οι υπόλοιπες τεχνολογίες διαθέτουν μια ουρά εργασιών, η φιλοσοφία των servlets, είναι ότι, μπορεί να διαθέσει πολλά και παράλληλα νήματα επεξεργασίας αιτήσεων από ένα instance μόνον, οπότε και να διαχειρίζεται πολλά και διαφορετικά αιτήματα τύπου GET και POST. Παράλληλα, λόγω του ότι ένας servlet δουλεύει στο ίδιο νήμα επεξεργασίας με την αίτηση, και δεν χρησιμοποιεί ένα δικό του, γλυτώνουμε επεξεργαστική ισχύ, μιας και δεν χρειάζονται επιπλέον πόροι (σχήμα 2).



Σχήμα 2. Η παράλληλη επεξεργασία νημάτων ενός servlet

Μεταξύ των αιτήσεων, ο εξυπηρετητής διατηρεί τον servlet στη μνήμη. Σε σχέση με τις παραδοσιακές τεχνολογίες CGI, όπου το πρόγραμμα φορτώνεται πάντα εκ νέου, έχουμε κέρδος στους πόρους του συστήματος και στον χρόνο επεξεργασίας ενός αιτήματος. Επειδή μάλιστα, ένα instance διαχειρίζεται όλες τις αιτήσεις, έχουμε περαιτέρω

οικονομία στη μνήμη. Επιπλέον, τα δεδομένα που μένουν στον εξυπηρετητή, συνεχίζει να τα επεξεργάζεται ο servlet μιας και παραμένει στη μνήμη.



Σχήμα 3. Οι αιτήσεις του πρωτοκόλλου HTTP πρέπει να περάσουν πρώτα από ένα servlet, για πρόσβαση στους φυσικούς πόρους ενός server.

Ως προς την ασφάλεια, ο servlet εκτελείται σε δικό του, εικονικό χώρο, που διαθέτει αποκλειστικά δικούς του πόρους. Αν ο servlet είναι π.χ. ιός, θα βλάψει μόνο τους συγκεκριμένους πόρους του συστήματος, χωρίς να θέσει σε κίνδυνο τη λειτουργία του εξυπηρετητή (σχήμα 3).

1.4: Προσωπικές συσκευές με δυνατότητα ανάγνωσης ιστοσελίδων:

1.4.1: Η ανάγκη της αμεσότητας

Όταν στα μέσα της δεκαετίας του '90, το διαδίκτυο έγινε δημοφιλές, κυρίως χάρη στην αποτελεσματική αναζήτηση κι εύρεση πληροφοριών και στις υπηρεσίες ηλεκτρονικού ταχυδρομείου που προσέφερε, παρουσιάστηκε επιτακτική η ανάγκη να γίνει μια πιο άμεση, πιο εύκολη και χωρίς εμπόδια τεχνολογία. Απώτερος σκοπός ήταν η προσφορά άπλετης πρόσβασης στο διαδίκτυο σε όλους αυτούς τους χρήστες που ζητούσαν να απολαμβάνουν τις διαδικτυακές υπηρεσίες, οπουδήποτε βρισκόντουσαν. Η τεχνολογία των ασύρματων τοπικών δικτύων είχε δρόμο ακόμα μέχρι να φτάσει σε ικανοποιητικό βαθμό να προσφέρει υπηρεσίες διαδικτύου, οπότε οι ερευνητές στράφηκαν σε μια πολλά υποσχόμενη τεχνολογία, αυτή των δικτύων κινητής τηλεφωνίας.

Καθώς όλο και περισσότερος κόσμος αποκτούσε κινητό τηλέφωνο, ήταν φανερό ότι θα φτάναμε σε σύντομο χρονικό διάστημα στην εποχή

που ο καθένας μας θα είχε τουλάχιστον μια συσκευή με πρόσβαση στην κυψελωτή τηλεφωνία. Η αμεσότητα που δίνει το κινητό τηλέφωνο στην επικοινωνία ήταν ιδανική στο να στραφούν οι ερευνητές στην αναζήτηση μιας τεχνολογίας διαδικτύου γι' αυτές τις συσκευές. Πράγματι, περίπου το 1998, το WAP [30] ήταν πραγματικότητα.

Εντούτοις, οι προσδοκίες σχετικά γρήγορα διαψεύστηκαν. Η αποτυχία του εγχειρήματος αυτού εντοπίζεται στο ότι το πρωτόκολλο αυτό ήταν «κλειστό» και «ξένο» από τη σφαίρα του διαδικτύου. Οι ιστοχώροι που έπρεπε να δημιουργηθούν εκ νέου, η πρόσβαση στην πληροφορία που ήταν προβληματική, αφενός λόγω των περιορισμένων δυνατοτήτων των συσκευών (μικρές οθόνες, έλλειψη πλήκτρων ή γραφίδων, αδύναμοι επεξεργαστές) και αφετέρου λόγω των τότε δικτύων GPRS (κακή πιστοποίηση συσκευών, προβληματικός έλεγχος λαθών μετάδοσης δεδομένων), η δική του γλώσσα σύνταξης ιστοσελίδων (WML) που ήταν ασύμβατη με την HTML, και οι υπερβολικές χρεώσεις για τις WAP υπηρεσίες υπήρξαν τροχοπέδη για την εξάπλωση της τεχνολογίας αυτής.

Παράλληλα, ο παγκόσμιος ιστός εξαπλώνονταν. Οι ερευνητές συνειδητοποίησαν γρήγορα πως αν ήθελαν να δώσουν μια ουσιαστική λύση στο πρόβλημα, θα έπρεπε να εναρμονιστούν στις ανάγκες των καιρών. Με την εμπειρία του προηγούμενου εγχειρήματος, οι συσκευές εξελίχθηκαν ώστε να μπορούν να λαμβάνουν και να προβάλλουν ιστοσελίδες του ιστοχώρου, άμεσα από τον παγκόσμιο ιστό, χωρίς διαμεσολαβητές και ξεχωριστά πρωτόκολλα, όπου μπορούσαν φυσικά. Σήμερα, βλέπουμε πως η κατεύθυνση αυτή ήταν πετυχημένη, αφού κινητές και προσωπικές συσκευές θεωρείται πλέον συνηθισμένο να έχουν πρόσβαση στο διαδίκτυο.

1.4.2: Δυνατότητες των τωρινών συσκευών

Οι εξελίξεις στην τεχνολογία των μικροπολογιστών, μας έδωσε δυνατούς επεξεργαστές σε λιλιπούτεια μεγέθη, ικανούς να διαχειρίζονται το υλικό του παγκόσμιου ιστού. Πλακέτες ηλεκτρονικών κυκλωμάτων ενός κινητού τηλεφώνου περιέχουν πια τα στοιχεία ενός Η/Υ 10ετίας, με καλύτερη διαχείριση της ενέργειας και των πόρων, μιας και τα ολοκληρωμένα κυκλώματα σήμερα είναι καλύτερα οργανωμένα και πιο πλήρη σε λειτουργίες προγραμματισμού, μεταφοράς δεδομένων και φωνής. Ρόλο σε αυτό διαδραμάτισαν και οι νέες κωδικοποιήσεις και ασύρματες τεχνολογίες μετάδοσης δεδομένων.

Παράλληλα, οι δεικτικές (pointing) τεχνολογίες γίνονται πιο προσιτές, οπότε το μεγάλο πρόβλημα εισαγωγής δεδομένων στα κινητά και PDA, λύνεται εύκολα με μια γραφίδα. Οθόνες αφής προσπαθούν να επιβληθούν στις καθιερωμένες δυνατότητες τέτοιων συσκευών, όπως και

οι αλγόριθμοι αναγνώρισης γραφής και ομιλίας στο προγραμματιστικό μέρος των συσκευών αυτών.

Λόγω του ότι σήμερα, ο περισσότερος κόσμος ζητάει πρόσβαση στον παγκόσμιο ιστό, όχι μόνο για τις κλασικές διαδικτυακές υπηρεσίες (web και email), αλλά και για πρόσβαση σε οπτικοακουστικό υλικό (multimedia), συμμετοχή σε κοινωνικές υπηρεσίες (social networking) αλλά και διαδικτυακά παιχνίδια, οι κινητές συσκευές οφείλουν να παρέχουν πλέον και τη δυνατότητα πρόσβασης, παρουσίασης κι επεξεργασίας διαδικτυακών πολυμεσικών εφαρμογών. Τα καινούργια μοντέλα τηλεφώνων και PDAs διαφημίζονται περισσότερο για τις πολυμεσικές τους δυνατότητες, παρά για την χρηστικότητά τους. Καθώς οι τάσεις του αγοραστικού κοινού αλλάζουν, το ίδιο συμβαίνει και στις τερματικές συσκευές.

Κεφάλαιο 2^ο – User agents και WURFL

2.1: Τι είναι οι user-agents:

2.1.1: Βασικές πληροφορίες

Με τη χρήση της αρχιτεκτονικής πελάτη - εξυπηρετητή, πληροφορίες πρέπει να δίνονται στον εξυπηρετητή, σχετικά με το τι ζητάει ο πελάτης, καθώς και το ποιος είναι. Για το λόγο αυτό, μέσα σε όλο το πακέτο πληροφοριών που αποστέλλει ο πελάτης στον εξυπηρετητή, μεταξύ άλλων περιέχεται ενημέρωση για τα τεχνικά χαρακτηριστικά της user-agent εφαρμογής του πελάτη, από την οποία εκκινούν τα αιτήματα προς τον εξυπηρετητή. Γενικά, οι user-agents είναι εφαρμογές που αναλαμβάνουν το ρόλο του πελάτη και υλοποιούν τα πρωτόκολλα επικοινωνίας μεταξύ πελάτη - εξυπηρετητή. Στην περίπτωση της πλοήγησης στο διαδίκτυο, τέτοιες εφαρμογές συμπεριλαμβάνονται συνήθως στους διάφορους πλοηγούς (desktop και κινητούς) και υλοποιούν την HTTP επικοινωνία από την πλευρά του πελάτη [1]. Στην παράγραφο 1.1.4 παρουσιάσαμε τα διαφορετικά μηνύματα που μπορεί να στείλει ένας HTTP πελάτης σε έναν εξυπηρετητή προκειμένου να αιτηθεί την εκπλήρωση συγκεκριμένων δράσεων (ανάγνωση, επανεγγραφή, διαγραφή) στα δεδομένα που ορίζονται μέσω των URIs. Επιπρόσθετα, το πρωτόκολλο HTTP [7] ορίζει ότι μέσω των μηνυμάτων αυτών κι άλλη πληροφορία μπορεί να μεταφερθεί από τον πελάτη στον εξυπηρετητή, μέσω των πρόσθετων επικεφαλίδων που μπορούν να συνοδεύουν κάθε αίτημα του πελάτη. Ακλουθεί λίστα με τις πιο συνηθισμένες επικεφαλίδες που συναντούμε εντός των μηνυμάτων του HTTP:

1. **Accept:** Η επικεφαλίδα αυτή ενημερώνει τον εξυπηρετητή, για το ποιους τύπους αρχείων και μορφοποιήσεων κειμένου, μπορεί να δεχτεί ο user-agent του πελάτη. Μπορεί να είναι της μορφής:

```
application/vnd.wap.wmlscriptc, text/vnd.wap.wml,
application/vnd.wap.xhtml+xml, application/xhtml+xml,
text/html, multipart/mixed, */*
```

Στην ουσία, είναι μια λίστα από τις υποστηριζόμενες μορφές πολυμέσων και οργάνωσης μιας ιστοσελίδας. Για κινητές συσκευές, μπορεί να συνδυάζει οποιαδήποτε από τις παρακάτω αναφορές:

Τύπος αρχείου	MIME τύποι πολυμέσων
Audio 3GPP files (.3gp)	audio/3gpp
Audio AMR files (.amr)	audio/amr

Τύπος αρχείου	MIME τύποι πολυμέσων
Audio AMR (wideband) files (.awb)	audio/amr-wb
Audio MIDI files (.mid or .midi)	audio/midi
Audio MP3 files (.mp3)	audio/mpeg
Audio MP4 files (.mp4)	audio/mp4
Audio WAV files (.wav)	audio/wav audio/x-wav
HTML files (.html or .htm)	text/html
Image BMP files (.bmp)	image/bmp image/x-bmp
Image GIF files (.gif)	image/gif
Image JPEG files (.jpg or .jpeg)	image/jpeg
Image PNG files (.png)	image/png
Image TIFF files (.tif or .tiff)	image/tiff
Image WBMP (Wireless BMP) files (.wbmp)	image/vnd.wap.wbmp
Java application JAR files (.jar)	application/java application/java-archive application/x-java-archive
Java application JAD files (.jad)	text/vnd.sun.j2me.app-descriptor
Plain text files (.txt)	text/plain
Symbian application SIS files (.sis)	application/vnd.symbian.install
Video 3GPP files (.3gp)	video/3gpp
Video MP4 files (.mp4)	video/mp4
WML files (compiled) (.wmlc)	application/vnd.wap.wmlc
WML files (plain text) (.wml)	text/vnd.wap.wml
WMLScript files (compiled) (.wmlsc)	application/vnd.wap.wmlscriptc
WMLScript files (plain text) (.wmls)	text/vnd.wap.wmlscript
XHTML MP files (.xhtml, .html or .htm)	application/vnd.wap.xhtml+xml application/xhtml+xml text/html

Το MIME (Multipurpose Internet Mail Extensions) [17], είναι ένα διαδικτυακό πρότυπο που επεκτείνει τη διαμόρφωση των μηνυμάτων του ηλεκτρονικού ταχυδρομείου, ως προς τους χαρακτήρες γραφής, το θέμα και την περιγραφή του μηνύματος, τα επισυναπτόμενα στοιχεία κ.α. Πέρα, όμως, από τις εφαρμογές ηλεκτρονικού ταχυδρομείου, για τις οποίες δημιουργήθηκε αρχικά, αυτό το πρότυπο περιγράφει, γενικά, τη μορφή που έχει κάθε πληροφορία, πολυμεσική ή μη, που ανταλλάσσεται μέσω δικτύου, όπως είναι, π.χ. μια ιστοσελίδα στο κινητό που την ζητάει. Σε γενικές γραμμές, μπορεί να πει κανείς ότι οι MIME περιγραφές περιεχομένου αποτελούν πλέον αναπόσπαστη παράμετρο των μηνυμάτων των περισσότερων διαδικτυακών πρωτοκόλλων του επιπέδου εφαρμογής (π.χ., HTTP, SIP, SMTP).

2. **User agent:** Κάθε HTTP user-agent στέλνει εντός κάθε αιτήματος προς έναν εξυπηρετητή πληροφορία, με τη μορφή κειμένου, που δηλώνει συνήθως τα τεχνικά χαρακτηριστικά της εφαρμογής (του πλοηγού δηλαδή) εντός της οποίας εκτελείται ο user agent, όπως όνομα και έκδοση του πλοηγού και του κατασκευαστή, γλώσσα επικοινωνίας, λειτουργικό σύστημα, διαθέσιμοι πόροι εφαρμογής, κ.α. Τελευταία, με την πρόσβαση των κινητών συσκευών στο διαδίκτυο, γίνεται προσπάθεια από τους διάφορους αρμόδιους οργανισμούς προτυποποίησης [21] [23] [24] [25] ώστε ένας user-agent να μπορεί, παράλληλα, να αποστέλλει και πληροφορίες για τον τύπο της τερματικής συσκευής, τα μηχανικά χαρακτηριστικά της, το λογισμικό της, κ.α. Περισσότερες πληροφορίες για την προσπάθεια αυτή δίνονται στην παράγραφο 2.2.

Σύμφωνα με την παράγραφο 10.15 του RFC 1945 [21], η σύνταξη του πεδίου user-agent πρέπει να είναι:

```
"User-Agent" ":" 1*( product | comment )
```

Ένα παράδειγμα μπορεί να δοθεί ως εξής:

```
User-Agent: CERN-LineMode/2.15 libwww/2.17b3
```

Ο user-agent στο παράδειγμα αυτό, μας πληροφορεί πως είναι εφαρμογή από το CERN, ονόματι Libwww έκδοση 2.17b3. Άλλο ένα, πιο ενδεικτικό παράδειγμα, είναι το εξής:

```
Mozilla/4.0 (compatible; MSIE 5.0; Windows ME) Opera 5.11 [en]
```

Ο user agent μας πληροφορεί πως ο πλοηγός που χρησιμοποιεί ο πελάτης, είναι ο Opera 5.11, Αγγλική έκδοση, με υποστηριζόμενες προτυποποιήσεις της Mozilla και του MS Internet Explorer, όπως και τις εκδόσεις των Windows ME.

3. **Accept-charset:** Η επικεφαλίδα αυτή, ενημερώνει τον εξυπηρετητή για την προτεινόμενη, από πλευράς πελάτη, μορφοποίηση χαρακτήρων. Αν ο πελάτης επιθυμεί πρότυπα που συνδιάζουν ειδικούς χαρακτήρες ή ξένες προς τα Αγγλικά γλώσσες, πρέπει να το ανακοινώσει στον εξυπηρετητή με αυτήν την επικεφαλίδα. Ένα παράδειγμα, για να ζητήσουμε κωδικοποίηση iso-8859-5, είναι το εξής:

```
Accept-Charset: iso-8859-5
```

4. **Accept-language:** Η επικεφαλίδα αυτή, ενημερώνει τον εξυπηρετητή, σχετικά με τις γλώσσες γραφής και ομιλίας που αναγνωρίζει ο πελάτης, ώστε ο εξυπηρετητής να απαντάει κάνοντας χρήση κάποιας από αυτών. Ένα παράδειγμα, ώστε να μας εμφανίζονται χαρακτήρες και ιστοσελίδες στην Ελληνική γλώσσα, είναι το εξής:

```
Accept-Language: gr
```

5. **Accept-encoding:** Η επικεφαλίδα αυτή, ενημερώνει τον εξυπηρετητή σχετικά με τις κωδικοποιήσεις δεδομένων που αναγνωρίζει ο πελάτης. Ένα παράδειγμα, που δηλώνει πως τα δεδομένα θα πρέπει να αποστέλλονται συμπιεσμένα κατά το πρότυπο συμπίεσης gzip, είναι το εξής:

```
Accept-Encoding: compress, gzip
```

6. **Cookie:** Με την επικεφαλίδα αυτή ο πελάτης διαβάζει τα δεδομένα από το κατάλληλο αρχείο κειμένου (web cookie) που έχει τοποθετήσει ο εξυπηρετητής σ' αυτόν, σε προηγούμενη συνεδρία και τα κοινοποιεί στον εξυπηρετητή. Ένα παράδειγμα, για ένα cookie που βρίσκεται αποθηκευμένο και περιέχει το παρακάτω κείμενο:

```
UserID=JohnDoe; Max-Age=3600; Version=1
```

είναι το εξής:

```
Cookie: $Version=1; Skin=new;
```

7. **Cache-control:** Η επικεφαλίδα αυτή οδηγεί σε συμφωνία για την κοινή τακτική, που θα ακολουθήσουν πελάτης και εξυπηρετητής, σχετικά με την προσωρινή αποθήκευση των δεδομένων κατά την αλληλουχία αιτήσεων-απαντήσεων. Ένα παράδειγμα, για να μην κρατιούνται δεδομένα σε καμιά πλευρά, είναι:

```
Cache-Control: no-cache
```

8. **Connection:** Η επικεφαλίδα αυτή, ενημερώνει τον εξυπηρετητή σχετικά με τον τύπο της επικοινωνίας που επιθυμεί ο πελάτης. Ένα παράδειγμα, για να σταματήσει η επικοινωνία μιας συνεδρίας μεταξύ πελάτη-εξυπηρετητή, είναι:

```
Connection: close
```

9. **Host:** Την επικεφαλίδα αυτή την είδαμε και όταν αναφερθήκαμε στα μηνύματα του HTTP. Η επικεφαλίδα αυτή περιέχει το όνομα του εξυπηρετητή προορισμού, γενικότερα το domain name του. Ένα παράδειγμα είναι το εξής:

```
Host: developershome.com
```

10. **X-wap-profile:** Η επικεφαλίδα αυτή περιέχει συνήθως μια αναφορά, με την οποία ο πελάτης επιχειρεί να ενημερώσει τον εξυπηρετητή σχετικά με το προφίλ δυνατοτήτων της συσκευής του. Συναντάται ιδιαίτερα στα κινητά τηλέφωνα, καθώς και σε υβριδικές συσκευές κινητού τηλεφώνου - PDA. Ένα παράδειγμα είναι:

```
x-wap-profile: HTTP://nds1.nds.nokia.com/uaprof/N6230ir200.xml
```

Αυτή η αναφορά, για παράδειγμα, οδηγεί σε ένα XML αρχείο με το προφίλ (περιγραφή) των δυνατοτήτων του κινητού τηλεφώνου Nokia 6230i. Με τον τρόπο αυτόν, η τελική αναγνώριση των δυνατοτήτων του πελάτη ανατίθεται στον εξυπηρετητή, με την ανάγνωση του αρχείου XML από τον διαδικτυακό τόπο που προσδιορίζεται στο πεδίο αυτό.

11. **Profile:** Για την αποστολή πληροφοριών σχετικών με τις δυνατότητες της συσκευής του πελάτη μπορεί να χρησιμοποιηθεί η προαιρετική αυτή επικεφαλίδα ως εξής:

```
Profile: <http://example.com/pl>, </profiles/other-profile>
```

Η επικεφαλίδα αυτή, λειτουργεί όπως και η ετικέτα <head> στην γλώσσα HTML[21]. Μπορεί να είναι ένα αναγνωριστικό, μοναδικό για την συσκευή, ή να δίνει την ανακατεύθυνση για τις επιπλέον πληροφορίες που σχετίζονται με τη συσκευή.

12. **Options:** Η προαιρετική αυτή επικεφαλίδα μεταφέρει επιπλέον πληροφορίες σχετικές με τον πελάτη, χωρίς να μεσολαβούν, υποχρεωτικά, ανακατευθύνσεις σε άλλα αρχεία. Χρησιμοποιείται, συνήθως, για τον καθορισμό του περιεχομένου που θα ανταλλάξουν εξυπηρετητής – πελάτης, όπως και την ανακοίνωση των δυνατοτήτων του πελάτη στον εξυπηρετητή. Μπορεί να έχει την παρακάτω μορφή:

```
Options: portrait.php?plantid%3D100
```

2.2:Χρήσεις των user-agents:

2.2.1:Εισαγωγικά στοιχεία

Ένας εξυπηρετητής, λαμβάνοντας μια HTTP αίτηση από κάποιον user-agent, αν διαθέτει ένα κατάλληλο CGI, μπορεί να αξιοποιήσει τις πληροφορίες που περιέχονται στις επικεφαλίδες του HTTP μηνύματος προς όφελος του πελάτη, δηλαδή, για παράδειγμα, για να ρυθμίσει το περιεχόμενο της ιστοσελίδας ώστε να εμφανίζεται σωστά στον πλοηγό του πελάτη, να πιστοποιήσει τον πελάτη, να περιορίσει την πρόσβαση ενός μη πιστοποιημένου πελάτη στο αποθηκευμένο υλικό που δεν τον αφορά ή που δεν θα μπορεί να διαχειριστεί, κ.α.

Ειδικά ως προς τη σωστή απόδοση μιας ιστοσελίδας, ανάλογα με τις δυνατότητες της τερματικής συσκευής του πελάτη, η προδιαγραφή που έχει οριστεί είναι τα επονομαζόμενα UAProfs ή User Agent Profiles [1]. Τα UAProfs αποτελούν λεπτομερή προφίλ με επεξηγήσεις σε μορφή κειμένου σχετικά με τις τεχνικές δυνατότητες (hardware και software) μιας ασύρματης ή κινητής συσκευής. Ο μηχανισμός λειτουργίας κι ανταλλαγής των UAProfs περιγράφεται στο πρότυπο CC/PP (Composite Capabilities / Preferences Profiles) [26], που αποτελεί πρόταση της W3C [22] για την περιγραφή των δυνατοτήτων μιας κινητής συσκευής.

2.2.2:Η πρόταση CC/PP

Το πρότυπο CC/PP χρησιμοποιεί την προδιαγραφή RDF (Resource Definition Framework) [1], για τη λεπτομερή καταγραφή μιας σειράς χαρακτηριστικών, που προσδιορίστηκαν από την W3C, με σκοπό την ανακοίνωση και γνωστοποίηση των δυνατοτήτων μιας κινητής συσκευής στον εξυπηρετητή. Η πρόταση αυτή αναφέρεται στα προαναφερθέντα user agents προφίλ, τα οποία δεν είναι κάτι παραπάνω από αρχεία XML

(περισσότερα για την XML θα αναφέρουμε στο 3^ο κεφάλαιο) που μέσω της συγκεκριμένης οργανωτικής τους δομής περιγράφουν τις τεχνικές δυνατότητες και τα χαρακτηριστικά μιας τερματικής συσκευής. Αυτά τα αρχεία, συνήθως βρίσκονται αποθηκευμένα σε ένα Server που λέγεται Profile Repository, του οποίου δουλειά είναι να αντιστοιχεί κάθε αίτημα για ανάκτηση ενός συγκεκριμένο προφίλ στο κατάλληλο αρχείο.

Η μορφή και οργάνωση αυτών των αρχείων, μπορεί να είναι η εξής:

```
<?xml version="1.0"?>
<!-- Any RDF graph is an RDF element
-->
<rdf:RDF xmlns:rdf="HTTP://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="HTTP://www.example.com/schema#">

  <!-- [Subject-resource] -propertyName-> [Object-resource]
  -->
  <rdf:Description
    rdf:about="HTTP://www.example.com/profile#Subject-resource">
    <propertyName>
      <rdf:Description
        rdf:about="HTTP://www.example.com/profile#Object-resource"
      />
    </propertyName>
  </rdf:Description>

  .....
  .....
  .....
```

Ακολούθως παρατίθενται πληροφορίες σχετικές με το περιεχόμενο αυτών των αρχείων, πώς περιγράφουν τις δυνατότητες των συσκευών, τι τύπους στοιχείων περιέχουν και σε τι σειρά:

1. Hardware Platform: Παρέχει πληροφορίες για τις δυνατότητες της τερματικής συσκευής σχετικά με το πλήθος των χρωμάτων, την εμφάνιση κειμένου, τον τύπο της συσκευής εισόδου, το μέγεθος της οθόνης κ.α.
2. Software Platform: Περιέχει πληροφορίες σχετικές με το λογισμικό της συσκευής, όπως για παράδειγμα αν υποστηρίζει MIME μορφοποιήσεις, την πλατφόρμα της Java που φιλοξενεί, τους πολυμεσικούς αποκωδικοποιητές κ.α
3. Network Characteristics: Περιέχει πληροφορίες για την υποστήριξη των δικτύων που προσφέρει η συσκευή, καθώς και για τα πρωτόκολλα επικοινωνίας και τις κωδικοποιήσεις ασφάλειας που χρησιμοποιεί.
4. Browser UA: Δίνει πληροφορίες για τον τύπο του πλοηγού που χρησιμοποιεί η συσκευή.
5. WAP Characteristics: Δίνει πληροφορίες για το αν η συσκευή είναι ικανή να χειριστεί την οικογένεια των WAP πρωτοκόλλων,

που συναντώνται στα κινητά, όπως τα DRM, WML, WAP κ.α. [26]

6. Push Characteristics: Δίνει πληροφορίες σχετικά με τις δυνατότητες που έχει η συσκευή όσον αφορά το πρωτόκολλο WAP push.
7. MMS Characteristics: Δίνει πληροφορίες για το κατά πόσο μπορεί η συσκευή να υποστηρίξει MMS λειτουργίες, όπως ποιο είναι το μέγιστο μέγεθος αρχείων πολυμέσων που μπορεί να δεχθεί, ποιο το μέγιστο μέγεθος αρχείων, συνολικά, κ.α.

Όπως είδαμε, τα HTTP αιτήματα που εκκινούν από κινητά τηλέφωνα και ασύρματες συσκευές χειρός, συνήθως περιέχουν, μέσα στα metadata που συνοδεύουν τα HTTP μηνύματα, αναφορές προς τον αρμόδιο Profile Repository Server του κατασκευαστή τους, οι οποίες, με κατάλληλη επεξεργασία από την πλευρά του εξυπηρετητή, μπορούν να γνωστοποιήσουν στον εξυπηρετητή τις δυνατότητες τους. Για παράδειγμα, είδαμε ότι μια από τις πολλές επικεφαλίδες που συνοδεύει μια HTTP αίτηση είναι και η «x-wap-profile», που για παράδειγμα, ένα κινητό τηλέφωνο τύπου Nokia 6230i θα έστειλε σε έναν εξυπηρετητή ως:

[HTTP://nds1.nds.nokia.com/uaprof/N6230ir200.xml](http://nds1.nds.nokia.com/uaprof/N6230ir200.xml)

Ανιχνεύοντας, επομένως, ένας εξυπηρετητής εντός των λαμβανομένων αιτήσεων από πελάτες τις αναφορές αυτού του είδους προς κάποιους Profile Repository Servers, μπορεί να ανακτήσει μέσω των URIs αυτών τα υποδεικνυόμενα αρχεία με τις τερματικές δυνατότητες των πελατών του. Μέσω αυτών των αρχείων, ο εξυπηρετητής, αν έχει προγραμματιστεί κατάλληλα, είναι σε θέση να κατανοήσει τους όποιους περιορισμούς έχει μια συσκευή όσον αφορά την προβολή διαδικτυακού περιεχομένου, και να μεριμνήσει ώστε να επιτευχθεί το καλύτερο δυνατό αποτέλεσμα για κάθε πελάτη. Αυτό μπορεί να σημαίνει είτε την προώθηση συγκεκριμένου περιεχομένου στον πελάτη, είτε την προσαρμογή κάποιου κοινού περιεχομένου στα μέτρα και τις δυνατότητες της συσκευής του κάθε πελάτη.

Αδυναμία του μοντέλου αυτού είναι ότι, ενδεχομένως, ένας «ευσυνείδητος» εξυπηρετητής θα χρειαστεί να επικοινωνήσει διαδοχικά με πολλούς User Agent Profile Repository Servers προκειμένου να ανακτήσει την πληροφορία δυνατοτήτων όλων των πελατών του, αφού δεν υπάρχει ένας κεντρικός Profile Repository Server με αποθηκευμένη πληροφορία για όλες τις διαφορετικές τερματικές συσκευές του εμπορίου. Συνήθως, κάθε κατασκευαστής, μέσω του δικού του Profile Repository Server, ανακοινώνει, αποκλειστικά, τα προφίλ των δικών του συσκευών. Γίνεται εύκολα αντιληπτό, επομένως, πόση χρονική επιβάρυνση θα

επέφερε μια τέτοια αρχιτεκτονική στην εξυπηρέτηση των πελατών ενός εξυπηρετητή.

Επιπλέον, η αποστολή τέτοιων αναφορών μέσα από τις επικεφαλίδες του HTTP δεν είναι υποχρεωτική για τους πελάτες. Δηλαδή κάποιοι πλοηγοί μπορεί να συνοδεύουν τα HTTP αιτήματα με τέτοιου είδους πληροφορία και κάποιοι άλλοι, συνήθως οι πιο παλιοί, όχι. Επίσης, η πληροφορία αυτής της μορφής, ακόμη κι αν υπάρχει, δεν είναι αυστηρά καθορισμένο εντός ποιου πεδίου των HTTP μηνυμάτων τοποθετείται. Για παράδειγμα, για το σκοπό αυτό μπορούν να χρησιμοποιηθούν εξίσου τα πεδία «x-wap-profile», «profile» ή «options». Οι ιδιαιτερότητες αυτές, περιπλέκουν ακόμη περισσότερο την προσπάθεια ανίχνευσης κι επεξεργασίας των δυνατοτήτων μιας τερματικής συσκευής-πελάτη.

Μια προσπάθεια απλοποίησης κι επίλυσης των ζητημάτων αυτών, η οποία, αφενός, εξασφαλίζει τη δυνατότητα προσαρμογής του διαδικτυακού περιεχομένου στα χαρακτηριστικά των τερματικών συσκευών, και, αφετέρου, δεν κατακερματίζει τις προφίλ πληροφορίες μεταξύ διαφορετικών Profile Repository Servers, διατηρώντας, συνάμα, ένα ενιαίο πρωτόκολλο ταξινόμησης, είναι η προσπάθεια της κοινότητας του WURFL.

2.3:WURFL:

2.3.1:Εισαγωγή στο WURFL



Σχήμα 4. Το σήμα κατατεθέν του WURFL

Το WURFL (Σχήμα 4), ακρωνύμιο του Wireless Universal Resource FiLe, είναι μια διαδικτυακή κοινότητα, που σκοπό έχει να επιλύσει το πρόβλημα της προσαρμογής και παρουσίασης περιεχομένου σε ασύρματες συσκευές. Είναι μέρος της FOSS (Free and Open Source Software) κοινότητας [31] στο προαναφερόμενο θέμα. Βρίσκεται στην ηλεκτρονική διεύθυνση [HTTP://wurfl.sourceforge.net/index.php](http://wurfl.sourceforge.net/index.php), απ' όπου και μπορούμε να μεταφορτώσουμε δωρεάν, ένα αρχείο XML που έχει καταχωρημένες όλες τις υπάρχουσες τερματικές συσκευές με τις αντίστοιχες δυνατότητές τους. Οι συσκευές διαχωρίζονται μεταξύ τους μέσω της περιγραφής που αποστέλλει κάθε πλοηγός πελάτη εντός της επικεφαλίδας «user agent» των μηνυμάτων του HTTP. Το πεδίο «user

agent» συνοδεύει υποχρεωτικά κάθε HTTP αίτημα, από όποιον πλοηγό κι αν πηγάζει, και συνήθως έχει διαφορετική τιμή για κάθε συσκευή, ακόμη κι αν ανήκουν στον ίδιο κατασκευαστή.

Η προσπάθεια αυτή, ξεκίνησε κοντά το 2001, όταν πια το WAP πρωτόκολλο υποσκελιζόταν από τις απαιτήσεις για διαδικτυακό περιεχόμενο από τον «κανονικό» παγκόσμιο ιστό, οπότε οι Luca Passani και Andrea Trassati, δύο μέλη της FOSS κοινότητας, δημιούργησαν την WURFL κοινότητα, με σκοπό την καταγραφή των δυνατοτήτων των περισσότερων, αν όχι όλων, των ασύρματων συσκευών.

Η κωδικοποίηση σε XML, προτιμήθηκε για την ευχρηστία και την ευελιξία που προσφέρει στη συγγραφή και καταχώριση των τερματικών δυνατοτήτων, καθώς και για το γεγονός ότι η XML αναγνωρίζεται και διαβάζεται από μια πληθώρα προγραμματιστικών γλωσσών και εφαρμογών ταξινόμησης και προβολής πληροφοριών. Περισσότερα για την XML, θα πούμε αργότερα, στο 3^ο κεφάλαιο.

2.3.2: Οργάνωση και δομή του WURFL αρχείου

Το βασικό δομικό στοιχείο που απαρτίζει ολόκληρο το αρχείο του WURFL, είναι το <devices>. Φωλιασμένες μέσα του, βρίσκονται ξεχωριστά, οι επιμέρους εγγραφές των διαφόρων συσκευών, με το αναγνωριστικό <device>. Κάθε στοιχείο <device> περιγράφει και μια διαφορετική τερματική συσκευή. Για κάθε <device>, οι πληροφορίες δίνονται με τα εξής πεδία:

- Id – μοναδικό αναγνωριστικό του στοιχείου.
- User Agent – το κείμενο του πεδίου user agent των HTTP μηνυμάτων.
- Actual device root – προαιρετικό, δηλώνει αν είναι τα πραγματικά στοιχεία που θα έστειλε μια συσκευή, σε αντίθεση με τους κλώνους της ή με το γενικό πρότυπο με το οποίο καταγράφονται οι συσκευές μέσα στο αρχείο.
- Fallback – περαιτέρω πληροφορίες για τον user agent.

Η μορφή έναρξης της καταγραφής, είναι η εξής:

```
<devices>
<device user_agent="O2-Xli"
      actual_device_root="true"
      fall_back="opwv_v61_generic"
      id="o2_xli_ver1">
```

Μέσα στο στοιχείο <device>, οι δυνατότητες μιας συσκευής, καταχωρούνται σε ομάδες, ώστε να διατηρείται μια συνοχή στην

ανάγνωση του αρχείου. Οι ομάδες είναι οι εξής (στις παρενθέσεις, τα ονόματα του πεδίου id του group στοιχείου):

1. Product info (product_info)
2. WML user interface (wml_ui)
3. Compact HTML user interface (chtml_ui)
4. HTML/XHTML-MP user interface (xhtml_ui)
5. CSS support (css)
6. AJAX support (ajax)
7. Markup language support (markup)
8. Packages cache (cache)
9. Display properties (display)
10. Image format (image_format)
11. Any know bug (bugs)
12. WTA support (wta)
13. Security (security)
14. Wireless capabilities (bearer)
15. String restrictions (storage)
16. Object types support (object_download)
17. Video functions support (playback)
18. Digital rights management (drm)
19. Video streaming support (streaming)
20. WAP functions (wap_push)
21. M.M.S. service (mms)
22. S.M.S. service (sms)
23. Java support (j2me)
24. Sound capabilities (sound_format)
25. Flash capabilities (flash_lite)
26. Information transcoding (transcoding)
27. R.S.S. support (rss_support)
28. PDF support (pdf_support)
29. Soon to remove capabilities (Deprecated)

Σε κάθε ομάδα, υπάρχουν ένα ή περισσότερα στοιχεία, που περιγράφουν τις δυνατότητες της συσκευής. Το στοιχείο (ή και στοιχεία), ονομάζεται capability, ταξινομείται κάτω από το στοιχείο Group, και περιγράφει μια ιδιότητα της συσκευής, που την ονομάζει, με ένα πεδίο name. Παραδίπλα, ένα άλλο πεδίο, το value, δίνει την τιμή του στοιχείου, είτε είναι αριθμός, είτε είναι κείμενο. Παράδειγμα για την τελική μορφή μιας καταχώρισης συσκευής, δίνεται παρακάτω:

```
<devices>
<device
```

```

id="nokia_n95_ver1_sub_mozilla_d"
user_agent="Mozilla/5.0 (SymbianOS/9.2; U; Series60/3.1
NokiaN95_8GB/10.0.021; Profile/MIDP-2.0 Configuration/CLDC-1.1 )
AppleWebKit/413 (KHTML, like Gecko) Safari/413"
fall_back="nokia_n95_ver1_sub_mozilla">

  <group id="bearer">
    <capability name="wifi" value="true"/>
    <capability name="max_data_rate" value="1800"/>
  </group>

  <group id="css">
    <capability name="css_supports_width_as_percentage"
value="true"/>
  </group>
</device>
</devices>

```

Παρατηρούμε πως, όποιο στοιχείο ξεκινήσουμε, το τελειώνουμε με το σύμβολο / και το όνομα του στοιχείου, ενδεικτικό ότι η αντίστοιχη περιγραφή ολοκληρώθηκε. Έτσι ομαδοποιούνται τα στοιχεία, γενικά, στην XML προτυποποίηση, κάτι που θα δούμε και στο κεφάλαιο 3.

Λεπτομερής αναφορά στα ονόματα των δυνατοτήτων κάθε συσκευής που ταξινομούνται ανά ομάδα, καθώς και στις τιμές που μπορούν να πάρουν τα διάφορα πεδία, περιέχεται στην τεκμηρίωση του WURFL που βρίσκεται αναρτημένη στην ιστοσελίδα [HTTP://wurfl.sourceforge.net/help_doc.php](http://wurfl.sourceforge.net/help_doc.php). Ενδεικτικά, η ταξινόμηση έχει τη μορφή που απεικονίζει το Σχήμα 5.

Group: **display**

Capability Name	Type	Description
resolution_width	any integer number	This field represents the screen width expressed in pixels
resolution_height	any integer number	This field represents the screen height expressed in pixels
columns	any integer number	Number of columns presented
rows	any integer number	Number of lines presented
max_image_width	any integer number	Width of the images viewable (usable) width expressed in pixels
max_image_height	any integer number	Height of the images viewable (usable) width expressed in pixels
dual_orientation	true/false	Some devices may be flipped, i.e. user may change orientation, effectively inverting screen_width and screen_height for mobile web browsing and, possibly, for other functions.

Σχήμα 5. Μορφή της ταξινόμησης των δυνατοτήτων, στην ιστοσελίδα τεκμηρίωσης του WURFL

Με μαυρισμένα γράμματα αναφέρεται το όνομα της ομάδος και μέσα στον πίνακα, το όνομα της δυνατότητας. Δίπλα, στη δεύτερη στήλη του πίνακα, αναφέρεται ο τύπος της τιμής που θα έχει η δυνατότητα, πάντα σε μορφή κειμένου, αλλά με διαφορετική, ανάλογα με τον τύπο της δυνατότητας, έννοια. Τέλος, στην Τρίτη στήλη, περιέχονται επεξηγηματικά σχόλια σχετικά με τη δυνατότητα, καθώς και κάποιες κρίσιμες παρατηρήσεις.

2.4: Πως χρησιμοποιούμε το WURFL:

2.4.1: Parsing

Ο όρος parsing αναφέρεται στην επεξεργασία ενός εγγράφου με σκοπό την ανίχνευση και απομόνωση των επιθυμητών αποσπασμάτων. Στην περίπτωσή μας, ζητάμε να διαβάσουμε τα στοιχεία που περιέχονται εντός του αρχείου wurfl.xml, προκειμένου να αναλύσουμε και να απομονώσουμε τις τιμές που λαμβάνουν κάποια συγκεκριμένα πεδία δυνατοτήτων, με σκοπό να τα εντάξουμε στο κεντρικό μας πρόγραμμα, την ρουτίνα λειτουργίας του servlet.

Το parsing του wurfl αρχείου αναλαμβάνει να το κάνει μια ξεχωριστή εφαρμογή, ο parser, ο οποίος τρέχει παράλληλα με το κυρίως πρόγραμμα επεξεργασίας των HTTP αιτημάτων και το τροφοδοτεί με δεδομένα από το αρχείο του wurfl όπου απαιτείται. Ανάλογα με τις ανάγκες του εξυπηρετητή, υπάρχουν πολλοί τρόποι υλοποίησης ενός parser. Στην υλοποίηση μας, όπως θα δούμε παρακάτω, ο parser είναι μια ξεχωριστή java κλάση, ένα instance της οποίας δημιουργείται όταν την καλέσει ο servlet.

2.4.2: Άντληση στοιχείων

Όπως είδαμε στη δομή του αρχείου, κάθε στοιχείο δυνατότητας της συσκευής, έχει ένα πεδίο με το όνομα αυτής και μια τιμή. Ο parser διαβάζει τα ονόματα των στοιχείων capabilities, και αν βρει πως υπάρχει κάποιο απ' αυτά που μας ενδιαφέρουν, καταχωρεί σε μια κοινοποιημένη μεταβλητή την τιμή αυτού, για μελλοντική χρήση.

Λόγω του ότι το αρχείο περιέχει καθαρό κείμενο και μόνον, πρέπει να δοθεί προσοχή κατά τη μεταφορά των τιμών του αρχείου στις μεταβλητές μιας εφαρμογής. Ιδιαίτερη μνεία πρέπει να γίνει στη μεταφορά αριθμητικών δεδομένων, τα οποία αν και στο αρχείο του wurfl περιέχονται ως κείμενο, πρέπει να αναγνωριστούν από τον μεταγλωττιστή ως αριθμοί.

2.5:Συνδιασμός WURFL με τους user agents:

2.5.1:WURFL αντί UAProf

Η πρόταση του WURFL, έγκειται στην υποκατάσταση του υπερσυνδέσμου του πεδίου του UAProf στα μηνύματα του HTTP, με μια διαφορετική ρουτίνα προγράμματος από την τεχνολογία του CGI, για την εύρεση πληροφοριών από ένα αρχείο καταγραφής, το αρχείο wurfl.xml, των δυνατοτήτων της συσκευής.

Αυτή η προσέγγιση, δεν καταπονεί το CGI με το να ψάχνει το αρχείο δυνατοτήτων της κάθε συσκευής ξεχωριστά στο διαδίκτυο, δεσμεύοντας διαδικτυακούς και φυσικούς πόρους που μπορεί και να μην αποδεσμευτούν αργότερα.

2.5.2:Δυναμική ανίχνευση του προφίλ UAProf ανάλογα με τον τύπο user agent της συσκευής-πελάτη

Με την ανάγνωση του πεδίου «user agent» στα metadata της επικεφαλίδας ενός HTTP μηνύματος από τον servlet, και εξαγωγή από το wurfl αρχείο των δυνατοτήτων της συσκευής με αυτόν τον agent, το σύστημα τροφοδοτείται δυναμικά με τις πληροφορίες δυνατοτήτων που έχουμε συμπεριλάβει στον servlet μας, προκειμένου να διαφοροποιήσουμε την παροχή περιεχομένου προς τον εκάστοτε πελάτη. Μπορούμε, για παράδειγμα, να αποκλείσουμε πληροφορίες για την WML γλώσσα, αν δεν πρόκειται να χρησιμοποιήσουμε το πρωτόκολλο WAP για επικοινωνία. Μπορούμε, επίσης, να βάλουμε μια συνθήκη, πως μόνο κινητά με δυνατότητα αναπαραγωγής video, θα μπορούν να έχουν πρόσβαση στην ιστοσελίδα μας, αν θέλουμε να έχουμε εκεί ένα εισαγωγικό, επεξηγηματικό video. Στην περίπτωση της εργασίας μας, απώτερος στόχος μας δεν είναι να αποκλείσουμε πελάτες, αλλά, αντίθετα, να επιτρέψουμε σε κάθε τερματική συσκευή, ανεξαρτήτως δυνατοτήτων, να έχει πρόσβαση στην αναρτημένη πληροφορία, προσαρμόζοντας δυναμικά το περιεχόμενο που θα βλέπει ο τελικός χρήστης στις δυνατότητες της τερματικής του συσκευής. Δηλαδή, το ίδιο περιεχόμενο θα προσφέρεται σε εναλλακτικές μορφές, κατάλληλο για χρήστες με διαφορετικών δυνατοτήτων τερματικές συσκευές.

2.5.3:Παράδειγμα

Έστω πως υλοποιούμε μια εφαρμογή servlet, που ελέγχει τον τύπο user agent της συσκευής του πελάτη, και αν βρεθεί μέσω του wurfl πως η συσκευή έχει δυνατότητα προβολής εικόνων τύπου tiff, τότε να αναγνωρίζει και ποιο είναι το μέγιστο πλάτος και ύψος της εικόνας που μπορεί να προβληθεί στην αντίστοιχη οθόνη του πελάτη.

Για παράδειγμα, στο WURFL περιέχεται η καταγραφή για το κινητό Blackberry 7130 (Σχήμα 6), που όπως βλέπουμε, έχει μορφή:


```
<device id="blackberry7130_ver1" user_agent="BlackBerry7130"
fall_back="blackberry_generic_ver4_sub10" actual_device_root="true">
  <group id="product_info">
    <capability name="has_qwerty_keyboard" value="false"/>
    <capability name="uaprof"
value="HTTP://www.blackberry.net/go/mobile/profiles/uaprof/7130/4.1.0
3.rdf"/>
    <capability name="model_name" value="BlackBerry 7130"/>
  </group>
  <group id="image_format">
    <capability name="jpg" value="true"/>
    <capability name="gif" value="true"/>
    <capability name="bmp" value="true"/>
    <capability name="wbmp" value="true"/>
    <capability name="gif_animated" value="true"/>
    <capability name="png" value="true"/>
    <capability name="colors" value="65536"/>
    <capability name="tiff" value="true"/>
  </group>
  <group id="markup">
    <capability name="html_wi_oma_xhtmlmp_1_0" value="true"/>
    <capability name="html_wi_w3_xhtmlbasic" value="true"/>
    .....
  </group>
  <group id="display">
    <capability name="columns" value="26"/>
    <capability name="rows" value="18"/>
    <capability name="max_image_width" value="235"/>
    <capability name="resolution_width" value="240"/>
    <capability name="resolution_height" value="260"/>
    <capability name="max_image_height" value="240"/>
  </group>
  <group id="storage">
    <capability name="max_deck_size" value="32768"/>
  </group>
  <group id="object_download">
    .....
  </group>
</device>
```



Σχήμα 6. Η συσκευή BlackBerry 7130

Συγκεκριμένα, περιέχονται πληροφορίες για τις δυνατότητες της συγκεκριμένης συσκευής, όπως ο user agent αυτής, ο υπεσύνδεσμος του UAProf, καθώς και πολλές ακόμα που δεν μας χρειάζονται στην παρούσα φάση.

Την προσοχή μας κεντρίζουν δύο ομάδες δυνατοτήτων, η `image_format` και η `display`. Στην πρώτη, στη δυνατότητα με το όνομα `tiff` βλέπουμε πως η τιμή της είναι `true`, δηλαδή, η συσκευή μπορεί να προβάλλει εικόνες τέτοιου τύπου. Με την βοήθεια του `parser`, τοποθετούμε σε μια μεταβλητή τύπου `Boolean` του σχετικού `servlet` την τιμή `true`.

Η δεύτερη ομάδα που μας ενδιαφέρει, η `display`, περιέχει δύο δυνατότητες, την `max_image_width` και την `max_image_height`, οι οποίες εκφράζουν την αριθμητική τιμή της μέγιστης ανάλυσης, μετρώντας την σε κουκκίδες οθόνης, με την οποία μπορεί να αποδώσει αυτό το κινητό μια εικόνα. Αυτές οι τιμές, καταχωρούνται σε δύο μεταβλητές τύπου `Integer`, της προαναφερθείσας εφαρμογής.

Όταν τώρα η εφαρμογή ξεκινήσει τις συνθήκες ελέγχου για το αν θα δώσει υλικό εικόνας στη συσκευή αυτή, θα αναγνωρίσει, μέσω της μεταβλητής `Boolean`, ότι η συσκευή διαθέτει τη δυνατότητα προβολής του συγκεκριμένου τύπου εικόνας, και θα διαμορφώσει τις διαστάσεις της εικόνας ώστε να συμφωνούν με τις τιμές των δύο μεταβλητών τύπου `Integer`.

Αυτό που επιτυγχάνουμε, δηλαδή, είναι ότι το CGI μας δε χρειάζεται να ανοίξει σύνδεση με την ιστοσελίδα που περιέχει το αντίστοιχο UAProf προφίλ για να αντλήσει τα στοιχεία μέσω διαδικτύου. Αντ' αυτού, με μια απλή σάρωση του `parser` στο αρχείο `wurfl.xml`, που περιέχεται στον αποθηκευτικό χώρο του `Server`, λαμβάνουμε τις τιμές που περιγράφουν τις δυνατότητες της συσκευής-πελάτη, για χρήση κατά βούληση, χωρίς να ξοδεύουμε το χρόνο που θα απαιτούσε η σύνδεση του `Servlet` με τους εκάστοτε απομακρυσμένους αποταμιευτήρες των UAProfs.

Κεφάλαιο 3^ο – Δομικά στοιχεία μιας ιστοσελίδας

3.1:HTML: Μια ιστορική αναδρομή:

3.1.1:ENQUIRE – Μια αρχή με πολλές υποσχέσεις

Στα τέλη της δεκαετίας του 1980, στον Ευρωπαϊκό οργανισμό για πυρηνική έρευνα (CERN), ο Tim Berners-Lee, ένας από τους ερευνητές που περάσανε από αυτό το ερευνητικό κέντρο, προτυποποίησε ένα σύστημα διαμοιρασμού και χρήσης αρχείων και εγγράφων, για τους ερευνητές του CERN, που ονομάστηκε ENQUIRE.



Σχήμα 7. Κοινόχρηστα αποθηκευτικά μέσα της εταιρίας Norsk Data.
Σε τέτοια μηχανήματα, δούλευε το πρώτο Internet.

Αυτό το σύστημα, είχε πολλά κοινά με το γνωστό σε όλους μας WWW του διαδικτύου. Επέτρεπε αμφίδρομες διαδρομές των αρχείων που διατηρούνταν αποθηκευμένα σε μια βάση δεδομένων της εταιρίας Norsk Data (σχήμα 7). Τα αρχεία αυτά, επιτρεπόταν να χρησιμοποιηθούν και να τροποποιηθούν απ' όλους, καθώς και να κοινοποιηθεί το τελικό, επεξεργασμένο κείμενο. Η λειτουργία αυτού του συστήματος βέβαια, θυμίζει περισσότερο τα σημερινά WiKi, όπου μια ιστοσελίδα είναι το αποτέλεσμα ενός συνόλου άρθρων, που τα γράφει από κοινού η διαδικτυακή κοινότητα, δηλαδή καθένας που έχει πρόσβαση στο διαδίκτυο.

Το σύστημα αυτό ήταν γραμμένο στη γλώσσα Pascal, και το πλήρες όνομα, σύμφωνα με τον Tim Berners-Lee, ήταν «Enquiry Within Upon Everything».

3.1.2: Η ιδέα του υπερκειμένου

Με την απόδοση σε λειτουργία του συστήματος ENQUIRE, ο ερευνητής του CERN αναφέρθηκε στην παγκόσμια κάλυψη του συστήματος, με μια γλώσσα υπερκειμένου, μια μορφή απεικόνισης πληροφοριών πέρα από το στατικό κείμενο, χάρη στις πολυμεσικές δυνατότητες των Η/Υ. Μια σηματοδοτική γλώσσα ήταν αναγκαία για το πού θα βρίσκονται τα πολυμεσικά αυτά στοιχεία, καθώς και πως θα δομείται η ιστοσελίδα υπερκειμένου, οπότε και αναπτύχθηκε στις αρχές του '90 η HyperText Markup Language, ή αλλιώς HTML [9].

3.1.3: HTML, η γέννηση

Ένα έγγραφο με όνομα «HTML tags» ήταν η αρχική αναφορά στη σηματοδοτική γλώσσα που χειριζόμαστε σήμερα για δημιουργία ιστοσελίδων. Είκοσι (20) ετικέτες, από τις οποίες οι δεκατρείς (13) χρησιμοποιούνται ακόμα και σήμερα, ήταν τα δομικά στοιχεία των πρώτων ιστοσελίδων, για την απεικόνιση της πληροφορίας, άμεσα και ολοκληρωμένα.

Η όλη προσπάθεια εντάχθηκε στα πλαίσια προτυποποίησης βάση της SGML (Standard Generalized Markup Language), καθώς η HTML αντιμετώπιστηκε σαν επέκταση αυτής, κυρίως για τα πρότυπα που έφερε. Συγκεκριμένα:

- Η σηματοδότηση πρέπει να περιγράφει την δομή της πληροφορίας στο κείμενο, παρά να περιγράφει τη λειτουργικότητα της.
- Η σηματοδότηση αυτή πρέπει να είναι αυστηρή, προκειμένου να επιτρέπει η μελλοντική χρήση κι επεξεργασία δεδομένων σε επόμενα κείμενα.

Αργότερα, και πιο συγκεκριμένα μετά από έξι μήνες, παρατηρήθηκε πως η κατά ISO προτυποποιημένη SGML, ήταν πιο δύσκαμπτη σε μεταβολές και χρήσεις, απ' ότι η HTML. Αποφασίστηκε λοιπόν, με την έναρξη λειτουργίας του παγκόσμιου διαδικτύου, να χρησιμοποιηθεί μόνο η HTML σαν γλώσσα σηματοδότησης και δομής των ιστοσελίδων.

Έτσι, το 2000, η HTML έγινε το παγκόσμιο πρότυπο για τη συγγραφή των υπερκειμένων μιας ιστοσελίδας, εμπλουτισμένη και προτυποποιημένη από το World Wide Web Consortium (W3C), τον οργανισμό που ασχολείται με την HTML σήμερα.

3.2: Εκδόσεις της HTML:

3.2.1: HTML

Στα μέσα της δεκαετίας του '90, η IETF ανακοίνωσε την έκδοση 2.0 της HTML, η οποία ήταν εμπλουτισμένη με υποστήριξη παραμέτρων για τη μορφοποίηση ενός κειμένου. Έπειτα, εισήλθαν οι πίνακες, οι διαμορφώσεις εικόνας από πλευράς πελάτη, η εισαγωγή μορφοποιήσεων χαρακτήρων κειμένων κ.α.

Η νεοσύστατη W3C, το 1997, ανακοινώνει την έκδοση 3.2. Αυτή, περιέχει πια εντολές για μαθηματικές φόρμουλες, ημι-πολυμεσικά αντικείμενα και αναπτυγμένες λειτουργίες παρουσίασης κειμένου.

Το 1998, περνάμε πια στην HTML 4.0, μια έκδοση που μοιάζει πια με την σημερινή γλώσσα που χρησιμοποιούμε. Ανά καιρούς, η W3C εμπλουτίζει με νέες ετικέτες την γλώσσα, για να προλάβει τις εξελίξεις στα πολυμέσα, όπως η προβολή video, η ενσωμάτωση ενός player, στερεοφωνικές υποστηρίξεις μουσικών αρχείων κ.α.

Το 2008, η W3C, απορρίπτοντας πια την SGML προτυποποίηση, ανακοινώνει με την έκδοση 5.0, πως η HTML είναι μια αυτόνομη γλώσσα, ικανή να διαχειρίζεται και να μορφοποιεί τα περισσότερα γνωστά standards πολυμέσων που υπάρχουν στην σημερινή τεχνολογία των Η/Υ.

3.2.2: XHTML

Η XHTML είναι μια ξεχωριστή γλώσσα, που βασίζεται στον συνδυασμό της HTML 4.0 και της XML 1.0. Έγινε, γιατί από το 2000 και μετά, έγινε φανερό ότι το πολυμεσικό περιεχόμενο γίνεται σημαντικός παράγοντας στην απόδοση μιας πληροφορίας και η HTML, που ήταν ακόμα υπό ανάπτυξη, έπρεπε να μπορεί να αποδώσει όσον το δυνατόν καλύτερα αυτές τις πληροφορίες.

Έτσι, δημιουργήθηκε η XHTML, ως απότοκος της «Reformulating HTML in a XML» του 1998, που προτάθηκε στην W3C.[7] Η γλώσσα αυτή, από την πρώτη έκδοση της μέχρι τώρα, ενσωμάτωσε αλλαγές επί της αυστηρής προτυποποίησης των πολυμεσικών στοιχείων, σχετικά με ποια standards θα χρησιμοποιούνται και πως. Επίσης, εμπλουτίστηκε με στοιχεία που συναντάμε σε εφαρμογές επεξεργαστών κειμένων, όπως την γεωμετρική τοποθέτηση ενός κειμένου πάνω στη σελίδα, στηλοθέτες, κ.α., που επιτρέπουν αποδοτικότερη μορφοποίηση των κειμένων.

3.2.3: XHTML - MP.

Η XHTML – MP είναι μια έκδοση XHTML για κινητά τηλέφωνα. Ξεκίνησε την ίδια περίοδο με την κύρια XHTML, γι' αυτό και δεν παρατηρούμε ιδιαίτερες διαφορές σε αυτές τις δύο, γιατί εξελίσσονταν μαζί, με κοινές προτυποποιήσεις και τεχνολογίες.

Η γλώσσα αυτή, απαιτεί από ένα κινητό τηλέφωνο να μπορεί να αναγνωρίζει μια περιγραφή DOCTYPE στην ιστοσελίδα που θα αναγνώσει, για μετατροπή των standards εμφάνισης της ιστοσελίδας κατά την προβολή της στη συσκευή αυτή. Η περιγραφή αυτή, είναι ένα κείμενο που παρεμβάλεται στην επικεφαλίδα της ιστοσελίδας, ώστε να συνδέσει τον υπόλοιπο κώδικα της HTML με ένα προσδιορισμό στα θέματα μορφοποίησης, απόδοσης και χρήσης των δεδομένων που απαραίτουν την πληροφορία.

Επίσης, απαιτεί οι φορητές συσκευές να υποστηρίζουν τη μορφή HTML και XML, στη MIME περιγραφή της ιστοσελίδας που πρόκειται να προβάλλουν (στο πρότυπο MIME αναφερθήκαμε στην παράγραφο 2.1.1).

3.2.4: Δυναμική XHTML

Η δυναμική XHTML, είναι ένας όρος «ομπρέλα», ο οποίος καλύπτει τις δραστηριότητες πολλών διαφορετικών τεχνολογιών για την παραγωγή μιας ιστοσελίδας, που δεν είναι αποθηκευμένη σε ένα στατικό αρχείο σε κάποιο Server αλλά παράγεται ανάλογα με τις ανάγκες και τα πρότυπα επικοινωνίας μεταξύ πελάτη – εξυπηρετητή.

3.3: Μορφή της HTML:

3.3.1: Τρόπος συγγραφής του κώδικα

Η απλότητα της HTML, έγκειται στο ότι δεν χρειαζόμαστε εξειδικευμένα εργαλεία, στην δόμηση μιας ιστοσελίδας. Ένας απλός κειμενογράφος, είναι ότι χρειαζόμαστε για να ξεκινήσουμε την υλοποίηση της ιστοσελίδας μας. Το κείμενο εντός του οποίου θα γράψουμε τις ετικέτες μας, καλό είναι να αποτελείται από χαρακτηρισες κάποιου διεθνούς προτύπου, για πλήρη υποστήριξη από τους χρήστες του διαδικτύου. Ένα παράδειγμα γραμμένο σε κώδικα HTML είναι αυτό που ακολουθεί:

```
<html>
<head>
<title>HelloWorld application</title>
</head>
<body>
<p><h1>Hello World!</h1></p>
<br/>
<h2>How are you?</h2>
</body>
</html>
```

Όταν γράφουμε μια ετικέτα, πρέπει μετά να την κλείσουμε, με το σύμβολο / και το όνομα της ετικέτας που έχουμε ανοιχτή. Μεταξύ των ετικετών, ότι κείμενο παρεμβάλεται, αναγνωρίζεται σαν κείμενο

πληροφορίας ή λειτουργίας, ανάλογα με την ετικέτα και την λειτουργία της. Έτσι, η ετικέτα:

```
<p>Hello World</p>
```

Θα μας δείξει στην οθόνη μας το μήνυμα «Hello World», στοιχειοθετημένο σε μία παράγραφο. Μια ετικέτα του τύπου:

```

```

Θα μας δείξει στην οθόνη μια εικόνα που έχουμε αποθηκευμένη στον σκληρό μας δίσκο. Αν δεν υπάρχει, θα προβάλλει μήνυμα «No picture». Επίσης, μπορούμε να κλείσουμε την εντολή με το να βάλουμε τον χαρακτήρα / στο τέλος, πριν τον χαρακτήρα > κλεισίματος της ετικέτας.

Ανάλογα με τον τύπο της εντολής, πρέπει να δίδονται και τα αντίστοιχα ορίσματα, ώστε τελικά να παράγεται η επιθυμητή ιστοσελίδα, που θα προβάλλει τις ζητούμενες πληροφορίες. Βιβλία και διαδικτυακοί ιστότοποι με μαθήματα κατασκευής ιστοσελίδων, υπάρχουν για να εξηγούν και να υπενθυμίζουν τον τρόπο χρήσης των ετικετών της HTML.

3.3.2: Δόμηση του κώδικα

Απαραίτητα, η αρχή του κώδικά μας είναι η ετικέτα <html>. Γράφοντας αυτό, και αποθηκευοντάς το σε αρχείο με τύπο .htm ή .html, ο πλοηγός θα το αναγνωρίσει αμέσως.

Μετά ακολουθεί η ετικέτα <head>, όπου περιέχονται πληροφορίες σχετικά με την μορφοποίηση του κειμένου, καθώς και κομμάτια κώδικα Javascript και Ajax. Πρέπει να περιέχεται και μια ετικέτα <title>, που παρουσιάζει το κείμενο στο όνομα του παραθύρου που βλέπουμε στον φυλλομετρητή.

Κλείνοντας τις ετικέτες, μέχρι και την head, καλύψαμε τις πληροφορίες της επικεφαλίδας της ιστοσελίδας. Περνάμε μετά στο κυρίως θέμα της, ανοίγοντας την ετικέτα <body>. Μέσα, ότι εισάγουμε από ετικέτες, θα παρουσιάζονται στο κυρίως παράθυρο απεικόνισης ιστοσελίδων του πλοηγού. Αν εισάγουμε απλό κείμενο, αυτό θα παρουσιαστεί σε μια σειρά, χωρίς καμία μορφοποίηση. Η HTML δεν καταλαβαίνει ειδικούς χαρακτήρες, όπως περισσότερα κενά από ένα ή το Return, παρά μόνο βάση της δικής της κωδικοποίησης, που είναι:

&# αύξων αριθμός χαρακτήρα ή συνθηματικό ;

Έτσι, για τον χαρακτήρα «"», έχουμε:

" ή &#quot;

Αφού ολοκληρώσουμε την περιγραφή της ιστοσελίδας μας, τότε κλείνουμε το κυρίως θέμα, με την ετικέτα `</body>`. Μετά δεν παρεμβάλεται κάτι άλλο, οπότε μπορούμε να κλείσουμε με τη γενική ετικέτα σηματοδότησης του τέλους ενός αρχείου σε γλώσσα HTML, που είναι η ετικέτα `</html>`.

3.4:Τι είναι η XML:

3.4.1:Ανάγκη για ελεύθερη οργάνωση στοιχείων

Το 1996, μια πιο ελεύθερη μορφή περιγραφής στοιχείων κρίθηκε αναγκαίο να παρουσιαστεί στην διαδικτυακή κοινότητα, για την οργάνωση και κατάταξη πολλών τύπων δεδομένων, που τα στενά όρια της SGML καθιστούσε αδύνατη την αξιοποίησή τους. Οι Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau, John Cowan, έδωσαν στην δημοσιότητα μια σειρά κανόνων και προτυποποιήσεων για την παραγωγή XML αρχείων [11], αρχείων, δηλαδή, που περιέχουν μια ελεύθερη και κατά τη βούληση του χρήστη περιγραφή των δεδομένων που θα συμπεριληφθούν σε μια ιστοσελίδα.

Η XML είναι ακρονύμιο της φράσης eXtensible Markup Language, που σημαίνει επεκταμένη σηματοδοτική γλώσσα.

3.4.2:Παράδειγμα

Έστω πως θέλουμε να αντιστοιχήσουμε σε κάθε πελάτη βιβλιοθήκης ονοματεπώνυμο, εικόνα βιβλίου που δανείστηκε, μέρες που το έχει και πόσες μέρες έχει ακόμα περιθώριο να το κρατήσει, πριν του επιβληθεί πρόστιμο. Μια μορφή του αρχείου XML θα μπορούσε να είναι η ακόλουθη:

```
<?xml version="1.0" encoding="UTF-8" ?>
<library>
  <loaner>
    <name>John</name>
    <surname>Martins</surname>
    <img_book>The wind in the willows</img_book>
    <days_has>3</days_has>
    <days_left>7</days_left>
  </loaner>
</library>
```

Παρατηρούμε πως υπάρχει μια σχέση με την HTML, σχετικά με την καταχώρηση των δεδομένων. Η μεγάλη διαφορά είναι, πως δεν υπάρχουν standard ετικέτες, αλλά δημιουργούμε τις δικές μας, δίνοντας τους ότι όνομα επιθυμούμε. Το ζητούμενο είναι να κάνουμε μια

διαστρωμάτωση, ώστε να υπάρχει ένα γονικό στοιχείο, στην περίπτωσή μας το <library>, το οποίο θα απαρτίζεται από στοιχεία – παιδιά, που μπορούν να περιέχουν αναφορές σε υλικό ή και κείμενο.

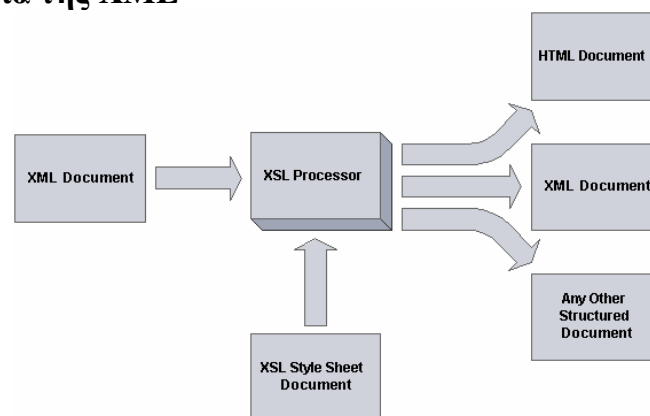
3.4.3: Δυνατότητες

Πέρα από την οργάνωση των δεδομένων για κατασκευή ιστοσελίδων, πολλές εταιρίες ανάπτυξης εφαρμογών, βρήκαν ιδανικό το σχήμα ταξινόμησης της XML. Έτσι, σήμερα οι περισσότερες εφαρμογές Βάσεων Δεδομένων και πελατολογίων, κρατούν τα δεδομένα των πελατών σε τέτοια αρχεία, καθώς οι εφαρμογές ανατρέχουν στην XML δομή που, ξεχωριστά, κάθε εταιρία έχει αναπτύξει, για την εξαγωγή της πληροφορίας. Επίσης, και σε θέματα ρύθμισης και λειτουργίας hardware, υπάρχουν τέτοιες δομές για καταχώρηση στοιχείων επικοινωνίας της μηχανής του Η/Υ, με το περιφερειακό.

Η πραγματική δύναμη όμως της XML είναι, η δυνατότητα μετασχηματισμού των στοιχείων που περιέχει ένα XML αρχείο, σε έναν άλλον τύπο σηματοδοτικής γλώσσας, όπως η HTML. Οι δυναμικά παραγόμενες ιστοσελίδες γίνονται πραγματικότητα, χάρη στην ευελιξία της XML: μέσω της ελευθερίας που προσφέρει η XML στην οργάνωση των δεδομένων, μπορούν να ξεπεραστούν οι φραγμοί της HTML όσον αφορά τη στοιχειοθέτηση των δεδομένων μας.

3.5: Τι είναι το XSLT:

3.5.1: Στα χνάρια της XML



Σχήμα 8. Σχεδιάγραμμα μεταγλώττισης ενός XML αρχείου σε άλλη μορφή, με χρήση του XSLT

Όπως αναφέρθηκε, η XML προσφέρει ευελιξία στη μετατροπή κειμένου σε άλλη μορφή σηματοδοτικής γλώσσας, όπως σε μια άλλη XML, αλλά και σε HTML. Η τελευταία μετατροπή μπορεί να επιτευχθεί εύκολα, μέσω χρήσης της γλώσσας XSL (eXtensible Stylesheet Language) [15]. Αυτή, θέτει κάποιες φόρμες μορφοποίησης, τους λεγόμενους μετασχηματισμούς, ώστε ως XSLT (eXtensible Stylesheet Language

Transformations) [14], πλέον, να παράξει HTML κώδικα, λαμβάνοντας δεδομένα από ένα XML αρχείο (Σχήμα 8). Όπως και η HTML, έτσι και η XSL, θέτει σηματοδοτικές κατευθύνσεις για το πού θα τοποθετηθούν τα δεδομένα, αλλά είναι πιο αυστηρή σε θέματα μορφοποίησης κι απόδοσης της πληροφορίας.

3.5.2: Παράδειγμα.

Έστω ότι τώρα, θέλουμε να στείλουμε μια ιστοσελίδα στο κινητό του αναγνώστη του βιβλίου που το δανείστηκε από την προαναφερόμενη βιβλιοθήκη, η οποία θα δείχνει πόσες μέρες μπορεί να το κρατήσει ακόμα, όπως και την εικόνα του βιβλίου που δανείστηκε. Ο κώδικας σε XSL θα είναι:

```
<xsl:stylesheet version="1.0"
xmlns:xsl="HTTP://www.w3.org/1999/XSL/Transform">

<xsl:output method="html" encoding="utf-8" doctype-public="-
//WAPFORUM//DTD XHTML Mobile 1.0//EN" doctype-
system="HTTP://www.wapforum.org/DTD/xhtml-mobile10.dtd"/>

<xsl:template match="/">

<html>
<head>
<title>A reminder!</title>
</head>
<body>
<p>Greetings <xsl:value-of select="name"/>. You have loaned this
book:
" />
<br/>
Remember! You have <xsl:value-of select="days_left"/> days to return
it.
<br/>
Thank you for your time!</p>
</body>
</html>

</xsl:template>
</xsl:stylesheet>
```

Παρατηρούμε ότι στην ουσία, γράφουμε HTML κώδικα, αλλά αυτός είναι περικλειόμενος από την XSL γλώσσα. Θέτοντας το `template`, ζητάμε ο μεταφραστής να μετατρέψει το XML σε λειτουργικό HTML, που μάλιστα θα μπορεί να το αναγνώσει κινητό τηλέφωνο, μιας και θέσαμε συγκεκριμένο DOCTYPE, από την αρχή:

```
<xsl:output method="html" encoding="utf-8" doctype-public="-
//WAPFORUM//DTD XHTML Mobile 1.0//EN" doctype-
system="HTTP://www.wapforum.org/DTD/xhtml-mobile10.dtd"/>
```

Αφού ενημερώσαμε τον μεταφραστή για το τι τύπου κείμενο θα μεταγλωττίσουμε, του λέμε επίσης τι να πάρει από το αρχείο XML:

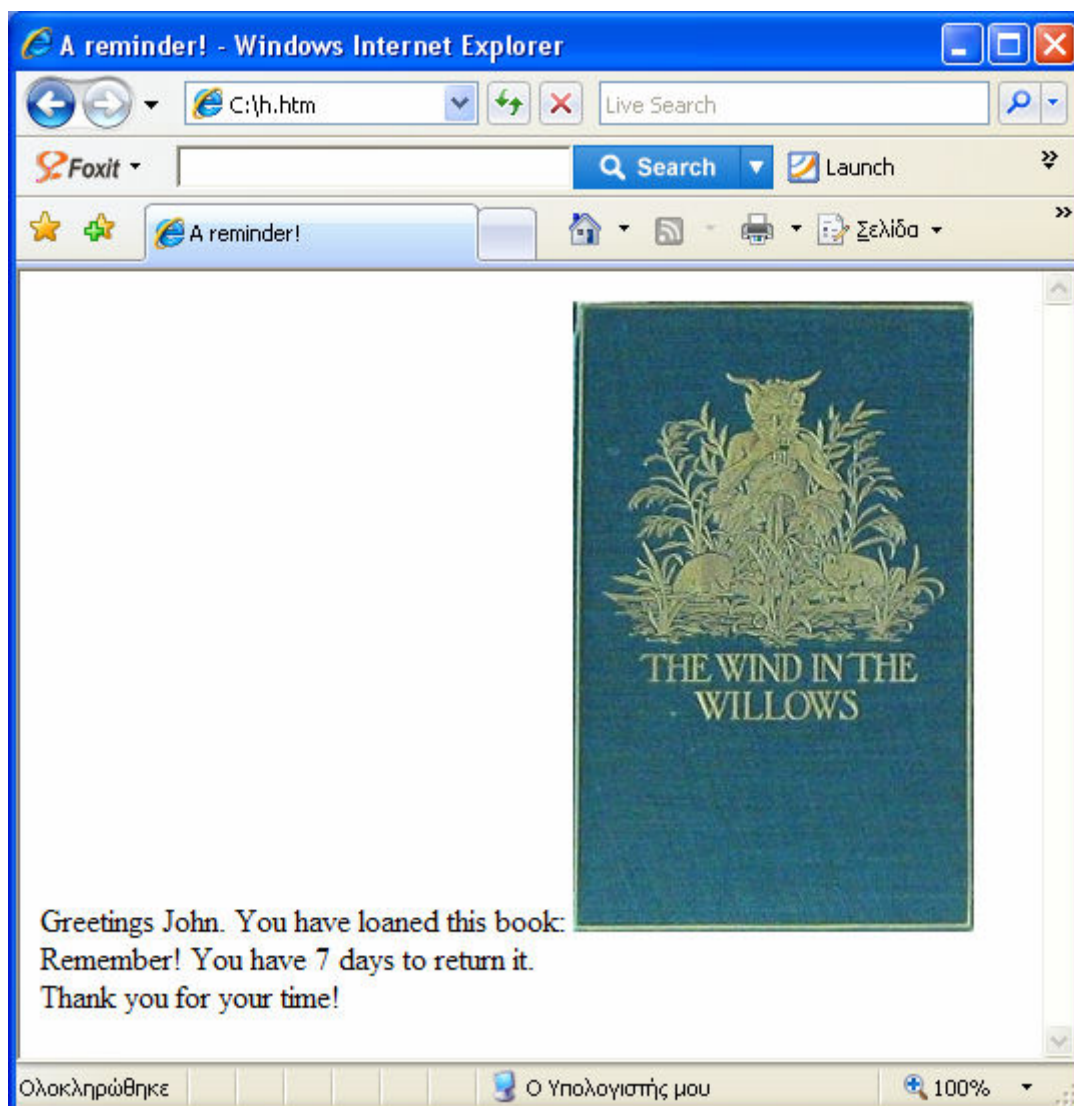
```
<xsl:template match="/">
```

Του λέμε να συμπεριλάβει όλο το κείμενο που θα διαβάσει από το αρχείο, ώστε να κάνει μια πλήρη αναφορά των στοιχείων, όποτε τα ζητάμε. Μετά, ακολουθεί ο κώδικας, και όπου ζητείται ένα στοιχείο, το καλούμε με την παρακάτω εντολή:

```
<xsl:value-of select="name"/>
```

Τα ορίσματα που παίρνει η εντολή αυτή της XSL, είναι συγκεκριμένα, αλλά οι τιμές καλούνται από τα πεδία του XML που εμείς ονομάσαμε, οπότε η ελευθερία κινήσεων που μας δίνεται, στο θέμα της οργάνωσης και χρήσης των στοιχείων, είναι απεριόριστη.

Το Σχήμα 9 απεικονίζει το αποτέλεσμα που παράγεται από την εφαρμογή των περιορισμών που περιέχει το αρχείο xslt πάνω στο αρχείο xml.



Σχήμα 9. Η παραγόμενη ιστοσελίδα, μετά την μετάφραση του XML με την χρήση ενός XSLT

Η μεγάλη δύναμη των μετασχηματισμών XSLT είναι ότι φτιάχνοντας ένα και μόνο τέτοιο template μπορούμε να το εφαρμόσουμε κατόπιν σε όλες τις ίδιες εγγραφές ενός XML αρχείου, μορφοποιώντας το περιεχόμενό τους αντίστοιχα.

ΜΕΡΟΣ 2^ο

Η ΕΦΑΡΜΟΓΗ MEDAPP

- Κεφάλαιο 4^ο: Εισαγωγή
- Κεφάλαιο 5^ο: Εφαρμογή
- Κεφάλαιο 6^ο: Αποτελέσματα

Κεφάλαιο 4^ο - Εισαγωγή

4.1:Εικονικό σενάριο:

4.1.1:HealthGuy General Hospital

Το υποθετικό σενάριο που έχουμε επεξεργαστεί στην παρούσα εργασία βρίσκει εφαρμογή σε νοσοκομειακά περιβάλλοντα. Ας θεωρήσουμε, λοιπόν, το γενικό νοσοκομείο HealthGuy, το οποίο σκοπεύει, στα πλαίσια της μηχανοργάνωσής του, να προσφέρει στο ιατρικό προσωπικό του υπηρεσίες παρουσίασης στοιχείων των ασθενών, μέσω ιστοσελίδας, αυστηρά εντός του ασύρματου ενδοδικτύου (intranet) που διαθέτει. Για το σκοπό αυτό, οι γιατροί, οι νοσηλευτές και το υπόλοιπο προσωπικό που έχει άμεση σχέση με την ίαση των ασθενών, πρόκειται να εξοπλιστούν με ασύρματες και κινητές συσκευές, που θα έχουν την δυνατότητα προβολής των πληροφοριών αυτών. Παράλληλα, είναι επιθυμητό το προσωπικό που διαθέτει φορητές συσκευές με δυνατότητες πλοήγησης στο διαδίκτυο, να μπορεί να έχει πρόσβαση και από αυτές τις συσκευές, στις πληροφορίες των ασθενών.

4.1.2:Λειτουργία του νοσοκομείου

Ενδεικτικά και μόνο, τέσσερις ασθενείς είναι καταγεγραμμένοι στο μητρώο του νοσοκομείου, καθ' ένας με μια συγκεκριμένη πάθηση. Είναι ασφαλισμένοι σε ασφαλιστικό φορέα, με αριθμό μητρώου μοναδικό για τον καθένα και μερικοί νοσηλεύονται ακόμα, ενώ άλλοι, μόλις πήραν εξιτήριο.

4.2:Ανάγκες:

4.2.1:Ιατροί

Το ιατρικό προσωπικό, έχει ανάγκη για άμεση και έγκυρη πληροφόρηση, σχετικά με την πορεία και την κατάσταση του κάθε ασθενή. Πρέπει να μπορεί, να έχει πρόσβαση σε έναν κατάλογο ασθενών, για τον καθέναν από τους οποίους θα πρέπει να αναφέρεται η πάθησή του, η αγωγή που ακολουθεί, η θερμοκρασία σώματός του στην παρούσα φάση της νοσηλείας του και οι τελευταίες αναλύσεις αίματος που υπεβλήθη. Επίσης, πρέπει να αναφέρεται, αν συνεχίζει να νοσηλεύεται στην κλινική ή αν πήρε εξιτήριο. Σημαντικό τέλος, θεωρείται να περιέχονται, αν υπάρχουν, πολυμεσικές πληροφορίες ιατρικού περιεχομένου, είτε εικόνες, είτε video.

4.2.2: Διοικητικοί

Το διοικητικό προσωπικό, πρέπει να μπορεί να ενημερώνεται μέσω της εφαρμογής για τον αριθμό ασφάλισης (ISN) του κάθε ασθενή, για την οικονομική κάλυψη των εξόδων νοσηλείας του, καθώς επίσης τη διάγνωση, τον τύπο της ασθένειας, αν νοσηλεύεται ακόμα και τη διάρκεια της νοσηλείας, για υπολογισμό της χρέωσης.

4.3: Ταξινόμηση τερματικού εξοπλισμού:

Για την καλύτερη οργάνωση της εφαρμογής μας και την αποδοτική διαχείριση και προσαρμογή του προσφερόμενου περιεχομένου, έχουμε προβεί σε ταξινόμηση σε κλάσεις των πιθανών τερματικών συσκευών πρόσβασης στην εφαρμογή με βάση τις δυνατότητες τους. Υπενθυμίζουμε ότι η ανίχνευση των δυνατοτήτων μιας τερματικής συσκευής γίνεται αρχικά μέσω ανάγνωσης της επικεφαλίδας user agent που περιέχεται εντός των HTTP μηνυμάτων του πελάτη και, εν συνεχεία, μέσω αναζήτησης του πεδίου αυτού εντός του wurfl αρχείου και εξαγωγής από το wurfl των δυνατοτήτων που του αντιστοιχούν. Οι βασικές κατηγορίες που έχουμε προδιαγράψει και οι απαιτήσεις που συνοδεύουν καθεμία από αυτές παρατίθενται ακολούθως:

4.3.1: Η/Υ, laptops, netbooks

Οι Η/Υ και οι φορητοί Η/Υ, είναι συσκευές που μπορούν να αποδώσουν οποιοδήποτε πολυμεσικό υλικό τους δοθεί. Λόγω των εφαρμοζόμενων τεχνικών εξοικονόμησης ενέργειας της μπαταρίας των φορητών Η/Υ, θεωρούμε ότι η προβολή στην οθόνη ενός φορητού υπολογιστή μιας ιστοσελίδας που απευθύνεται σε Η/Υ με συνεχόμενη τροφοδοσία ρεύματος (desktop), δε θα έχει αντίκτυπο στη διάρκεια ζωής της συσκευής όταν αυτή τροφοδοτείται από τη μπαταρία της.

4.3.2: PDAs και κονσόλες ψυχαγωγίας με δυνατότητα πρόσβασης στο διαδίκτυο

Οι Η/Υ χειρός, τα λεγόμενα PDAs, όπως και οι κονσόλες ψυχαγωγίας, έχουν περιορισμένες δυνατότητες σε σύγκριση με τους φορητούς Η/Υ, αλλά ικανοποιητικές για την προβολή πολυμεσικού περιεχομένου. Λόγω του ότι η ενέργεια της μπαταρίας μιας τέτοιας συσκευής, καταναλώνεται ως επί το πλείστον στην οθόνη, προσοχή δίνεται στο να μη φωτοβολεί χωρίς λόγο για πολλή ώρα. Επίσης, η παρουσίαση του πολυμεσικού περιεχομένου φροντίζουμε να μη γίνεται όλη σε μια συνεδρία, προκειμένου να μην καταναλώνει μεγάλα ποσά ενέργειας ο επεξεργαστής, για την απόδοση πολλών τέτοιων στοιχείων μονομιάς.

4.3.3:Συσκευές κινητής τηλεφωνίας

Οι χρήστες της εφαρμογής, αν επιθυμούν πρόσβαση με χρήση συσκευής κινητής τηλεφωνίας, θα πρέπει να διαθέτουν λειτουργία WiFi ή, εναλλακτικά, σύνδεση 3G ή GPRS, για πρόσβαση στο ενδοδίκτυο του νοσοκομείου. Η εφαρμογή έχει σχεδιαστεί μόνο για κινητές συσκευές με δυνατότητες αναπαραγωγής και παρουσίασης HTML κώδικα, έστω και σε βασικό επίπεδο. Ανάλογα με το επίπεδο που μια τέτοια συσκευή μπορεί να αποδώσει τα HTML στοιχεία, θα δέχεται και την ανάλογη ιστοσελίδα. Για το λόγο αυτό, η κλάση αυτή υποδιαιρείται σε τρεις επιμέρους κατηγορίες:

- Στις κινητές συσκευές με αυξημένες δυνατότητες αναπαραγωγής πολυμεσικών στοιχείων
- Στις κινητές συσκευές με περιορισμένες δυνατότητες αναπαραγωγής πολυμέσων
- Και στις κινητές συσκευές χωρίς δυνατότητες αναπαραγωγής πολυμέσων.

Οι πληροφορίες λαμβάνονται σε μορφή HTML κειμένου, με υπερσυνδέσεις για τα πολυμέσα (όπου επιτρέπονται), για τη λιγότερη δυνατή απασχόληση του επεξεργαστή και τη μεγαλύτερη οικονομία στην ενέργεια της μπαταρίας. Επίσης, προσοχή δίνεται στον χρωματισμό της ιστοσελίδας, όπως πρέπει να γίνεται σε κάθε συσκευή με μπαταρία.

4.3.4:Server με συνεχή δυνατότητα επεξεργασίας αιτήσεων

Ο εξυπηρετητής οφείλει να εξασφαλίζει την αδιάλειπτη διάθεση των πληροφοριών στους πιστοποιημένους χρήστες, μεριμνώντας για την απόδοση της σωστής ιστοσελίδας σε κάθε συσκευή, ανάλογα με τις δυνατότητές της.

4.4:Προτυποποίηση

4.4.1: Κατάταξη των συσκευών σε κλάσεις δυνατοτήτων

Η εφαρμογή διακρίνει τον τύπο της τερματικής συσκευής του εκάστοτε πελάτη και ανάλογα παρουσιάζει στον πελάτη το αιτούμενο περιεχόμενο προσαρμοσμένο στις δυνατότητες της συσκευής του. Τα κριτήρια που χρησιμοποιούμε για την κατάταξη κάθε τερματικής συσκευής σε μία από τις κλάσεις συσκευών που έχουμε ορίσει με βάση τις γενικές δυνατότητές τους, τα εξάγουμε από τη μελέτη του αρχείου του WURFL. Αναλυτικότερα, έχουμε προδιαγράψει τις εξής κλάσεις:

4.4.1.1:Κλάση H/Y και laptop

Αν ο user agent μιας συσκευής δεν εντοπιστεί στη λίστα του WURFL, υποθετούμε πως η συσκευή ανήκει στην κλάση των συσκευών τύπου H/Y ή laptop. Το WURFL αναφέρεται αποκλειστικά σε φορητές συσκευές μικρού μεγέθους, με αποτέλεσμα να μην συμπεριλαμβάνει τους user-agents των πιο διάσημων περιηγητών (IE, Firefox, Opera, Chrome) για συσκευές τύπου desktop ή laptop. Ως ετούτου, η ταυτοποίηση μιας συσκευής – user-agent αυτού του είδους, δεν είναι δυνατό να γίνει μέσω WURFL. Με δεδομένο όμως ότι εντός του WURFL περιλαμβάνονται οι περισσότερες φορητές συσκευές της αγοράς, είναι εύκολο να συνάγουμε ότι η σύμβαση που εισάγουμε εδώ (μη εύρεση στο αρχείο WURFL => H/Y ή laptop), πολύ λίγο απέχει από την πραγματικότητα.

Οι συσκευές της κλάσης H/Y – LAPTOP διακρίνονται για την πληρότητα τους σε δυνατότητες αναπαραγωγής και παρουσίασης πολυμεσικού περιεχομένου, ανεξαρτήτως του δικτυακού τύπου σύνδεσης που έχουν με τον εξυπηρετητή της νοσοκομειακής εφαρμογής μας. Είτε ασύρματα (WiFi ή 3G), είτε ενσύρματα, έχουν τις δυνατότητες να αποδώσουν το περιεχόμενο της εφαρμογής σε ικανοποιητικούς χρόνους και στις δύο περιπτώσεις.

4.4.1.2:Κλάση συσκευών τύπου PDA

Στην κλάση συσκευών τύπου PDA δεχόμαστε ότι ανήκουν οι φορητές ασύρματες συσκευές των οποίων, αφενός, ο user-agent ανιχνεύεται στο αρχείο WURFL, και αφετέρου, στο αντίστοιχο πεδίο δυνατοτήτων του WURFL ανιχνεύονται συγκεκριμένες κωδικές λέξεις οι οποίες χαρακτηρίζουν συσκευές τύπου PDA. Οι κωδικές αυτές λέξεις είναι:

- Windows CE, το λειτουργικό σύστημα της Microsoft για κινητές συσκευές και PDAs,
- HP iPAQ, ένα από τα πιο διάσημα PDA της εταιρίας HP,
- Ονομασίες φορητών παιχνιδομηχανών των εταιριών Playstation, Nintendo κ.α., με δυνατότητα πρόσβασης στο διαδίκτυο.

Βάση παρατηρήσεων στις καταγραφές συσκευών του WURFL, οι PDA συσκευές εμφανίζονται να διαθέτουν μια ανοιχτή πλατφόρμα εγκατάστασης περιηγητών. Οπότε, είναι αρκετά εύκολο, για την ώρα, να χαράξουμε τη διαχωριστική γραμμή μεταξύ των PDAs και των κινητών τηλεφώνων, με πάσα επιφύλαξη ότι στο κοντινό μέλλον αυτές οι συσκευές θα λογίζονται ως μια.

4.4.1.3: Κλάση κινητών συσκευών με πολυμεσικές δυνατότητες

Ως κινητές λογαριάζουμε όλες τις συσκευές των οποίων ο user agent εντοπίζεται εντός του αρχείου WURFL και οι οποίες δεν ανήκουν στην κλάση συσκευών PDA. Για τη διάκριση των κινητών συσκευών σε υποκλάσεις, προβαίνουμε σε συγκρίσεις συγκεκριμένων δυνατοτήτων κάθε συσκευής, όπως μας τις δίνει το WURFL, με κάποιες συμβάσεις που δεχόμαστε ότι θα πρέπει να ισχύουν, οι οποίες είναι:

- Η δυνατότητα “is_wireless_device” της ομάδος “product_info” του WURFL, είναι το πρώτο βήμα διαχωρισμού. Αναφέρει αν η συσκευή μπορεί να λειτουργήσει ασύρματα, κάτι που το επιθυμούμε και επιβάλλεται.
- Η δυνατότητα “wifi” της ομάδος “bearer”, είναι το δεύτερο σημαντικό στοιχείο που χρειαζόμαστε. Λόγω του ότι θα έχουμε ασύρματο ενδοδίκτυο στο νοσοκομείο, η συσκευή πρέπει να μπορεί να δέχεται wifi συνδέσεις. Στο σχεδιασμό της εφαρμογής μας έχουμε υιοθετήσει τη σύμβαση ότι οι ασύρματες συσκευές με δυνατότητα πρόσβασης στο διαδίκτυο τόσο μέσω ασύρματου δικτύου, όσο και μέσω 3G, πρώτα θα επιχειρούν τη σύνδεσή τους μέσω WiFi και αν αυτό δεν είναι δυνατό, μέσω 3G.
- Η δυνατότητα “xhtml_support_level” της ιδιότητας “markup”, χαρακτηρίζει την ικανότητα μιας συσκευής ως προς την απόδοση των παρεχόμενων ιστοσελίδων. Για το λόγο αυτό έχουμε ορίσει μια αριθμητική κλίμακα που ξεκινά από το -1 (εκφράζει την αδυναμία μιας συσκευής να μεταφράσει HTML κώδικα, αλλά δεν αποκλείει το ενδεχόμενο να μπορεί να μεταφράσει WML) και φτάνει μέχρι το 4 (για τις συσκευές με δυνατότητα άριστης μετάφρασης HTML κώδικα). Η κλίμακα χρησιμοποιείται από την εφαρμογή μας ως κριτήριο για την παραγωγή του περιεχομένου που στέλνεται για παρουσίαση στις αντίστοιχες συσκευές. Οι συσκευές αυτής της κατηγορίας, πρέπει να έχουν αριθμητική τιμή από 3 και άνω.
- Οι κινητές συσκευές που μπορούν να αποδώσουν σωστά τον πλήρη κώδικα της εφαρμογής απαιτείται να διαθέτουν ελάχιστο εύρος οθόνης της τάξης των 164 εικονοστοιχείων (pixels). Αν η οθόνη μιας συσκευής διαθέτει λιγότερα εικονοστοιχεία, τότε αυτόματα υποβαθμίζεται η ποιότητα της παρεχόμενης ιστοσελίδας, για καλύτερη απεικόνιση των πληροφοριών.
- Λαβάνεται υπόψη πως, για να αποδοθούν οι ιστοσελίδες που περιέχουν εικόνες, η συσκευή πρέπει να περιέχει την τιμή “true” στις ακόλουθες δυνατότητες της ομάδος “image_format”: “jpg”,

“tiff”, “gif”, “gif_animated” και “bmp”. Πρέπει δηλαδή η συσκευή, να υποστηρίζει τα παραπάνω πρότυπα ψηφιακής εικόνας.

4.4.1.4:Κλάση κινητών συσκευών με περιορισμένες πολυμεσικές δυνατότητες

Σε αυτή την κλάση κινητών υπάγονται οι συσκευές που δεν πληρούν τα επίπεδα τιμών που αναφέρθηκαν στην κατηγορία των συσκευών με πλήρεις πολυμεσικές δυνατότητες. Συγκεκριμένα, οι συμβάσεις που δεχόμαστε ότι πρέπει να ισχύουν είναι:

- Η δυνατότητα “is_wireless_device” της ομάδος “product_info” του WURFL να έχει την τιμή true.
- Η δυνατότητα “wifi” της ομάδος “bearer”, είναι το δεύτερο σημαντικό στοιχείο που χρειαζόμαστε.
- Η δυνατότητα “xhtml_support_level” της ιδιότητας “markup”, απαιτούμε να είναι τουλάχιστον επιπέδου 2. Το επίπεδο 2 από μόνο του είναι αμφιλεγόμενο για να διακρίνει κανείς τις δυνατότητες μιας συσκευής όσον αφορά τις πολυμεσικές απεικονίσεις. Όμως, σε συνδιασμό με την δυνατότητα “wifi”, που απαιτούμε να είναι “true” στα κινητά αυτής της κλάσης, συμπεραίνουμε ότι η συσκευή, για να διαθέτει τέτοιες ασύρματες δικτυακές δυνατότητες, θα μπορεί να προβάλει πολυμεσικά στοιχεία.
- Οι συσκευές αυτής της κλάσης, απαιτούμε να διαθέτουν ένα ελάχιστο εύρος οθόνης της τάξης των 96 εικονοστοιχείων (pixels).
- Εξυπακούεται, τέλος, πως για να αποδοθούν σωστά οι ιστοσελίδες που περιέχουν εικόνες, η συσκευή θα πρέπει να υποστηρίζει τα πρότυπα ψηφιακής εικόνας: “jpg”, “tiff”, “gif”, “gif_animated” και “bmp”.

4.4.1.5:Κλάση κινητών συσκευών χωρίς πολυμεσικές δυνατότητες και με περιορισμούς στην προβολή των ιστοσελίδων

Σε αυτήν την κλάση κινητών υπάγονται οι συσκευές που δεν πληρούν τις προδιαγραφές που αναφέρθηκαν στις κατηγορίες των συσκευών με πλήρεις ή περιορισμένες πολυμεσικές δυνατότητες. Πιο αναλυτικά, οι ελάχιστες απαιτήσεις που πρέπει να ικανοποιούνται ώστε να πετύχουμε την απολύτως αναγκαία προβολή των ιατρικών δεδομένων σε μορφή κειμένου, είναι:

- Η δυνατότητα “is_wireless_device” της ομάδος “product_info” του WURFL, είναι απαραίτητο αν μιλάμε για ασύρματες, κινητές συσκευές.

- Η δυνατότητα “xhtml_support_level” της ιδιότητας “markup”, απαιτείται να είναι από 0 έως 2. Όπως αναφέραμε, το επίπεδο 2 είναι αμφιλεγόμενο για να διακρίνει κανείς τις δυνατότητες μιας συσκευής όσον αφορά τις πολυμεσικές απεικονίσεις. Όμως, σε συνδιασμό με την ιδιότητα “wifi”, που στην κατηγορία αυτή θα έχει την τιμή “false”, συμπεραίνουμε ότι η συσκευή, για να μη διαθέτει ασύρματες δικτυακές δυνατότητες, δε θα μπορεί να προβάλει πολυμεσικά στοιχεία. Τα επίπεδα 1 και 0, δηλώνουν από μόνα τους, την ανικανότητα προβολής πολυμεσικών στοιχείων.
- Οι συσκευές της κατηγορίας αυτής απαιτούμε να διαθέτουν ελάχιστο εύρος οθόνης της τάξης των 64 εικονοστοιχείων (pixels). Αν η οθόνη μιας συσκευής διαθέτει λιγότερα εικονοστοιχεία, τότε δεν μπορεί να απεικονίσει σωστά καμία ιστοσελίδα.

4.4.2: Προσαρμογή περιεχομένου ανάλογα με την κλάση κάθε τερματικής συσκευής

Διαφορετικές μορφοποιήσεις του περιεχομένου έχουν προβλεφθεί για τους διάφορους τύπους τερματικών συσκευών του νοσοκομειακού προσωπικού. Διευκρινίζουμε ότι η παραγωγή της ιστοσελίδας που προωθείται στον τελικό χρήστη γίνεται δυναμικά.

4.4.2.1:Μορφή του περιεχομένου της ιστοσελίδας για τερματικές συσκευές πελάτη τύπου H/Y (desktop, laptop)

Η ιστοσελίδα που παρέχει τις πληροφορίες είναι αυστηρά ένας πίνακας που περιέχει τα εξής πεδία:

1. Το επίθετο του ασθενή.
2. Το όνομα του ασθενή.
3. Τον αριθμό μητρώου του ασφαλιστικού του φορέα.
4. Την διάγνωση της πάθησής του.
5. Την θεραπευτική του αγωγή.
6. Την τρέχουσα θερμοκρασία σώματος. Αν έχει πάρει εξιτήριο, αναφέρεται η τελευταία καταγεγραμμένη θερμοκρασία.
7. Σε περίπτωση που έχει πραγματοποιηθεί αιματοληψία και έχουμε αιματολογικά στοιχεία, να εμφανίζεται ένας υπερσύνδεσμος που να γράφει «yes» και να ανακατευθύνει στο ξεχωριστό πίνακα αιματολογικών στοιχείων, για κάθε ασθενή ξεχωριστά. Σε περίπτωση που δεν έχει πραγματοποιηθεί αιματολογική εξέταση, να εμφανίζει το μήνυμα «N/A».
8. Μια αναφορά στην τρέχουσα τιμή του αιματοκρίτη του ασθενή, σε μορφή γραφήματος, μορφή οριζόντιας μπάρας, πράσινου χρώματος, σε φόντο χρωματισμένης διαβάθμισης, που οι μπλέ περιοχές είναι τα χαμηλά όρια, οι πράσινες τα φυσιολογικά και τα

κόκκινα τα υψηλά. Αν είναι εκτός των φυσιολογικών ορίων, η μπάρα χρωματίζεται κόκκινη, για έμφαση της μη φυσιολογικής τιμής. Επίσης, να υπάρχει πρόβλεψη για διαφοροποίηση των ενδείξεων ανάλογα με το φύλο του ασθενή. Σε περίπτωση που δεν έχουμε στοιχεία, λόγω της απουσίας αιματολογικών εξετάσεων, να εμφανίζει το μήνυμα «No data available».

9. Τις συνολικές μέρες νοσηλείας.
10. Αν συνεχίζει να νοσηλεύεται.
11. Εικόνα της πάθησης αν υπάρχει, διαφορετικά, εμφανίζεται το μήνυμα «no image».
12. Video από την πάθηση, διαφορετικά, εμφανίζεται το μήνυμα «no video».

Η ιστοσελίδα έχει άσπρο φόντο και μαύρα γράμματα. Η κάθε εικόνα δεν υπερβαίνει τις 100 μονάδες κουκκίδας οθόνης πλάτος και 100 μονάδες κουκκίδας οθόνης ύψος, ώστε να χωράει στις διαστάσεις των κελιών του πίνακα. Η εικόνα είναι υπερσύνδεσμος που οδηγεί στην ίδια εικόνα, με τις πραγματικές της διαστάσεις, όμως.

Το video, έχει διαστάσεις 150 μονάδες κουκκίδας οθόνης πλάτος και 150 μονάδες κουκκίδας οθόνης ύψος. Αν το παρεχόμενο video έχει την κατάλληλη κωδικοποίηση (.ogg στην περίπτωσή μας), υπάρχει υπερσύνδεσμος πάνω από τον χώρο του video, μέσα στο κελί, που αναγράφει «Full screen» για προβολή του video σε πλήρη οθόνη. Αν το video είναι από ιστοχώρο φιλοξενίας video (π.χ. youtube), τέτοιες λειτουργίες γίνονται στην ενσωματωμένη μηχανή αναπαραγωγής του μέσου, οπότε ο υπερσύνδεσμος εκλείπει.

4.4.2.2:Μορφή του περιεχομένου της ιστοσελίδας για τερματικές συσκευές πελάτη τύπου PDA

Η ιστοσελίδα περιέχει ότι και η ιστοσελίδα για τους Η/Υ, αλλά με τις εξής διαφορές:

- Το φόντο είναι γκρι και τα γράμματα άσπρα. Αυτό, βοηθάει στην ξεκούραση του ματιού σε συρρικνωμένη οθόνη, παράλληλα με την εξοικονόμηση της ενέργειας της μπαταρίας που προκύπτει λόγω της αποφυγής του λευκού φωτισμού της οθόνης που θα συνέβαινε αν το φόντο ήταν άσπρο.
- Ο αιματοκρίτης συνεχίζει και εμφανίζεται στην ίδια μορφή γραφήματος, με την διαφορά ότι το φόντο αντικαθίσταται με τρία πεδία, που ονοματίζονται «L» για χαμηλές τιμές, «N» για τις φυσιολογικές και «H» για τις υψηλές. Η πρόβλεψη έγγυται στο ότι πολλά PDA δεν μπορούν να αποδώσουν σωστά στην χρωματική διαβάθμιση, λόγω του περιορισμού της παλέτας χρωμάτων της οθόνης, κάτι που θα μπορούσε να δημιουργήσει περιοχές ανόμοιων

χρωμάτων, που θα μπερδεύαν το ιατρικό προσωπικό με όρια φυσιολογικών και μη τιμών.

- Η εικόνα προβάλλεται, αλλά το video, αν υπάρχει, είναι ένας υπερσύνδεσμος που γράφει «yes» και οδηγεί στην προβολή του video σε πλήρη οθόνη. Αν δεν υπάρχει τέτοιο υλικό εμφανίζεται το μήνυμα «no video».

4.4.2.3:Μορφή του περιεχομένου της ιστοσελίδας για κινητές συσκευές πελάτη με πολυμεσικές δυνατότητες

Η ιστοσελίδα περιέχει ότι και η ιστοσελίδα για τους Η/Υ, αλλά με τις εξής διαφορές:

- Το φόντο είναι μαύρο και τα γράμματα άσπρα. Η αντίθεση είναι εμφανής και βοηθάει στην ανάγνωση των στοιχείων στις μικρές οθόνες των κινητών τηλεφώνων, καθώς και κάνει οικονομία στην ενέργεια της μπαταρίας της συσκευής, λόγω του ότι οι κουκκίδες οθόνης, δεν φωτοβολάνε για την απόδοση του χρώματος του φόντου.
- Οι μέγιστες διαστάσεις του πίνακα, είναι σε συμφωνία, με τη μέγιστη δυνατότητα απεικόνισης διαδικτυακών στοιχείων της συσκευής, για την επίτευξη του καλύτερου δυνατού αποτελέσματος προβολής της πληροφορίας. Ενδεχομένως, να είναι μεγαλύτερος ο πίνακας από την οθόνη της συσκευής, αλλά είναι δυνατή η περιήγηση από πανοραμική όψη (pan).
- Η αναφορά στην τρέχουσα τιμή του αιματοκρίτη του ασθενή, θα γράφεται με έναν απλό αριθμό, σε γραμματοσειρά και χρώμα όπως και το υπόλοιπο κείμενο. Αν είναι εκτός των φυσιολογικών ορίων, η τιμή αναγράφεται με κόκκινα γράμματα. Σε περίπτωση που δεν έχουμε στοιχεία, λόγω της απουσίας αιματολογικών εξετάσεων, να εμφανίζει το μήνυμα «No data available».
- Τα πολυμεσικά στοιχεία, όπου υπάρχουν, είναι υπερσύνδεσμοι που γράφουν «yes» και οδηγούν στην προβολή της εικόνας ή του video σε πλήρη οθόνη. Αν δεν υπάρχει τέτοιο υλικό, αναγράφεται η ένδειξη «no».

4.4.2.4:Μορφή του περιεχομένου της ιστοσελίδας για κινητές συσκευές πελάτη με περιορισμένες πολυμεσικές δυνατότητες

Η ιστοσελίδα περιέχει ότι και η ιστοσελίδα για τους Η/Υ, αλλά με τις εξής διαφορές:

- Το φόντο είναι μαύρο και τα γράμματα άσπρα. Η αντίθεση είναι εμφανής και βοηθάει στην ανάγνωση των στοιχείων στις μικρές οθόνες των κινητών τηλεφώνων, καθώς και κάνει οικονομία στην

ενέργεια της μπαταρίας της συσκευής, λόγω του ότι οι κουκκίδες οθόνης, δεν φωτοβολάνε για την απόδοση του χρώματος του φόντου.

- Οι μέγιστες διαστάσεις του πίνακα, είναι σε συμφωνία, με τη μέγιστη δυνατότητα απεικόνισης διαδικτυακών στοιχείων της συσκευής, για την επίτευξη του καλύτερου δυνατού αποτελέσματος προβολής της πληροφορίας. Ενδεχομένως, να είναι μεγαλύτερος ο πίνακας από την οθόνη της συσκευής, αλλά είναι δυνατή η περιήγηση από πανοραμική όψη (pan).
- Τα πολυμεσικά στοιχεία, μόνο οι εικόνες στην συγκεκριμένη περίπτωση, όπου υπάρχουν, είναι υπερσύνδεσμοι που γράφουν «yes» και οδηγούν στην προβολή της εικόνας ή του video σε πλήρη οθόνη. Αν δεν υπάρχει τέτοιο υλικό, αναγράφεται η ένδειξη «no».

4.4.2.5:Μορφή του περιεχομένου της ιστοσελίδας για κινητές συσκευές πελάτη χωρίς πολυμεσικές δυνατότητες και με περιορισμούς στην προβολή ιστοσελίδων

Η ιστοσελίδα περιέχει ότι και η ιστοσελίδα για τους Η/Υ, αλλά με τις εξής διαφορές:

- Το φόντο είναι μαύρο και τα γράμματα άσπρα. Η αντίθεση είναι εμφανής και βοηθάει στην ανάγνωση των στοιχείων στις μικρές οθόνες των κινητών τηλεφώνων, ενώ κάνει και οικονομία στην ενέργεια της μπαταρίας της συσκευής, λόγω του ότι οι κουκκίδες οθόνης δε χρειάζεται να φωτοβολάνε για την απόδοση του χρώματος του φόντου.
- Οι μέγιστες διαστάσεις του πίνακα, είναι σε συμφωνία, με τη μέγιστη δυνατότητα απεικόνισης διαδικτυακών στοιχείων της συσκευής, για την επίτευξη του καλύτερου δυνατού αποτελέσματος προβολής της πληροφορίας. Ενδεχομένως, να είναι μεγαλύτερος ο πίνακας από την οθόνη της συσκευής, αλλά είναι δυνατή η περιήγηση από πανοραμική όψη (pan).
- Τα πεδία Image και Video εκλείπουν.

4.4.3:Μορφή του περιεχομένου της ιστοσελίδας υποδοχής

Ο εξυπηρετητής, για να δεχτεί ένα αίτημα από μία συσκευή, στέλνει πρώτα μια ιστοσελίδα υποδοχής, κοινή για όλους τους χρήστες, που την κοινοποιεί σε όλους τους πελάτες (clients) που επισκέπτονται τη διεύθυνση <http://www.servername.gr/MedApp/index.jsp>. Αυτή είναι μια πολύ απλή jsp (javascript) σελίδα που περιέχει το όνομα του νοσοκομείου, μια εικόνα με το σύμβολο του νοσοκομείου και, τέλος, ένα πλήκτρο που αναγράφει «get medical stats». Το πλήκτρο αυτό ενεργοποιεί τη ρουτίνα του servlet, ώστε αυτός να ελέγξει τι τύπου πλοηγό και τι τύπου συσκευή

χρησιμοποιεί ο χρήστης, για την παράδοση του ανάλογου περιεχομένου. Το φόντο της ιστοσελίδας υποδοχής είναι μαύρο, και τα γράμματα άσπρα. Το πλήκτρο έχει γκρι χρώμα, τέτοιο ώστε να δημιουργεί αντίθεση που λειτουργεί σαν οδηγός για το τι πρέπει να πατήσει ο χρήστης.

Κεφάλαιο 5^ο - Εφαρμογή

5.1:Ζητούμενη παραγόμενη σελίδα:

5.1.1:Γενικά

Η ιστοσελίδα, δεν μπορεί να είναι στατική, αποθηκευμένη στον αποθηκευτικό χώρο του Server, γιατί διαφορετικά, θα απαιτούνταν εξειδικευμένες γνώσεις από το προσωπικό, επί του θέματος διαχείρισης και ανάπτυξης ιστοσελίδων, ώστε να ενημερώνεται κάθε φορά η σελίδα με τις πληροφορίες των ασθενών.

Το ζητούμενο είναι, να χρησιμοποιηθεί ένας τρόπος εισαγωγής δεδομένων σε ένα σύστημα τήρησης αρχείου ασθενών που να είναι εύχρηστος και κατανοητός από όλο το προσωπικό, ανεξαρτήτως της ευχέρειας και των δεξιοτήτων που κάποιοι κατέχουν επί της χρήσης των Η/Υ. Ο τρόπος αυτός, ακολούθως, θα αναλαμβάνει αυτόματα την παραγωγή της ιστοσελίδας με τις πληροφορίες των ασθενών, χωρίς να παρεμβάλλεται ανθρώπινος παράγοντας, για την αποφυγή αλλοίωσης του κώδικα της ιστοσελίδας και των πληροφοριών των ασθενών από αμέλεια και λάθη χρηστών.

5.1.2:Γιατί XML, XSLT

Η οργάνωση των δεδομένων σε XML προτιμάται, γιατί πολλές εφαρμογές διαχείρισης βάσης δεδομένων μπορούν να εξάγουν τέτοια αρχεία, με τις πληροφορίες των πινάκων που περιέχουν, σε απλή και κατανοητή μορφή.

Επίσης, η συγγραφή τέτοιου αρχείου, με την χρήση ενός βασικού επεξεργαστή κειμένου, είναι εύκολη υπόθεση, λόγω του ελεύθερου να θέσει ο χρήστης, ότι ονόματα κατηγοριών ταξινόμησης θελήσει.

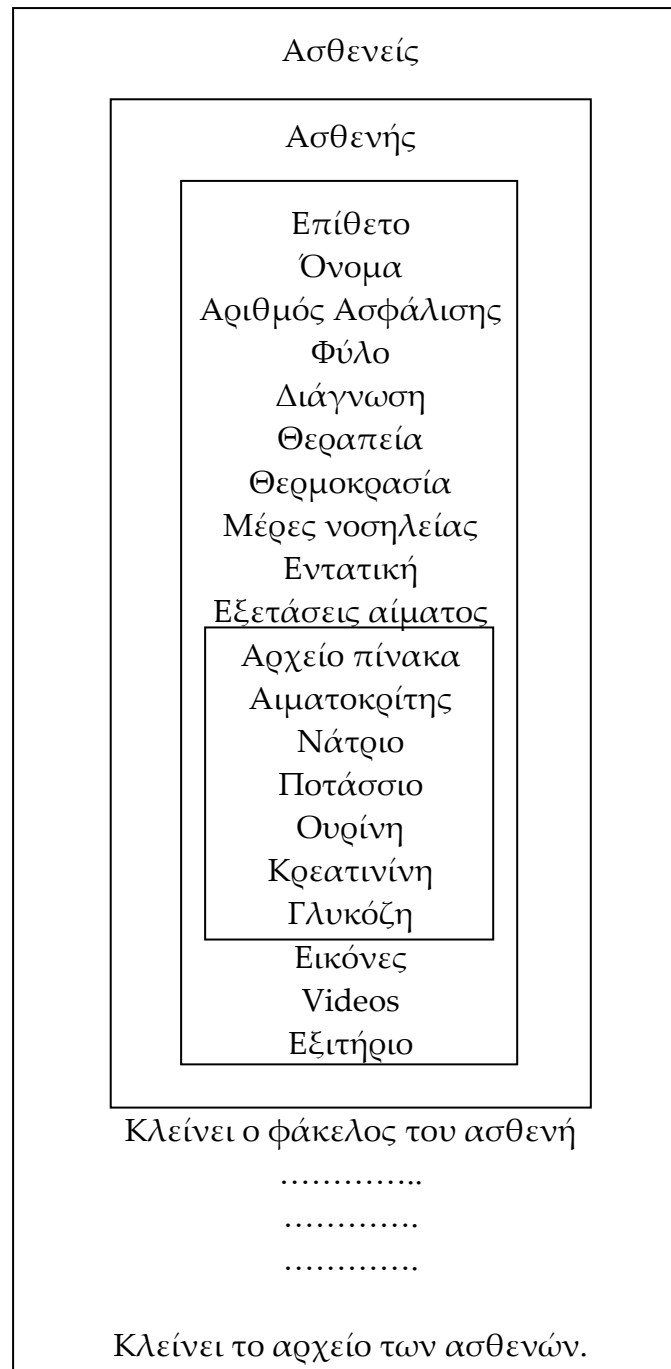
Στην παραγωγή ιστοσελίδων του παραδείγματός μας, τα XSLTs αποδεικνύονται πανίσχυρα, στο ζήτημα της επιλεκτικής προβολής πληροφοριών. Ο συνδυασμός της ευκολίας παραγωγής του XML αρχείου, σε συνδυασμό με την ευχρηστία των XSLTs, ουσιαστικά αποτελεί μονόδρομο στην επιλογή αυτής της μεθόδου εργασίας για την υλοποίηση της εφαρμογής.

5.2:XML ασθενών:

5.2.1:Οργάνωση

Εκτός από τις πληροφορίες που αναφέρθηκαν στην προτυποποίηση, επιπλέον πληροφορίες θα πρέπει να μπορούν να εισάγονται σε τέτοιου είδους αρχεία, για πρόληψη των μελλοντικών αναγκών της μηχανοργάνωσης και του νοσοκομείου. Έτσι, σε πρώτη φάση, εισάγουμε και ένα επιπλέον στοιχείο, αυτό του αν ο ασθενής βρίσκεται στη Μ.Ε.Θ. (Μονάδα Εντατικής Θεραπείας), το οποίο προσδιορίζεται με ένα ναι (yes) ή όχι (no), στο σχετικό πεδίο ER του πίνακα ασθενών.

Η γενικότερη δομή που ακολουθούμε στην οργάνωση του XML αρχείου, απεικονίζεται στο Σχήμα 10, στην επόμενη σελίδα.



Σχήμα 10. Οργάνωση του αρχείου patients.xml

5.2.2:Κώδικας του XML

Παρατίθεται ο κώδικας, με τις πληροφορίες των ασθενών:

```
<?xml version="1.0" encoding="utf-8"?>
<patients>
  <patient name="Annah" surname="Lirdo" ISN="769215">
    <gender>female</gender>
    <diagnoses>gastroenteritis</diagnoses>
    <treatment>antibiotics and diet</treatment>
    <temperature>38C</temperature>
    <days>3</days>
    <ER>no</ER>
    <bloodsample>
      <file>BSLirdo.html</file>
      <hematocrit>38.2</hematocrit>
      <Na>136</Na>
      <K>2.7</K>
      <Urea>22</Urea>
      <Creatinine>1.8</Creatinine>
      <Glucose>5.8</Glucose>
    </bloodsample>
    <images></images>
    <video></video>
    <nursing>no</nursing>
  </patient>
  <patient name="Fred" surname="Wilson" ISN="784457">
    <gender>male</gender>
    <diagnoses>swine flu</diagnoses>
    <treatment>antibiotics, natural serum, feeding
serum</treatment>
    <temperature>42C</temperature>
    <days>3</days>
    <ER>yes</ER>
    <bloodsample>
      <file>BSWilson.html</file>
      <hematocrit>27.9</hematocrit>
      <Na>112</Na>
      <K>1.3</K>
      <Urea>23</Urea>
      <Creatinine>0.5</Creatinine>
      <Glucose>1.5</Glucose>
    </bloodsample>
    <images>fred.bmp</images>
    <video></video>
    <nursing>yes</nursing>
  </patient>
  <patient name="Kith" surname="Raxon" ISN="788214">
    <gender>male</gender>
    <diagnoses>herpes</diagnoses>
    <treatment>antibiotics</treatment>
    <temperature>36,6C</temperature>
    <days>1</days>
    <ER>no</ER>
    <bloodsample>
      <file>BSRaxon.html</file>
      <hematocrit>45.1</hematocrit>
      <Na>142</Na>
      <K>3.8</K>
      <Urea>18</Urea>
      <Creatinine>1.2</Creatinine>
    </bloodsample>
  </patient>
</patients>
```

```
        <Glucose>3.8</Glucose>
    </bloodsample>
    <images>herpes.jpg</images>

    <video>http://www.youtube.com/watch?v=2CfNVKvMLGM</video>
    <nursing>yes</nursing>
</patient>
<patient name="Clarabell" surname="Freeman" ISN="788214">
    <gender>female</gender>
    <diagnoses>pregnant</diagnoses>
    <treatment>Ceasar section, antibiotics</treatment>
    <temperature>38C</temperature>
    <days>5</days>
    <ER>no</ER>
    <bloodsample>
        <file>BSFreeman.html</file>
        <hematocrit>40.0</hematocrit>
        <Na>148</Na>
        <K>5.9</K>
        <Urea>25</Urea>
        <Creatinine>0.5</Creatinine>
        <Glucose>4.2</Glucose>
    </bloodsample>
    <images>pregnant.jpg</images>
    <video>pregnant.ogg</video>
    <nursing>yes</nursing>
</patient>
<patient name="John" surname="Gergg" ISN="789429">
    <gender>male</gender>
    <diagnoses>Hypertrichosis</diagnoses>
    <treatment>haircut</treatment>
    <temperature>36C</temperature>
    <days>1</days>
    <ER>no</ER>
    <bloodsample>
        <file></file>
        <hematocrit></hematocrit>
    </bloodsample>
    <images>john.jpg</images>
    <video>hair.ogg</video>
    <nursing>no</nursing>
</patient>
</patients>
```

5.3:Εισαγωγική ιστοσελίδα:

5.3.1:Κώδικας

Παρακάτω, παρατίθεται ο κώδικας HTML του αρχείου JSP της εισαγωγικής ιστοσελίδας:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "HTTP://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta HTTP-equiv="Content-Type" content="text/html;
charset=utf-8">
```

```
<title>HealthGuy General Hospital</title>
<style type="text/css">
<!--
body,td,th {
    color: #FFFFFF;
}
body {
    background-color: #000000;
}
-->
</style></head>
<body>
    <p align="center">Welcome to the HealthGuy General Hospital
real-time services.</p>
    <p align="center"></p>
<form action="MedServlet" method="POST">
    <div align="center">
        <input value="Get medical stats" type="submit">
    </div>
</form>
</body>
</html>
```

5.3.2: Παρατηρήσεις

Λόγω του ότι η ιστοσελίδα υποδοχής είναι κοινή για όλους τους χρήστες της εφαρμογής, ανεξαρτήτως των δυνατοτήτων της συσκευής που διαθέτουν, υπάρχουν αναφορές στην επικεφαλίδα της ιστοσελίδας, όπως και στο κυρίως θέμα, σχετικά με το τι προσφέρει αυτή. Η συγκεκριμένη ιστοσελίδα, περιέχει επίσης μια εικόνα με το έμβλημα του νοσοκομείου, για άμεση οπτικοποίηση της υπηρεσίας, σε περίπτωση που μπορεί η τερματική συσκευή να αποδώσει εικόνες, καθώς και το πλήκτρο ενεργοποίησης του Servlet για λήψη των στοιχείων των ασθενών σε μορφή πίνακα.

5.3.3: Επεξηγήσεις

Το πλήκτρο περιέχεται μέσα στο tag «input» της HTML, που με τη σειρά του περιέχεται μέσα σε μια φόρμα ελέγχου για κοινοποίηση ενός POST αιτήματος. Η αναγνώριση μιας δράσης πάνω στο πλήκτρο γίνεται με τη βοήθεια της ρουτίνας MedServlet.

Η ιστοσελίδα περιέχει μόνο το πλήκτρο που παρέχει πρόσβαση στις πληροφορίες των ασθενών. Στα πλαίσια της υλοποίησης μιας εισαγωγικής ιστοσελίδας ενός πραγματικού νοσοκομείου, φυσικό είναι να περιέχονται και άλλες πληροφορίες και υπηρεσίες που αφορούν τη λειτουργία αυτού. Εντούτοις, η παρούσα εργασία εστιάζει μόνο στην υλοποίηση μιας επιλεκτικής, δυναμικής παρουσίασης περιεχομένου ανάλογα με τις δυνατότητες των τερματικών συσκευών των χρηστών. Οπότε, δεν μπορεί να χαρακτηριστεί η παρούσα ιστοσελίδα υποδοχής, σε καμιά των περιπτώσεων, ως η κεντρική ιστοσελίδα υποδοχής, ενός πραγματικού, πλήρως λειτουργικού, νοσοκομείου.

5.4:XSLT για συσκευές τύπου Η/Υ:

5.4.1:Κώδικας

Παρακάτω, παρατίθεται ο κώδικας του XSLT στην περίπτωση που η τερματική συσκευή πελάτη είναι τύπου Η/Υ:

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="utf-8" doctype-public="-
//W3C//DTD XHTML 1.0 Transitional//EN" doctype-
system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"/>
<xsl:template match="/">
<table border="1">
  <tr>
    <th scope="col">Name</th>
    <th scope="col">Surname</th>
    <th scope="col">ISN</th>
    <th scope="col">Diagnoses</th>
    <th scope="col">Treatment</th>
    <th scope="col">C&#186;</th>
    <th scope="col">Blood<br/>sample</th>
    <th scope="col">Hematocrit</th>
    <th scope="col">Days<br /> nursing</th>
    <th scope="col">Still<br/> nursing?</th>
    <th scope="col">Image</th>
    <th scope="col">Video</th>
  </tr>
  <xsl:for-each select="patients/patient">
    <tr>
      <td width="10%" align="center"><xsl:value-of
select="@name"/></td>
      <td width="10%" align="center"><xsl:value-of
select="@surname"/></td>
      <td width="10%" align="center"><xsl:value-of select="@ISN"/></td>
      <td width="10%" align="center"><xsl:value-of
select="diagnoses"/></td>
      <td width="10%" align="center"><xsl:value-of
select="treatment"/></td>
      <td width="5%" align="center"><xsl:value-of
select="temperature"/></td>
      <xsl:if test="bloodsample/file!=''">
        <td width="5%" align="center"><a
href="{bloodsample/file}">yes</a></td>
      </xsl:if>
      <xsl:if test="bloodsample/file=''">
        <td width="5%" align="center">N/A</td>
      </xsl:if>
      <xsl:if test="bloodsample/hematocrit=''">
        <td width="10%" align="center">No data<br/>available.</td>
      </xsl:if>
      <xsl:if test="gender='male' and
(bloodsample/hematocrit>='50.4' or
bloodsample/hematocrit<='40.6')">
        <td width="10%" align="left" background="HPCM.bmp">
          <table width="{bloodsample/hematocrit}%"
bgcolor="#FF0000"><tr><td></td></tr></table>
        </td>
      </xsl:if>
    </tr>
  </xsl:for-each>
</table>
</template>
</xsl:stylesheet>
```

```

</xsl:if>
  <xsl:if test="gender='female' and
(bloodsample/hematocrit>='44.4' or
bloodsample/hematocrit<='36.0') ">
    <td width="10%" align="left" background="HPCF.bmp">
      <table width="{bloodsample/hematocrit}%"
bgcolor="#FF0000"><tr><td></td></tr></table>
    </td>
  </xsl:if>
  <xsl:if test="gender='male' and
(bloodsample/hematocrit<='50.3' and
bloodsample/hematocrit>='40.7') ">
    <td width="10%" align="left" background="HPCM.bmp">
      <table width="{bloodsample/hematocrit}%"
bgcolor="#00FF00"><tr><td></td></tr></table>
    </td>
  </xsl:if>
  <xsl:if test="gender='female' and
(bloodsample/hematocrit<='44.3' and
bloodsample/hematocrit>='36.1') ">
    <td width="10%" align="left" background="HPCF.bmp">
      <table width="{bloodsample/hematocrit}%"
bgcolor="#00FF00"><tr><td></td></tr></table>
    </td>
  </xsl:if>
  <td width="10%" align="center"><xsl:value-of select="days"/></td>
  <td width="10%" align="center"><xsl:value-of
select="nursing"/></td>
  <td><a href="{images}"></a></td>
  <xsl:if test="video='http://www.youtube.com/watch?v=2CfNVKvMLGM'">
  <td><object width="150" height="150">
  <param name="movie" value="{video}"></param>
  <param name="allowFullScreen" value="true"></param>
  <param name="allowsriptaccess" value="always"></param>
  <embed src="{video}" type="application/x-shockwave-flash"
allowsriptaccess="always" allowfullscreen="true" width="150"
height="150">
  </embed>
  </object></td>
  </xsl:if>
  <xsl:if test="video!='' and
video!='http://www.youtube.com/watch?v=2CfNVKvMLGM'">
  <td><a href="{video}">Full screen</a><video src="{video}"
controls="controls" width="150" height="150">
  No video in this browser!
  </video><br/></td>
  </xsl:if>
  <xsl:if test="video=''">
  <td><p align='center'>no video</p></td>
  </xsl:if>
</tr>
</xsl:for-each>
</table>
</xsl:template>
</xsl:stylesheet>

```

5.4.2: Παρατηρήσεις

Οι πληροφορίες συσχέτισης με το XML των ασθενών δεν απαιτούνται, διότι ο Servlet αναλαμβάνει να το κάνει αυτό.

Για εμφάνιση στον πίνακα ασθενών επιλέγουμε κάθε στοιχείο φωλιασμένο μέσα στο γονικό, κεντρικό στοιχείο Patients του XML, και εξάγουμε τα δεδομένα στις αντίστοιχες θέσεις, είτε με το όνομά τους μέσα στο αρχείο αυτό, είτε βάζοντας το σύμβολο @ για τα στοιχεία που είναι επεξηγήσεις του στοιχείου patient. Αυτό γίνεται με χρήση του tag <xsl:for each>.

Για το αν υπάρχει ή όχι, πίνακας αιματολογικών εξετάσεων του ασθενή, ώστε να βγεί και η ανάλογη ανακατεύθυνση ή μήνυμα, υπεύθυνος είναι ο βρόγχος ελέγχου <xsl:if>. Αυτή η εντολή, διαβάζει τα δεδομένα στο XML αρχείο, στην περιοχή της τιμής <file> της υποομάδας <bloodsample>, και αν υπάρχει κενό, μας απαντάει με το μήνυμα «no», όπως ορίζει η προτυποποίηση. Διαφορετικά, εμφανίζει έναν υπερσύνδεσμο, με μήνυμα «yes», που μας ανακατευθύνει στον πίνακα αιματολογικών εξετάσεων του κάθε ασθενή.

Στη περιοχή των video, παρατηρούμε παρόμοιους βρόχους ελέγχου <xsl:if>, καθώς και μερικούς ακόμα, φωλιασμένους βρόχους. Αυτοί αναλαμβάνουν να αναγνωρίσουν αν περιέχεται ένα video στις πληροφορίες του χρήστη, τι τύπου είναι, και να συμπεριλάβουν έναν υπερσύνδεσμο για πλήρη οθόνη σε περίπτωση που το video είναι σε μορφή OGG.

Για την ένδειξη του αιματοκρίτη του ασθενή, χρησιμοποιούμε την τιμή που έχει το πεδίο hematocrit του αρχείου XML, ως πλάτος ενός πίνακα που εμπεριέχεται στο κελί της στήλης του αιματοκρίτη, καθώς και με κατάλληλο χρώμα στο προσκήνιο του πίνακα. Στο παρασκήνιο του κελιού, τοποθετούμε μια εικόνα με βαθμωτά χρώματα, πράσινο για φυσιολογικά όρια, μπλέ για χαμηλά και κόκκινα για υψηλά. Παράλληλα, αν ο ασθενής είναι άντρας ή γυναίκα, οι εικόνες διαφέρουν ως προς την βαθμονόμισή τους, ακριβώς γιατί είναι διαφορετικά τα φυσιολογικά όρια στα δύο φύλα. Τέλος, για ακόμα καλύτερη ένδειξη της καταστάσεως του ασθενή, αν τα όρια του αιματοκρίτη είναι εκτός των φυσιολογικών, το χρώμα του προσκηνίου του μεταβαλλόμενου πίνακα είναι κόκκινο, αν όμως είναι εντός των φυσιολογικών ορίων, τότε είναι πράσινο.

5.5:XSLT για συσκευές τύπου PDA:

5.5.1:Κώδικας

Παρακάτω, παρατίθεται ο κώδικας του XSLT στην περίπτωση που η τερματική συσκευή πελάτη είναι τύπου PDA:

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="utf-8" doctype-public="-
//W3C//DTD XHTML 1.0 Transitional//EN" doctype-
system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"/>
```



```

<xsl:template match="/">
<table border="1">
  <tr>
    <th scope="col">Name</th>
    <th scope="col">Surname</th>
    <th scope="col">ISN</th>
    <th scope="col">Diagnoses</th>
    <th scope="col">Treatment</th>
    <th scope="col">Temperature</th>
    <th scope="col">Blood<br/>sample</th>
    <th scope="col">Hematocrit</th>
    <th scope="col">Days<br/>nursing</th>
    <th scope="col">Still<br/>nursing?</th>
    <th scope="col">Image</th>
    <th scope="col">Video</th>
  </tr>
  <xsl:for-each select="patients/patient">
    <tr>
      <td width="10%"><xsl:value-of select="@name"/></td>
      <td width="10%" align="center"><xsl:value-of
select="@surname"/></td>
      <td width="10%" align="center"><xsl:value-of select="@ISN"/></td>
      <td width="10%" align="center"><xsl:value-of
select="diagnoses"/></td>
      <td width="10%" align="center"><xsl:value-of
select="treatment"/></td>
      <td width="10%" align="center"><xsl:value-of
select="temperature"/></td>
      <xsl:if test="bloodsample/file!=''">
        <td width="5%" align="center"><a
href="{bloodsample/file}">yes</a></td>
      </xsl:if>
      <xsl:if test="bloodsample/file=''">
        <td width="5%" align="center">N/A</td>
      </xsl:if>
      <xsl:if test="bloodsample/hematocrit=''">
        <td width="10%" align="center">No data<br/>avaivable.</td>
      </xsl:if>
      <xsl:if test="gender='male' and
(bloodsample/hematocrit>='50.4' or
bloodsample/hematocrit<='40.6') ">
        <td width="10%" align="left" background="HPDAM.bmp">
          <table width="{bloodsample/hematocrit}%"
bgcolor="#FF0000"><tr><td></td></tr></table>
        </td>
      </xsl:if>
      <xsl:if test="gender='female' and
(bloodsample/hematocrit>='44.4' or
bloodsample/hematocrit<='36.0') ">
        <td width="10%" align="left" background="HPDAF.bmp">
          <table width="{bloodsample/hematocrit}%"
bgcolor="#FF0000"><tr><td></td></tr></table>
        </td>
      </xsl:if>
      <xsl:if test="gender='male' and
(bloodsample/hematocrit<='50.3' and
bloodsample/hematocrit>='40.7') ">
        <td width="10%" align="left" background="HPDAM.bmp">
          <table width="{bloodsample/hematocrit}%"
bgcolor="#00FF00"><tr><td></td></tr></table>
        </td>
      </xsl:if>
    </tr>
  </xsl:for-each>
</table>

```

```

        </xsl:if>
        <xsl:if test="gender='female' and
        (bloodsample/hematocrit<='44.3' and
        bloodsample/hematocrit>='36.1') ">
        <td width="10%" align="left" background="HPDAF.bmp">
        <table width="{bloodsample/hematocrit}%"
        bgcolor="#00FF00"><tr><td></td></tr></table>
        </td>
        </xsl:if>
        <td width="10%" align="center"><xsl:value-of select="days"/></td>
        <td width="10%" align="center"><xsl:value-of
        select="nursing"/></td>
        <td><a href="{images}"></a></td>
        <!--Check if a video is available-->
        <xsl:if test="video!=''">
        <td><p align="center"><a href="{video}">yes</a></p></td>
        </xsl:if>
        <xsl:if test="video=''">
        <td><p align="center">no</p></td>
        </xsl:if>
        </tr>
    </xsl:for-each>
</table>
</xsl:template>
</xsl:stylesheet>

```

5.5.2: Παρατηρήσεις

Όπως και με τον κώδικα για τους Η/Υ, έτσι και εδώ, έχουμε παρόμοιο κώδικα, με την διαφορά ότι, στα video, αν υπάρχουν, ο βρόγχος ελέγχου, θα προβάλλει έναν υπεσύνδεσμο για αναπαραγωγή σε πλήρη οθόνη.

Επίσης, η εικόνα του παρασκηνίου στα κελιά της στήλης του αιματοκρίτη, αλλάζει από βαθμωτά χρώματα, σε απλές γραμμές με τρία πεδία, τα χαμηλά (L), τα φυσιολογικά (N) και τα υψηλά (H). Η πρόβλεψη γίνεται λόγω του περιορισμού πολλές φορές του πλήθους χρωμάτων σε τέτοιες συσκευές, προς αποφυγήν συγχίσεως στο τελικό αποτέλεσμα.

5.6: XSLT για κινητά με δυνατότητες πολυμέσων:

5.6.1: Κώδικας

Παρακάτω, παρατίθεται ο κώδικας του XSLT στην περίπτωση που ανιχνεύουμε ότι η τερματική συσκευή πελάτη ανήκει στην κατηγορία των εξελιγμένων κινητών τηλεφώνων:

```

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="utf-8" doctype-public="-
//WAPFORUM//DTD XHTML Mobile 1.0//EN" doctype-
system="http://www.wapforum.org/DTD/xhtml-mobile10.dtd"/>
<xsl:template match="/">

```

```

<table border="1">
  <tr>
    <th scope="col">Name</th>
    <th scope="col">Surname</th>
    <th scope="col">ISN</th>
    <th scope="col">Diagnoses</th>
    <th scope="col">Treatment</th>
    <th scope="col">Temperature</th>
    <th scope="col">Blood<br/>sample</th>
    <th scope="col">Hematocrit</th>
    <th scope="col">Days<br /> nursing</th>
    <th scope="col">Still<br/> nursing?</th>
    <th scope="col">Image</th>
    <th scope="col">Video</th>
  </tr>
  <xsl:for-each select="patients/patient">
    <tr>
      <td><xsl:value-of select="@name"/></td>
      <td align="center"><xsl:value-of select="@surname"/></td>
      <td align="center"><xsl:value-of select="@ISN"/></td>
      <td width="20%" align="center"><xsl:value-of
select="diagnoses"/></td>
      <td width="20%" align="center"><xsl:value-of
select="treatment"/></td>
      <td width="8%" align="center"><xsl:value-of
select="temperature"/></td>
      <xsl:if test="bloodsample/file!=''">
      <td width="5%" align="center"><a
href="{bloodsample/file}">yes</a></td>
      </xsl:if>
      <xsl:if test="bloodsample/file=''">
      <td width="5%" align="center">N/A</td>
      </xsl:if>
      <xsl:if test="bloodsample/hematocrit=''">
      <td width="10%" align="center">No data<br/>available.</td>
      </xsl:if>
      <xsl:if test="(gender='male' and
(bloodsample/hematocrit>='50.4' or
bloodsample/hematocrit<='40.6')) or (gender='female' and
(bloodsample/hematocrit>='44.4' or
bloodsample/hematocrit<='36.0'))">
      <td width="10%" align="center"><font color="red"><xsl:value-
of select="bloodsample/hematocrit"/></font></td>
      </xsl:if>
      <xsl:if test="(gender='male' and
(bloodsample/hematocrit<='50.3' and
bloodsample/hematocrit>='40.7')) or (gender='female' and
(bloodsample/hematocrit<='44.3' and
bloodsample/hematocrit>='36.1'))">
      <td width="10%" align="center"><xsl:value-of
select="bloodsample/hematocrit"/></td>
      </xsl:if>
      <td width="5%" align="center"><xsl:value-of select="days"/></td>
      <td width="5%" align="center"><xsl:value-of
select="nursing"/></td>
      <!-- check if an image is available -->
      <xsl:if test="images!=''">
      <td><p align="center"><a href="{images}">yes</a></p></td>
      </xsl:if>
      <xsl:if test="images=''">

```

```

<td><p align='center'>no</p></td>
</xsl:if>
<!--Check if a video is available-->
<xsl:if test="video!=''">
<td><p align="center"><a href="{video}">yes</a></p></td>
</xsl:if>
<xsl:if test="video=''">
<td><p align="center">no</p></td>
</xsl:if>
</tr>
</xsl:for-each>
</table>

</xsl:template>
</xsl:stylesheet>

```

5.6.2: Παρατηρήσεις

Ο κώδικας είναι παρόμοιος με τους προηγούμενους, με τη διαφορά πως για το πολυμεσικό περιεχόμενο ακολουθείται η τακτική που υιοθετήθηκε και στις συσκευές PDA: βρόγχοι ελέγχου για την παραγωγή υπερσυνδέσμων, μέσω των οποίων γίνεται προβολή του πολυμεσικού περιεχομένου σε πλήρη οθόνη.

Στην προσπάθεια απόδοσης της πληροφορίας του αιματοκρίτη κάθε ασθενούς, αντί για την απεικόνιση μέσω γραφήματος, που σε τέτοιες μικρές οθόνες δεν προσφέρει ευκολία ανάγνωσης, το πεδίο απλά περιέχει την τιμή του αιματοκρίτη, με άσπρα γράμματα για φυσιολογικές τιμές και με κόκκινα για τιμές εκτός των φυσιολογικών ορίων.

5.7: XSLT για κινητά με περιορισμένες δυνατότητες πολυμέσων:

5.7.1: Κώδικας

Παρακάτω, παρατίθεται ο κώδικας του XSLT στην περίπτωση που ανιχνεύουμε ότι η τερματική συσκευή πελάτη είναι κινητό τηλέφωνο με περιορισμένες δυνατότητες αναπαραγωγής πολυμέσων:

```

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="utf-8" doctype-public="-
//WAPFORUM//DTD XHTML Mobile 1.0//EN" doctype-
system="http://www.wapforum.org/DTD/xhtml-mobile10.dtd"/>
<xsl:template match="/">

<table border="1">
<tr>
<th scope="col">Name</th>
<th scope="col">Surname</th>
<th scope="col">ISN</th>
<th scope="col">Diagnoses</th>

```

```

<th scope="col">Treatment</th>
<th scope="col">Temperature</th>
<th scope="col">Blood<br/>sample</th>
<th scope="col">Hematocrit</th>
<th scope="col">Days<br /> nursing</th>
<th scope="col">Still<br/> nursing?</th>
<th scope="col">Image</th>
</tr>
<xsl:for-each select="patients/patient">
<tr>
<td><xsl:value-of select="@name"/></td>
<td align="center"><xsl:value-of select="@surname"/></td>
<td align="center"><xsl:value-of select="@ISN"/></td>
<td width="20%" align="center"><xsl:value-of
select="diagnoses"/></td>
<td width="20%" align="center"><xsl:value-of
select="treatment"/></td>
<td width="8%" align="center"><xsl:value-of
select="temperature"/></td>
<xsl:if test="bloodsample/file!=''">
<td width="5%" align="center"><a
href="{bloodsample/file}">yes</a></td>
</xsl:if>
<xsl:if test="bloodsample/file=''">
<td width="5%" align="center">N/A</td>
</xsl:if>
<xsl:if test="bloodsample/hematocrit=''">
<td width="10%" align="center">No data<br/>avaiable.</td>
</xsl:if>
<xsl:if test="(gender='male' and
(bloodsample/hematocrit>='50.4' or
bloodsample/hematocrit<='40.6')) or (gender='female' and
(bloodsample/hematocrit>='44.4' or
bloodsample/hematocrit<='36.0'))">
<td width="10%" align="center"><font color="red"><xsl:value-
of select="bloodsample/hematocrit"/></font></td>
</xsl:if>
<xsl:if test="(gender='male' and
(bloodsample/hematocrit<='50.3' and
bloodsample/hematocrit>='40.7')) or (gender='female' and
(bloodsample/hematocrit<='44.3' and
bloodsample/hematocrit>='36.1'))">
<td width="10%" align="center"><xsl:value-of
select="bloodsample/hematocrit"/></td>
</xsl:if>
<td width="5%" align="center"><xsl:value-of select="days"/></td>
<td width="5%" align="center"><xsl:value-of
select="nursing"/></td>
<!-- check if an image is available -->
<xsl:if test="images!=''">
<td><p align='center'><a href="{images}">yes</a></p></td>
</xsl:if>
<xsl:if test="images=''">
<td><p align='center'>no</p></td>
</xsl:if>
</tr>
</xsl:for-each>
</table>
</xsl:template>

```

```
</xsl:stylesheet>
```

5.7.2: Παρατηρήσεις

Ο κώδικας είναι παρόμοιος με τους προηγούμενους, με τη διαφορά πως τώρα μόνο εικόνες προβάλλονται, σε πλήρη οθόνη, και αυτές κατόπιν αιτήσεως μέσω του σχετικού υπερσυνδέσμου.

5.8: XSLT για απλά κινητά:

5.8.1: Κώδικας

Παρακάτω, παρατίθεται ο κώδικας του XSLT στην περίπτωση που ανιχνεύουμε ότι η τερματική συσκευή πελάτη είναι κινητό τηλέφωνο, χωρίς ή με πολύ περιορισμένες δυνατότητες πολυμέσων, τόσο που να καθίσταται προβληματική η παρουσίαση αυτών:

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="utf-8" doctype-public="-
//WAPFORUM//DTD XHTML Mobile 1.0//EN" doctype-
system="http://www.wapforum.org/DTD/xhtml-mobile10.dtd"/>
<xsl:template match="/">
<table border="0" width="96">
<tr>
<th scope="col">Name</th>
<th scope="col">Surname</th>
<th scope="col">ISN</th>
<th scope="col">Diagnoses</th>
<th scope="col">Treatment</th>
<th scope="col">Temperature</th>
<th scope="col">Blood<br/>sample</th>
<th scope="col">Hematocrit</th>
<th scope="col">Days<br /> nursing</th>
<th scope="col">Still<br/> nursing?</th>
</tr>
<xsl:for-each select="patients/patient">
<tr>
<td><xsl:value-of select="@name"/></td>
<td align="center"><xsl:value-of select="@surname"/></td>
<td align="center"><xsl:value-of select="@ISN"/></td>
<td width="20%" align="center"><xsl:value-of
select="diagnoses"/></td>
<td width="20%" align="center"><xsl:value-of
select="treatment"/></td>
<td width="8%" align="center"><xsl:value-of
select="temperature"/></td>
<xsl:if test="bloodsample/file!=''">
<td width="5%" align="center"><a
href="{bloodsample/file}">yes</a></td>
</xsl:if>
<xsl:if test="bloodsample/file=''">
<td width="5%" align="center">N/A</td>
</xsl:if>
<xsl:if test="bloodsample/hematocrit=''">
<td width="10%" align="center">No data<br/>avaiable.</td>
</xsl:if>
```

```

    <xsl:if test="(gender='male' and
(bloodsample/hematocrit>='50.4' or
bloodsample/hematocrit<='40.6')) or (gender='female' and
(bloodsample/hematocrit>='44.4' or
bloodsample/hematocrit<='36.0'))">
      <td width="10%" align="center"><font color="red"><xsl:value-
of select="bloodsample/hematocrit"/></font></td>
    </xsl:if>
    <xsl:if test="(gender='male' and
(bloodsample/hematocrit<='50.3' and
bloodsample/hematocrit>='40.7')) or (gender='female' and
(bloodsample/hematocrit<='44.3' and
bloodsample/hematocrit>='36.1'))">
      <td width="10%" align="center"><xsl:value-of
select="bloodsample/hematocrit"/></td>
    </xsl:if>
    <td width="5%" align="center"><xsl:value-of select="days"/></td>
    <td width="5%" align="center"><xsl:value-of
select="nursing"/></td>
  </tr>
</xsl:for-each>
</table>
</xsl:template>
</xsl:stylesheet>

```

5.8.2: Παρατηρήσεις

Ο κώδικας είναι παρόμοιος με τους προηγούμενους, με τη διαφορά πως δεν προβάλλονται καθόλου πολυμέσα και ο πίνακας δεν περιέχει διαχωριστικά για καλύτερη οργάνωση σε τέτοια κινητά.

5.9: XSLT για αιματολογικό πίνακα:

5.9.1: Κώδικας

Παρακάτω, παρατίθεται ο κώδικας του XSLT για τον αιματολογικό πίνακα, που δημιουργείται ξεχωριστά για κάθε ασθενή, αρκεί να υπάρχουν στοιχεία στην υποομάδα <bloodsample>:

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="utf-8" doctype-public="-
//W3C//DTD XHTML 1.0 Transitional//EN" doctype-
system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"/>
<xsl:template match="/">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>Blood Sample analysis</title>
</head>

<body bgcolor="#000000" text="#FFFFFF">
<xsl:for-each select="patients/patient">
<xsl:if test="@ISN='769215'">

```

```

<table border="1">
  <tr>
    <td colspan="2" align="center">Patient's Name: <xsl:value-of
select="@name"/>&#160;<xsl:value-of select="@surname"/></td></tr>
  <!--HCT-->
  <tr>
    <th>HCT</th>
    <xsl:if test="(gender='male' and
(bloodsample/hematocrit>='50.4' or
bloodsample/hematocrit<='40.6')) or (gender='female' and
(bloodsample/hematocrit>='44.4' or
bloodsample/hematocrit<='36.0'))">
      <td width="10%" align="center"><font color="red"><xsl:value-of
select="bloodsample/hematocrit"/></font></td>
    </xsl:if>
    <xsl:if test="(gender='male' and
(bloodsample/hematocrit<='50.3' and
bloodsample/hematocrit>='40.7')) or (gender='female' and
(bloodsample/hematocrit<='44.3' and
bloodsample/hematocrit>='36.1'))">
      <td width="10%" align="center"><xsl:value-of
select="bloodsample/hematocrit"/></td>
    </xsl:if>
  </tr>
  <!--Na-->
  <tr>
    <th>Na</th>
    <xsl:if test="bloodsample/Na>='146' or
bloodsample/Na<='135'">
      <td width="10%" align="center"><font color="red"><xsl:value-of
select="bloodsample/Na"/></font></td>
    </xsl:if>
    <xsl:if test="bloodsample/Na>='136' and
bloodsample/Na<='145'">
      <td width="10%" align="center"><xsl:value-of
select="bloodsample/Na"/></td>
    </xsl:if>
  </tr>
  <!--K-->
  <tr>
    <th>K</th>
    <xsl:if test="bloodsample/K>='5.6' or
bloodsample/K<='3.4'">
      <td width="10%" align="center"><font color="red"><xsl:value-of
select="bloodsample/K"/></font></td>
    </xsl:if>
    <xsl:if test="bloodsample/K>='5.5' and
bloodsample/K>='3.5'">
      <td width="10%" align="center"><xsl:value-of
select="bloodsample/K"/></td>
    </xsl:if>
  </tr>
  <!--Urine-->
  <tr>
    <th>Urea</th>
    <xsl:if test="bloodsample/Urea>='19.0' or
bloodsample/Urea<='6.0'">
      <td width="10%" align="center"><font color="red"><xsl:value-of
select="bloodsample/Urea"/></font></td>
    </xsl:if>
  </tr>

```



```

    <xsl:if test="bloodsample/Urea&gt;='7.0' and
bloodsample/Urea&lt;='18.0'">
      <td width="10%" align="center"><xsl:value-of
select="bloodsample/Urea"/></td>
    </xsl:if>
  </tr>
  <!--Cr (eatinine) Cl (earance) -->
  <tr>
    <th>CrCl</th>
    <xsl:if test="(gender='male' and
(bloodsample/Creatinine&gt;='1.4' or
bloodsample/Creatinine&lt;='0.6')) or (gender='female' and
(bloodsample/Creatinine&gt;='1.2' or
bloodsample/Creatinine&lt;='0.5'))">
      <td width="10%" align="center"><font color="red"><xsl:value-of
select="bloodsample/Creatinine"/></font></td>
    </xsl:if>
    <xsl:if test="(gender='male' and
(bloodsample/Creatinine&lt;='1.3' and
bloodsample/Creatinine&gt;='0.7')) or (gender='female' and
(bloodsample/Creatinine&lt;='1.1' and
bloodsample/Creatinine&gt;='0.6'))">
      <td width="10%" align="center"><xsl:value-of
select="bloodsample/Creatinine"/></td>
    </xsl:if>
  </tr>
  <!--Clc-->
  <tr>
    <th>Glc</th>
    <xsl:if test="bloodsample/Glucose&gt;='5.9' or
bloodsample/Glucose&lt;='3.8'">
      <td width="10%" align="center"><font color="red"><xsl:value-of
select="bloodsample/Glucose"/></font></td>
    </xsl:if>
    <xsl:if test="bloodsample/Glucose&gt;='3.9' and
bloodsample/Glucose&lt;='5.8'">
      <td width="10%" align="center"><xsl:value-of
select="bloodsample/Glucose"/></td>
    </xsl:if>
  </tr>
</table>
</xsl:if>
</xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

5.9.2: Παρατηρήσεις

Όλος ο πίνακας, είναι φωλιασμένος σε μια `<xsl:if>`, που σκοπό έχει να διαλέξει τα αιματολογικά στοιχεία του επιλεγμένου ασθενή. Οι ασθενείς διακρίνονται μέσω του αριθμού ασφάλισης τους, μοναδικό στον καθένα, το οποίο περνάει στο πεδίο `test` της εντολής. Παρατηρούμε ότι χρησιμοποιούνται οι ίδιοι ελεγκτικοί μηχανισμοί, για το αν μια τιμή σε κάποιο στοιχείο είναι εκτός των φυσιολογικών ορίων, με αυτούς που

διέπουν την εμφάνιση του αιματοκρίτη. Δηλαδή, οι τιμές εκτός των φυσιολογικών ορίων χρωματίζονται με κόκκινο.

5.10:Java Servlet: MedServlet:

5.10.1:Κώδικας

Η κεντρική κλάση που πραγματοποιεί την προσαρμογή του περιεχομένου της ιστοσελίδας, ανάλογα με τον τύπο του user agent που ανιχνεύει εντός της αίτησης POST του πρωτοκόλλου HTTP και τις αντίστοιχες τερματικές δυνατότητες που εξορύσει από το αρχείο WURFL, δίνεται παρακάτω:

```
package medical;

import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.xml.transform.*;

/**
 *
 * @author Rob
 */
//Main servlet class. This instance is triggered when a JSP page
sends a POST header to the Server (Glassfish)
public class MedServlet extends HttpServlet {
    //Global variables

    File patientsxml = new File("C:\\MedicalApp\\web\\WEB-
INF\\patients.xml");
    File patientsMobilexsl = new File("C:\\MedicalApp\\web\\WEB-
INF\\patientsMobile.xsl");
    File patientsMobileTextxsl = new File("C:\\MedicalApp\\web\\WEB-
INF\\patientsMobileText.xsl");
    File patientsMobileNoVideoxsl = new
File("C:\\MedicalApp\\web\\WEB-INF\\patientsMobileNoVideo.xsl");
    File patientsPDAXsl = new File("C:\\MedicalApp\\web\\WEB-
INF\\patientsPDA.xsl");
    File patientsComputerxsl = new File("C:\\MedicalApp\\web\\WEB-
INF\\patientsComputer.xsl");
    WurflParser wp;
    //Method that output the dynamical HTML code

    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html; charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {/*
            //What works and what not in an user agent (single-line
quotes for non working)
```

```

        out.println("<h1>Accept: " + request.getHeader("accept")
+ "</h1>");
        out.println("<h1>User-agent: " + request.getHeader("user-
agent") + "</h1>");
        out.println("<h1>Accept-charset: " +
request.getHeader("accept-charset") + "</h1>");
        out.println("<h1>Accept-language: " +
request.getHeader("accept-language") + "</h1>");
        out.println("<h1>Accept-encoding: " +
request.getHeader("accept-encoding") + "</h1>");
        out.println("<h1>Cookie: " + request.getHeader("cookie")
+ "</h1>");
        //out.println("<h1>Cache-control: " +
request.getHeader("cache-control") + "</h1>");
        out.println("<h1>Connection: " +
request.getHeader("connection") + "</h1>");
        out.println("<h1>Host: " + request.getHeader("host") +
"</h1>");
        //out.println("<h1>Oracle-ECID: " +
request.getHeader("oracle-ecid") + "</h1>");
        //out.println("<h1>X-wap-profile: " +
request.getHeader("x-wap-profile") + "</h1>");*/

        //Initial of the dynamical web page code
        out.println("<html>");
        out.println("<head>");
        out.println("<title>HealthyGuy General Hospital stat
services</title>");
        //Parser's object
        wp = new WurflParser(request.getHeader("user-agent"));
        //Information upon how many agents have been found
        //out.println("Total Matches for \"" + wp.keyword + "\":
" + wp.deviceList.size() + "\n");
        //Check if the device is mobile, PDA or computer.
wp.deviceList
        //must return 1 or more results to validate that WURFL
found a matching device
        if (wp.deviceList.size() == 0) {
            doComputer(request, response, out);
        } else if ((wp.WIFI && (wp.XHTMLsl>=3) &&
(request.getHeader("user-agent").indexOf("Windows CE") >= 0)
            || (request.getHeader("user-
agent").indexOf("Nintendo") >= 0)
            || (request.getHeader("user-
agent").indexOf("iPhone") >= 0)
            || (request.getHeader("user-
agent").indexOf("iPAQ") >= 0)
            || (request.getHeader("user-
agent").indexOf("PlayStation") >= 0)
            || (request.getHeader("user-
agent").indexOf("Android") >= 0))) {
            doPDA(request, response, out);
        } else if ((wp.XHTMLsl>=3) && wp.WIFI &&
(wp.resolution_height > 0 && wp.resolution_width > 164)) {
            doMobile(request, response, out);
        } else if (wp.WIFI && wp.XHTMLsl==2){
            doMobileNoVideo(request, response, out);
        } else if ((wp.XHTMLsl==2) || (wp.XHTMLsl==1) ||
(wp.XHTMLsl==0)){
            doMobileText(request, response, out);
        } else

```

```

        out.println("</title>\n<body>\n<h1>Can't show
data</h1>");
        /**/Device's capabilities display
        for (int i = 0; i < wp.deviceList.size(); i++) {
            out.print("<br>" + wp.deviceList.elementAt(i));
        }*/
        //end of the dynamical web page code
        out.println("</body>");
        out.println("</html>");
    } catch (Exception e) {
        e.printStackTrace();
    } finally { //end of String stream
        out.close();
    }
}
// GET and POST methods, which call processRequest() both of them

/**
 * Handles the HTTP <code>GET</code> method.
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HTTPServletRequest request,
HTTPServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HTTPServletRequest request,
HTTPServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>

//What if the device is a computer (Tower, laptop, netbook)
public void doComputer(HTTPServletRequest request,
HTTPServletResponse response, PrintWriter out)
    throws TransformerConfigurationException,
TransformerException {
    out.println("<style type=\"text/css\">\nbody,td,th {color:
#000000;");

```

```

        + "\nbody {background-color:
#FFFFFF;}\n</style>\n</head>\n<body>\n");
        TransformerFactory tFactory =
TransformerFactory.newInstance();
        if (request.getHeader("user-agent").indexOf("Firefox") >= 0)
        {
            out.println("<h1>So, Mozilla user, huh?</h1>");
        } else if (request.getHeader("user-agent").indexOf("MSIE") >=
0) {
            out.println("<h1>Still in Internet explorer?</h1>");
        } else if (request.getHeader("user-agent").indexOf("Opera")
>= 0) {
            out.println("<h1>Opera... best browser to pass ajax
test!</h1>");
        } else if (request.getHeader("user-agent").indexOf("Safari")
>= 0) {
            out.println("<h1>Macintosh? Safari? wow!</h1>");
        } else if (request.getHeader("user-agent").indexOf("Chrome")
>= 0) {
            out.println("<h1>Google Chrome! Lightweight!
Yeah!</h1>");
        } else if (request.getHeader("user-
agent").indexOf("Konqueror") >= 0) {
            out.println("<h1>Konqueror of the web browsers!</h1>");
        } else {
            out.println("<h1>I don't have a clue what browser you
use!</h1>");
        }
        Transformer transformer =
            tFactory.newTransformer(new
javax.xml.transform.stream.StreamSource(patientsComputerxsl));
        transformer.transform(new
javax.xml.transform.stream.StreamSource(patientsxml),
            new javax.xml.transform.stream.StreamResult(out));
    }
    //What if the device is a Personal Digital Assistant
    public void doPDA(HTTPRequest request, HttpServletResponse
response, PrintWriter out) throws TransformerConfigurationException,
TransformerException {
        out.println("<style type=\"text/css\">\nbody,td,th {color:
#FFFFFF;\"
        + "\nbody {background-color:
#999999;}\n</style>\n</head>\n<body>\n");
        TransformerFactory tFactory =
TransformerFactory.newInstance();
        Transformer transformer =
            tFactory.newTransformer(new
javax.xml.transform.stream.StreamSource(patientsPDAXsl));
        transformer.transform(new
javax.xml.transform.stream.StreamSource(patientsxml),
            new javax.xml.transform.stream.StreamResult(out));
    }
    //What if the device is a mobile phone
    public void doMobile(HTTPRequest request,
HttpServletResponse response, PrintWriter out) throws
TransformerConfigurationException, TransformerException {
        out.println("<style type=\"text/css\">\nbody,td,th {color:
#FFFFFF;\"

```

```

        + "\nbody {background-color:
#000000;}\na{color:yellow;}\ntable{max-width: "
        + wp.resolution_width + ";max-height: " +
wp.resolution_height + "};</style>\n</head>\n<body>\n");
        TransformerFactory tFactory =
TransformerFactory.newInstance();
        Transformer transformer =
            tFactory.newTransformer(new
javax.xml.transform.stream.StreamSource(patientsMobilexsl));
        transformer.transform(new
javax.xml.transform.stream.StreamSource(patientsxml),
            new javax.xml.transform.stream.StreamResult(out));
    }
    //What if the device is a mobile phone
    public void doMobileNoVideo(HttpServletRequest request,
HTTPServletResponse response, PrintWriter out) throws
TransformerConfigurationException, TransformerException {
        out.println("<style type=\"text/css\">\nbody,td,th {color:
#FFFFFF; "
            + "\nbody {background-color:
#000000;}\na{color:yellow;}\ntable{max-width: "
            + wp.resolution_width + ";max-height: " +
wp.resolution_height + "};</style>\n</head>\n<body>\n");
        TransformerFactory tFactory =
TransformerFactory.newInstance();
        Transformer transformer =
            tFactory.newTransformer(new
javax.xml.transform.stream.StreamSource(patientsMobileNoVideoxsl));
        transformer.transform(new
javax.xml.transform.stream.StreamSource(patientsxml),
            new javax.xml.transform.stream.StreamResult(out));
    }
    //What if the device is a mobile phone with limited XHTML-MP
capabilities
    public void doMobileText(HttpServletRequest request,
HTTPServletResponse response, PrintWriter out) throws
TransformerConfigurationException, TransformerException {
        out.println("</head>\n<body bgcolor=\"black\"
text=\"white\">\n");
        TransformerFactory tFactory =
TransformerFactory.newInstance();
        Transformer transformer =
            tFactory.newTransformer(new
javax.xml.transform.stream.StreamSource(patientsMobileTextxsl));
        transformer.transform(new
javax.xml.transform.stream.StreamSource(patientsxml),
            new javax.xml.transform.stream.StreamResult(out));
    }
}

```

5.10.2: Τα βασικά της λειτουργίας του Servlet

Ο servlet εκτελεί τη μέθοδο `processRequest` με το που δημιουργείται ένα instance αυτού, ταυτόχρονα δηλαδή με την άφιξη της εντολής POST που εκκινά από την ιστοσελίδα `index.jsp`. Η πραγματική μέθοδος που δέχεται την αίτηση είναι η `doPOST`, που όμως, η μόνη της εντολή είναι το κάλεσμα της μεθόδου `processRequest`.

Λόγω του ότι δεν πρέπει να διακοπεί η λειτουργία, τόσο του Server, όσο και του servlet, αρχικά όλος ο κώδικας δοκιμάζεται αν μπορεί να εκτελεστεί. Διαφορετικά, αναφέρεται το πρόβλημα χωρίς, όμως, να διακοπεί η υπηρεσία HTTP από πλευράς χρήστη, ο οποίος λαμβάνει την εισαγωγική ιστοσελίδα και μπορεί ανα πάσα στιγμή να στείλει αίτηση POST.

Η μέθοδος `processRequest` αναλαμβάνει να δώσει τους αλφαριθμητικούς χαρακτήρες για έναρξη του κώδικα HTML, να αναγνωρίσει αν μια συσκευή που αιτείται περιεχόμενο είναι Η/Υ, κινητό ή PDA, ώστε να χρησιμοποιηθεί η κατάλληλη μέθοδος, και να ολοκληρώσει την δυναμική παραγωγή ιστοσελίδας με τις HTML ετικέτες κλεισίματος.

Επίσης, στην `processRequest` μέθοδο, δηλώνονται τρία αντικείμενα, τα:

- `out`, αντικείμενο τύπου `PrintWriter`.
- `request`, αντικείμενο τύπου `HttpServletRequest`.
- `response`, αντικείμενο τύπου `HttpServletResponse`.

Τα αντικείμενα αυτά, διαχειρίζονται βασικές λειτουργίες του servlet, και μελετούνται παρακάτω διεξοδικά.

Μέθοδοι επικάλυσης υλοποιούνται, για την πολλαπλή κληρονομικότητα που χρειάζεται ο servlet για να δουλέψει, με τα ονόματα `doGET`, `doPOST`, `getServletInfo`. Οι δύο πρώτες διαχειρίζονται τις αιτήσεις του HTTP πρωτοκόλλου, ενώ η τελευταία γυρνάει ένα κείμενο με μια περιγραφή του servlet, για κάθε χρήση. Δεν θα μας απασχολήσουν οι συγκεκριμένες μέθοδοι, διότι δεν μας χρειάζονται στην υλοποίηση της εφαρμογής, παρά μόνον αναφορικά.

Τέλος, οι μέθοδοι που αναλαμβάνουν την παραγωγή και παρουσίαση του περιεχομένου της ιστοσελίδας HTML από το XML αρχείο των ασθενών, με χρήση του XSLT που αντιστοιχεί στην περίσταση των δυνατοτήτων της τερματικής συσκευής πελάτη, βρίσκονται στο τέλος. Είναι οι:

- `doComputer`, μέθοδος για Η/Υ.
- `doPDA`, μέθοδος για PDAs.
- `doMobile`, μέθοδος για κινητά τηλέφωνα.
- `doMobileNoVideo`, μέθοδος για κινητά τηλέφωνα με περιορισμένες δυνατότητες πολυμέσων.
- `doMobileText`, μέθοδος για κινητά τηλέφωνα χωρίς ή πάρα πολύ περιορισμένες πολυμεσικές δυνατότητες.

5.10.3: Το αντικείμενο `out`

Η μεταβλητή `out`, είναι υπεύθυνη να αποστείλει μια ροή χαρακτήρων, στον πελάτη. Από την αρχή, καθορίζεται ο τύπος του

περιεχομένου, που στην περίπτωση μας είναι text/html, με την μέθοδο `setContentTypes`, σύμφωνα με τις πληροφορίες επικεφαλίδας του πρωτοκόλλου HTTP της W3C. Ο πελάτης, με αυτή την συνθήκη, ότι δεδομένα δεχτεί, την ακολουθία των χαρακτήρων δηλαδή, θα τα εκλάβει ως κώδικα HTML.

Με την μέθοδο `print` και `println`, δίνουμε μέσα στις παρενθέσεις, το όρισμα των χαρακτήρων που επιθυμούμε να στείλουμε στον πελάτη. Για αρχή, ξεκινάμε με την έναρξη του κώδικα της ιστοσελίδας:

```
out.println("<html>");
```

Ο πελάτης θα δεχτεί το κείμενο `<html>`, και λόγω του ότι δέχτηκε πως θα το μεταφράσει ο πλοηγός ως κώδικα HTML, ξεκινάει η μεταγλώττιση. Περαιτέρω ετικέτες και δεδομένα δίνονται στον πελάτη, μέχρι να φτάσουμε στο σημείο της επιλογής περιεχομένου, ανάλογα με τις δυνατότητες της τερματικής συσκευής. Εκεί, αφού μεταφραστεί σε HTML το XML μέσω του XSLT, όλο το παραγόμενο αποτέλεσμα καταλήγει στο αντικείμενο `out`, οπότε και δίνεται στον πελάτη, ως κώδικας HTML.

```
Transformer transformer =  
    tFactory.newTransformer(new  
    javax.xml.transform.stream.StreamSource(patientsComputerxsl));  
  
    transformer.transform(new  
    javax.xml.transform.stream.StreamSource(patientsxml),  
        new javax.xml.transform.stream.StreamResult(out));
```

Η δεύτερη εντολή, είναι στην ουσία είσοδος των δεδομένων του XML αρχείου και έξοδος κώδικα HTML σε κάποιο αντικείμενο, βάση της φόρμας XSL που του αναφέραμε στην πρώτη εντολή. Το αντικείμενο εξόδου μας, είναι στην περίπτωσή μας, η αποστολή μιας ροής δεδομένων προς τον πελάτη.

Αφού μεταδοθεί και η ετικέτα `</html>` στον πελάτη, κλείνουμε την ροή των χαρακτήρων από το αντικείμενο `out`, με τη χρήση της μεθόδου `close`.

5.10.4: Το αντικείμενο request

Για να αναγνωρίσουμε τι ζητάει ο πελάτης από τον εξυπηρετητή, τοποθετούμε όλη την αίτησή του στο αντικείμενο `request`. Αυτό, κρατάει τους αλφαριθμητικούς χαρακτήρες, βάση της προτυποποίησης του W3C, και επιτρέπει σε συγκεκριμένες μεθόδους να εξάγουν πληροφορίες σχετικά με το τι ζητάει ο πελάτης.

Στα σχόλια εντός του κώδικα, υπάρχουν, επιπλέον, κάποιες απενεργοποιημένες γραμμές κώδικα που αναλύουν και καταχωρούν σε

αλφαριθμητικές μεταβλητές, τις πληροφορίες που στέλνει ο πελάτης στον εξυπηρετητή. Από αυτές, εμάς μας ενδιαφέρει μόνο ο user agent του πελάτη, που τον διαβάζουμε με την εξής εντολή:

```
request.getHeader("user-agent")
```

Η μέθοδος `getHeader`, διαβάζει το πεδίο της αίτησης του πελάτη που του ζητάμε στα εισαγωγικά (το user agent δηλαδή), και αυτή μας γυρίζει πίσω την αλφαριθμητική του τιμή. Την τιμή αυτή τη χρησιμοποιούμε σε συνδυασμό με τον parser του WURFL αρχείου, δημιουργώντας ένα αντικείμενο τύπου `WURFLParser`, που είναι η κλάση του, για να τροφοδοτήσουμε μέσω αυτού με δεδομένα τον parser, ώστε να μας γυρίσει σημαντικές πληροφορίες.

Η σύνταξη του είναι η εξής:

```
WURFLParser wp = new WurflParser(request.getHeader("user-agent"));
```

Παράλληλα, την τιμή του user agent τη χρησιμοποιούμε, επίσης, στους βρόγχους επιλογής του περιεχομένου ιστοσελίδας, μελετώντας αν η τιμή του user agent περιέχει κάποιες λέξεις κλειδιά που είναι ενδεικτικές των πιο διάσημων συσκευών PDA και παιχνιδομηχανών με δυνατότητα πρόσβασης στο διαδίκτυο. Συγκεκριμένα, στον servlet έχουμε:

```
if (wp.deviceList.size() == 0) {
    doComputer(request, response, out);
}
else
if ((wp.WIFI && (wp.XHTMLsl>=3) && (request.getHeader("user-agent").indexOf("Windows CE")>=0) || (request.getHeader("user-agent").indexOf("Nintendo")>=0) || (request.getHeader("user-agent").indexOf("iPhone")>=0) || (request.getHeader("user-agent").indexOf("iPAQ")>=0) || (request.getHeader("user-agent").indexOf("PlayStation")>=0) || (request.getHeader("user-agent").indexOf("Android") >= 0)))
{
    doPDA(request, response, out);
}
else
if ((wp.XHTMLsl>=3) && wp.WIFI && (wp.resolution_height > 0 && wp.resolution_width > 164))
{
    doMobile(request, response, out);
}
else
if (wp.WIFI && wp.XHTMLsl==2) {
    doMobileNoVideo(request, response, out);
}
else
if ((wp.XHTMLsl==2) || (wp.XHTMLsl==1) || (wp.XHTMLsl==0)) {
    doMobileText(request, response, out);
}
else
```

```
out.println("</title>\n<body>\n<h1>Can't show  
data</h1>");
```

Αν δεν βρεθεί καταχώριση στο αρχείο του WURFL που να αντιστοιχεί στην τιμή του user agent του πελάτη, αποφασίζεται πως η συσκευή είναι τύπου Η/Υ, οπότε και εκτελείται η μέθοδος doComputer. Αν, αντίθετα, εντός του WURFL για τις δυνατότητες του συγκεκριμένου user agent, βρεθεί μια λέξη κλειδί από τις πιο γνωστές μάρκες PDA που κυκλοφορούν στην αγορά, ή αν περιέχεται η ένδειξη "Windows CE", που απαιτούν δυνατότητες PDA, τουλάχιστον, για να εκτελεστούν, τότε αποφασίζεται πως η συσκευή είναι τύπου PDA και εκτελείται η μέθοδος doPDA. Στις υπόλοιπες περιπτώσεις, λαμβάνεται υπ' όψιν το επίπεδο απόδοσης ιστοσελίδας που έχει η συσκευή, καθώς και η μέγιστη ανάλυση οθόνης, σε εικονοστοιχεία, για την επιλογή της δυναμικής προσαρμογής περιεχομένου που υλοποιηθεί για τον συγκεκριμένο πελάτη.

Την τιμή του user agent τη χρησιμοποιούμε, τέλος, στους βρόγχους αναγνώρισης του τύπου πλοηγού πελάτη, στη μέθοδο doComputer.

5.10.5: Το αντικείμενο response

Το αντικείμενο response, περιέχει την απάντηση του εξυπηρετητή προς τον πελάτη. Καθορίζουμε ότι θα είναι μια σειρά αλφαριθμητικών χαρακτήρων, κάτι που το αναφέρουμε με την εντολή:

```
PrintWriter out = response.getWriter();
```

Αναγκάζουμε το αντικείμενο out, δηλαδή, να περιέχει αυστηρά, μόνο χαρακτήρες. Επίσης, όπως προαναφέραμε, με την βοήθεια της μεθόδου setContentTypes, θέτουμε και την οδηγία ο πελάτης να μεταφράσει το ληφθέν κείμενο σε γλώσσα HTML.

5.10.6: Μέθοδοι μεταγλώττισης XML, ανάλογα με τις δυνατότητες της συσκευής

Οι πέντε προαναφερόμενες μέθοδοι, για την μετάφραση μιας ιστοσελίδας σε HTML από XML, με χρήση του XSLT, παρατίθενται προς το τέλος του κώδικα. Όλες οι μέθοδοι περιέχουν παρόμοιο κώδικα, διαθέτουν πληροφορίες φόρμας της ιστοσελίδας, αλλά έχουν τις εξής διαφορές:

- Η doComputer, περιέχει βρόγχους για την αναγνώριση του τύπου του φυλλομετρητή.
- Η doPDA, διαφέρει στη φόρμα προβολής της ιστοσελίδας, σχετικά με το χρώμα φόντου και κειμένου.
- Οι μέθοδοι που αναλαμβάνουν την προβολή ιστοσελίδας σε κινητά, εκτός του ότι έχουν διαφορετικό χρωματισμό για το φόντο

και το κείμενο, διαθέτουν επίσης «κατώφλια» πλάτους και ύψους της ιστοσελίδας. Τις τιμές για τις διαστάσεις αυτές, τις λαμβάνουμε από τον WURFL parser, μέσω των κοινοποιημένων μεταβλητών `resolution_width` και `resolution_height`.

```
out.println("<style type=\"text/css\">\nbody,td,th {color:
#FFFFFF;"+ " }\nbody {background-color:
#000000;}\na{color:yellow;}\ntable{max-width: " +
wp.resolution_width + ";max-height: " + wp.resolution_height +
";}</style>\n</head>\n<body>\n");
```

Επειδή οι μεταβλητές αυτές θα επιστρέψουν αριθμητικές τιμές, εντάσσονται κατ' ευθείαν στον HTML κώδικα και παράγουν το επιθυμητό αποτέλεσμα, δηλαδή, μορφοποιούν την ιστοσελίδα ανάλογα με τις δυνατότητες της οθόνης της τερματικής συσκευής.

5.11:WURFL Parser:

5.11.1:Κώδικας

Μια επιπλέον κλάση, αυτή του parser του αρχείου WURFL πρέπει να συμπεριληφθεί. Παρακάτω δίνεται ο κώδικας:

```
package medical;

/**
 *
 * @author Rob
 */
import java.io.File;
import java.util.Hashtable;
import java.util.Vector;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;
import org.xml.sax.Attributes;
import org.xml.sax.helpers.DefaultHandler;

public class WurflParser extends DefaultHandler
{
    //public variables to be used in MedServlet
    public Vector<Device> deviceList;
    public Device currentDevice;
    public Group currentGroup;
    public String keyword, mobilebrowser;
    File wurflxml = new File("C:\\MedicalApp\\web\\WEB-
INF\\wurfl.xml");
    Boolean
    WIFI=false, jpg=false, gif_animated=false, bmp=false, tiff=false;
    int
    XHTMLsl, resolution_width=64, resolution_height=0, max_image_width, max_i
mage_height;
    /* //A main method, to test the performance of the parser
    public static void main(String[] args)
    {
        WurflParser wp = new WurflParser("Nokia6600");
```

```

        System.out.println("Total Matches for \"" + wp.keyword + "\":
" + wp.deviceList.size() + "\n");
        for(int i = 0; i < wp.deviceList.size(); i++)
            System.out.println(wp.deviceList.elementAt(i));
    }*/
    //Parser's method
    public WurflParser(String keyword)
    {
        this.keyword = keyword.toLowerCase().trim();
        this.deviceList = new Vector<Device>();

        try
        {
            SAXParserFactory spf = SAXParserFactory.newInstance();
            SAXParser sp = spf.newSAXParser();
            sp.parse(wurflxml, this);
        }
        catch (Exception e)
        {
            System.out.println("Exception while parsing the file. " +
e);
        }
    }
    //Registration of the groups of capabilities of the WURFL's XML
file
    public void startElement(String uri, String localName, String
qName,
        Attributes attributes)
    {
        if (qName.equalsIgnoreCase("device") &&
attributes.getLength() > 0)
        {
            if (!isValidDevice(attributes.getValue("user_agent")))
                return;
            this.currentDevice = new Device();
            this.currentDevice.id = attributes.getValue("id");
            this.currentDevice.fallBack =
attributes.getValue("fall_back");
            this.currentDevice.actualDeviceRoot =
Boolean.parseBoolean(attributes.getValue("actual_device_root"));
            this.currentDevice.userAgent =
attributes.getValue("user_agent");
        }

        else if (qName.equalsIgnoreCase("group") &&
(attributes.getLength() > 0)
            && (this.currentDevice != null))
        {
            this.currentGroup = new Group();
            this.currentGroup.id = attributes.getValue("id");
        }

        else if (qName.equalsIgnoreCase("capability") &&
(attributes.getLength() > 0)
            && (this.currentGroup != null))
        {
            this.currentGroup.addAttribute(attributes.getValue("name"),
attributes.getValue("value"));
            //Those logical tests, registry the values to public
variables

```

```

        //so they can be used to MedServlet

        if (attributes.getValue("name").compareToIgnoreCase("wifi")==0 &&
            attributes.getValue("value").compareToIgnoreCase("true")==0)
            WIFI=true;

        if (attributes.getValue("name").compareToIgnoreCase("resolution_width"
        )==0)

        resolution_width=Integer.parseInt(attributes.getValue("value"));

        if (attributes.getValue("name").compareToIgnoreCase("resolution_height
        ")==0)

        resolution_height=Integer.parseInt(attributes.getValue("value"));

        if (attributes.getValue("name").compareToIgnoreCase("jpg")==0 &&
            attributes.getValue("value").compareToIgnoreCase("true")==0)
            jpg=true;

        if (attributes.getValue("name").compareToIgnoreCase("bmp")==0 &&
            attributes.getValue("value").compareToIgnoreCase("true")==0)
            bmp=true;

        if (attributes.getValue("name").compareToIgnoreCase("gif_animated")==0
        && attributes.getValue("value").compareToIgnoreCase("true")==0)
            gif_animated=true;

        if (attributes.getValue("name").compareToIgnoreCase("tiff")==0 &&
            attributes.getValue("value").compareToIgnoreCase("true")==0)
            tiff=true;

        if (attributes.getValue("name").compareToIgnoreCase("mobile_browser")=
        =0 && attributes.getValue("value").compareToIgnoreCase("")>0)
            mobilebrowser=attributes.getValue("value");

        if (attributes.getValue("name").compareToIgnoreCase("xhtml_support_lev
        el")==0)

        XHTMLsl=Integer.parseInt(attributes.getValue("value"));

        }
    }
    //Finishing the registration
    public void endElement(String uri, String localName, String
    qName)
    {
        if (qName.equalsIgnoreCase("device") && (this.currentDevice
        != null))
        {
            this.deviceList.add(this.currentDevice);
            this.currentDevice = null;
        }

        else if (qName.equalsIgnoreCase("group") &&
        (this.currentGroup != null))
        {
            this.currentDevice.addGroup(this.currentGroup);
            this.currentGroup = null;
        }
    }
}

```

```

//Check if the user agent exists. If not, the MedServlet will
proceed to
//the doComputer method
private boolean isValidDevice(String userAgent)
{
    boolean status = (userAgent != null) &&
(userAgent.toLowerCase().indexOf(this.keyword) > -1);
    if (status)
        System.out.println("Comparing [" + status + "] " +
userAgent);
    return (status);
}
//This class, emulates the parent element of the XML structure
public static class Device
{
    public String id;
    public String userAgent;
    public String fallBack;
    public boolean actualDeviceRoot;
    public Vector<Group> group;

    public void addGroup(Group group)
    {
        this.group = (this.group == null)
            ? new Vector<Group>()
            : this.group;
        this.group.add(group);
    }

    public String toString()
    {
        StringBuilder stringBuilder = new StringBuilder();

        stringBuilder.append("Id: " + this.id + "\n");
        stringBuilder.append("UserAgent: " + this.userAgent +
"\n");
        stringBuilder.append("FallBack: " + this.fallBack +
"\n");
        stringBuilder.append("Actual Device Root: " +
this.actualDeviceRoot + "\n");

        if (this.group != null)
        {
            for(int i = 0; i < this.group.size(); i++)
                stringBuilder.append(this.group.elementAt(i));
        }

        return (stringBuilder.toString());
    }
}
//This class, emulates the child element of the XML structure
public static class Group
{
    public String id;
    public Hashtable<String, String> attributes = null;

    public void addAttribute(String key, String value)
    {
        this.attributes = (this.attributes == null)
            ? new Hashtable<String, String>()
            : this.attributes;
    }
}

```

```

        this.attributes.put(key, value);
    }

    public String toString()
    {
        StringBuilder stringBuilder = new StringBuilder();

        if (this.attributes != null)
        {
            stringBuilder.append(" [" + this.id + "] \n");
            for (String key : this.attributes.keySet())
            {
                stringBuilder.append("   " + key + ": "
                    + this.attributes.get(key) + "\n");
            }
        }

        stringBuilder.append("\n");
        return (stringBuilder.toString());
    }
}

```

5.11.2:Βασική λειτουργία

Ο parser που χρησιμοποιήσαμε στην εφαρμογή μας είναι μια προσαρμογή του WURFL parser που περιέχεται στην ιστοσελίδα [HTTP://javabeanz.wordpress.com/2007/06/27/wurfl-parser/](http://javabeanz.wordpress.com/2007/06/27/wurfl-parser/), με επιπλέον κώδικα για εξαγωγή και κοινοποίηση τιμών, σε μεταβλητές που χρησιμοποιούμε στον servlet.

Ο συγκεκριμένος parser χρησιμοποιεί το SAX API [2] για να αναγνώσει τις τιμές εντός του αρχείου XML, και όταν φτάσει στο σημείο να διαβάσει τα capabilities μιας συσκευής, ελέγχει αν περιέχεται τιμή για τα ονόματα που αναζητούμε. Τις τιμές που ανιχνεύει τις περνάει στις κοινοποιημένες μεταβλητές ολόκληρης της κλάσης, για χρήση εντός του servlet.

5.11.3:Κοινοποιημένες μεταβλητές

Για τη χρήση των πληροφοριών των δυνατοτήτων μιας συσκευής στον servlet, πρέπει να συσχετίζουμε το instance του parser με το όνομα της μεταβλητής. Για παράδειγμα, αν θέλουμε να ξεχωρίσουμε αν μια συσκευή διαθέτει κάρτα WiFi ή όχι, τότε στον κεντρικό κώδικα του servlet ζητάμε τη μεταβλητή ως **wp.WIFI**. Παρατηρούμε πολλές τέτοιες αναφορές στον κώδικα, μιας και αυτός είναι ο μόνος συνδετήριος κρίκος του servlet με τις δυνατότητες της συσκευής.

Μερικές ενδεικτικές μεταβλητές έχουν δηλωθεί στον κώδικα του parser, ενώ μπορούν να εισαχθούν και άλλες, για πιο λεπτομερή περιγραφή, για τις δυνατότητες μιας συσκευής. Κατά βούληση, ζητάμε στη μέθοδο startElement να τροφοδοτεί με τιμές τις αντίστοιχες

μεταβλητές του servlet, ανάλογα με τον κώδικα που έχουμε συμπεριλάβει στην κεντρική ρουτίνα.

Κεφάλαιο 6^ο - Αποτελέσματα

6.1: Παραγόμενες ιστοσελίδες:

6.1.1: Εξομοίωση

Για τις ανάγκες ελέγχου της ορθής λειτουργίας του servlet, χρησιμοποιήθηκε ο πλοηγός Firefox, για την ανάγνωση και παρουσίαση του παραγόμενου κώδικα. Λόγω της δυσχέρειας να βρεθεί εξοπλισμός, ίδιος ή αντάξιος του εικονικού σεναρίου, χρησιμοποιήθηκε ένα πρόσθετο (plug in) του άνωθεν πλοηγού, που λέγεται user agent switcher, το οποίο δίνει τη δυνατότητα στον firefox να μεταλλάσσει κατά βούληση την ταυτότητα πλοηγού που ανακοινώνει προς τα έξω, αποστέλλοντας στον εκάστοτε server HTTP αιτήσεις με τιμή στο πεδίο user agent αυτήν που επιθυμεί κάθε φορά ο χρήστης. Πιο συγκεκριμένα, το συγκεκριμένο πρόσθετο, αλλάζει τον user agent του πελάτη κατά βούληση, ανάλογα με την τιμή που επιλέγουμε από ένα μενού με όλα τα διαθέσιμα user agents της αγοράς και η οποία καταχωρείται, ακολούθως, σε ένα XML αρχείο. Ο δημιουργός του είναι ο Chris Pederick και το δίνει για δωρεάν μεταφόρτωση από την ιστοσελίδα [HTTP://chrispederick.com/work/user-agent-switcher/](http://chrispederick.com/work/user-agent-switcher/).

6.1.2: Καταχωρίσεις των user agents

Στην λίστα αυτού του πρόσθετου, εκτός από τους user agents των πιο γνωστών πλοηγών, περιλαμβάνονται και μερικές από τις πιο γνωστές συσκευές, κινητής τηλεφωνίας και PDAs. Κάθε μία, συνοδεύεται από ένα επίπεδο σωστής απόδοσης μιας HTML ιστοσελίδας, το οποίο και αναφέρεται μέσα σε αστεράκια. Το XML του πρόσθετου αυτού είναι ο εξής κώδικας:

```
<useragentswitcher>

<folder description="Chrome">
<useragent description="Chrome 0.2" useragent="Mozilla/5.0 (Windows;
U; Windows NT 5.1; en-US) AppleWebKit/525.19 (KHTML, like Gecko)
Chrome/0.2.153.1" appcodename="Mozilla" appname="Netscape"
appversion="5.0 (Windows; el)" platform="Win32" vendor=""
vendorsub="" />
</folder>

<folder description="Firefox">
<useragent description="Firefox 3.x" useragent="Mozilla/5.0 (Windows;
U; Windows NT 5.1; en-US; rv:x.x.x) Gecko/20041107 Firefox/x.x"
appcodename="Mozilla" appname="Netscape" appversion="5.0 (Windows;
el)" platform="Win32" vendor="" vendorsub="" />
</folder>
```

```

<folder description="Internet Explorer">
<useragent description="Internet Explorer 6" useragent="Mozilla/4.0
(compatible; MSIE 6.0; Windows NT 5.1)" appcodename="Mozilla"
appname="Microsoft Internet Explorer" appversion="4.0 (compatible;
MSIE 6.0; Windows NT 5.1)" platform="Win32" vendor="" vendorsub=""/>
<useragent description="Internet Explorer 7" useragent="Mozilla/4.0
(compatible; MSIE 7.0; Windows NT 6.0)" appcodename="Mozilla"
appname="Microsoft Internet Explorer" appversion="4.0 (compatible;
MSIE 7.0; Windows NT 6.0)" platform="Win32" vendor="" vendorsub=""/>
<useragent description="Internet Explorer 8" useragent="Mozilla/4.0
(compatible; MSIE 8.0; Windows NT 6.1)" appcodename="Mozilla"
appname="Microsoft Internet Explorer" appversion="4.0 (compatible;
MSIE 8.0; Windows NT 6.1)" platform="Win32" vendor="" vendorsub=""/>
</folder>

<folder description="Konqueror"><useragent description="Konqueror
2.1.1" useragent="Mozilla/5.0 (compatible; Konqueror/2.1.1; X11)"
appcodename="Mozilla" appname="Netscape" appversion="5.0 (Windows;
el)" platform="Win32" vendor="" vendorsub=""/>
<useragent description="Konqueror 2.2" useragent="Mozilla/5.0
(compatible; Konqueror/2.2.2; Linux 2.4.14-xf; X11; i686)"
appcodename="Mozilla" appname="Netscape" appversion="5.0 (Windows;
el)" platform="Win32" vendor="" vendorsub=""/></folder><folder
description="Opera">
<useragent description="Opera 5 SunOS" useragent="Mozilla/5.0 (SunOS
5.8 sun4u; U) Opera 5.0 [en]" appcodename="Mozilla"
appname="Netscape" appversion="5.0 (Windows; el)" platform="Win32"
vendor="" vendorsub=""/>
<useragent description="Opera 5.12" useragent="Mozilla/4.76 (Windows
98; U) Opera 5.12 [en]" appcodename="Mozilla" appname="Netscape"
appversion="5.0 (Windows; el)" platform="Win32" vendor=""
vendorsub=""/>
</folder>

<folder description="Safari">
<useragent description="Safari 525 Mac" useragent="Mozilla/5.0
(Macintosh; U; Safari/525" appcodename="Mozilla" appname="Netscape"
appversion="5.0 (Windows; el)" platform="Win32" vendor=""
vendorsub=""/>
<useragent description="Safari 525 Win" useragent="Mozilla/5.0
(Windows; U; Safari/525" appcodename="Mozilla" appname="Netscape"
appversion="5.0 (Windows; el)" platform="Win32" vendor=""
vendorsub=""/>
</folder>

<folder description="Hospital devices">
<useragent description="HP iPAQ 5450 *lvl 3*" useragent="Mozilla/4.0
(compatible; MSIE 4.01; Windows CE; PPC; 240x320; HP iPAQ h5450)"
appcodename="Mozilla" appname="Netscape" appversion="5.0 (Windows;
el)" platform="Win32" vendor="" vendorsub=""/>
<useragent description="BlackBerry 8120 *lvl 3*"
useragent="BlackBerry8120/4.3.0 Profile/MIDP-2.0 Configuration/CLDC-
1.1 VendorID/1" appcodename="Mozilla" appname="Netscape"
appversion="5.0 (Windows; el)" platform="Win32" vendor=""
vendorsub=""/>
<useragent description="Nintendo DS *lvl 2*" useragent="Opera/9.50
(Nintendo DSi; Opera/483; U; en-US)" appcodename="Mozilla"
appname="Netscape" appversion="5.0 (Windows; el)" platform="Win32"
vendor="" vendorsub=""/>
<useragent description="PSP Playstation *lvl 3*"
useragent="Mozilla/4.0 (PSP (PlayStation Portable); 2.00)"

```

```
appcodename="Mozilla" appname="Netscape" appversion="5.0 (Windows;
el)" platform="Win32" vendor="" vendorsub=""/>
<useragent description="iPhone 3.0 *lvl 4*" useragent="Mozilla/5.0
(iPhone; U; CPU iPhone OS 3_0 like Mac OS X; en-us)
AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7A341
Safari/528.16" appcodename="Mozilla" appname="Netscape"
appversion="5.0 (iPhone; U; CPU iPhone OS 3_0 like Mac OS X; en-us)
AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7A341
Safari/528.16" platform="iPhone" vendor="Apple Computer, Inc."
vendorsub=""/>
<useragent description="Asus Galaxy II *lvl 3*"
useragent="Mozilla/4.0 (compatible; MSIE 6.0; Windows CE; IEMobile
6.12) ASUS-GalaxyII/1.0" appcodename="Mozilla" appname="Netscape"
appversion="5.0 (Windows; el)" platform="Win32" vendor=""
vendorsub=""/>
<useragent description="Dopod 900 *lvl -1*" useragent="Dopod900
Mozilla/4.0 (compatible; MSIE 4.01; Windows CE; PPC; 480x640;
Dopod900)" appcodename="Mozilla" appname="Netscape" appversion="5.0
(Windows; el)" platform="Win32" vendor="" vendorsub=""/>
<useragent description="HTC &quot;Star Trek&quot; 3100 *lvl 3*"
useragent="HTC-3100/1.2 Mozilla/4.0 (compatible; MSIE 5.5; Windows
CE; Smartphone; 240x320)" appcodename="Mozilla" appname="Netscape"
appversion="5.0 (Windows; el)" platform="Win32" vendor=""
vendorsub=""/>
<useragent description="HTC Dream *lvl 4*" useragent="Mozilla/5.0
(Linux; U; Android HTC_Dream-1.13.0.1; en-us; dream)
AppleWebKit/525.10+ (KHTML, like Gecko) Version/3.0.4 Mobile
Safari/523.12.2" appcodename="Mozilla" appname="Netscape"
appversion="5.0 (Windows; el)" platform="Win32" vendor=""
vendorsub=""/>
<useragent description="LG GT 500 *lvl 3*" useragent="LG/GT500"
appcodename="Mozilla" appname="Netscape" appversion="5.0 (Windows;
el)" platform="Win32" vendor="" vendorsub=""/>
<useragent description="Nokia N81 *lvl 3*" useragent="Mozilla/5.0
(SymbianOS/9.2; U; Series60/3.1 NokiaN81-1/1.00 Profile/MIDP-2.0
Configuration/CLDC-1.1) AppleWebKit/413 (KHTML, like Gecko)
Safari/413" appcodename="Mozilla" appname="Netscape" appversion="5.0
(Windows; el)" platform="Win32" vendor="" vendorsub=""/>
<useragent description="Samsung S8000 *lvl 4*" useragent="SAMSUNG-
S8000/S8000XBIF4 SHP/VPP/R5 Jasmine/1.0 Nextreaming SMM-MMS/1.2.0
profile/MIDP-2.1 configuration/CLDC-1.1" appcodename="Mozilla"
appname="Netscape" appversion="5.0 (Windows; el)" platform="Win32"
vendor="" vendorsub=""/>
<useragent description="Sony Ericsson Pli/R100 *lvl 2*"
useragent="SonyEricssonPli/R100 Mozilla/4.0 (compatible; MSIE 6.0;
Symbian OS; 598) Opera 8.65 [en]" appcodename="Mozilla"
appname="Netscape" appversion="5.0 (Windows; el)" platform="Win32"
vendor="" vendorsub=""/>
<useragent description="LG 8600 *lvl 1 no WIFI*"
useragent="LG8600/1.0 UP.Browser/6.2.3.9 (GUI) MMP/2.0"
appcodename="Mozilla" appname="Netscape" appversion="5.0 (Windows;
el)" platform="Win32" vendor="" vendorsub=""/>
<useragent description="Nokia 6555 *lvl 2*" useragent="Nokia6555"
appcodename="Mozilla" appname="Netscape" appversion="5.0 (Windows;
el)" platform="Win32" vendor="" vendorsub=""/>
</folder>

<separator/>

<useragent description="Android" useragent="Mozilla/5.0 (Linux; U;
Android 1.5; en-us; Android Dev Phone 1 Build/CRB21)
```

```

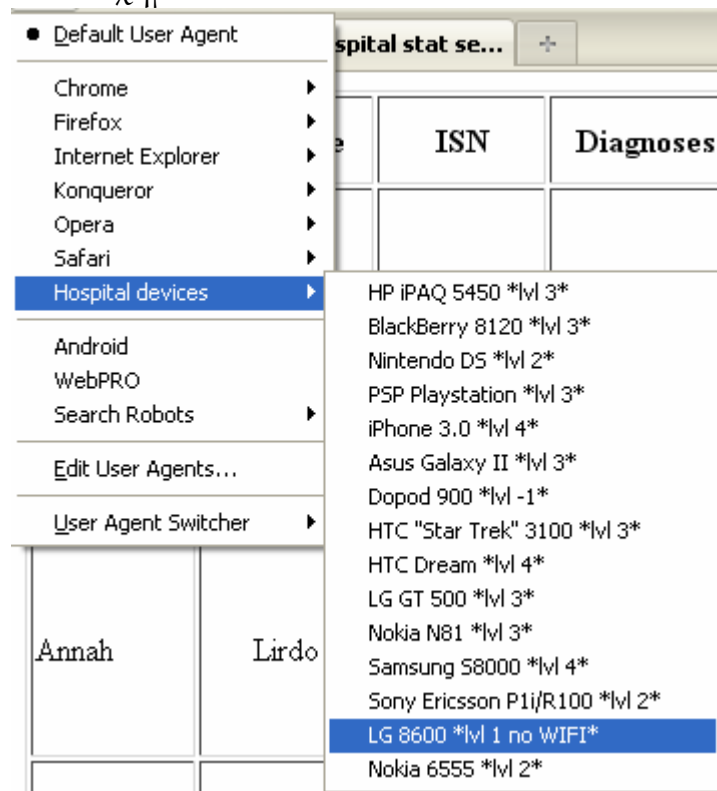
AppleWebKit/528.5+ (KHTML, like Gecko) Version/3.1.2 Mobile
Safari/525.20.1" appcodename="Mozilla" appname="Netscape"
appversion="5.0 (Windows; el)" platform="Win32" vendor=""
vendorsub=""/>
<useragent description="WebPRO" useragent="Mozilla/4.76 [en] (PalmOS;
U; WebPro3.0" appcodename="Mozilla" appname="Netscape"
appversion="5.0 (Windows; el)" platform="Win32" vendor=""
vendorsub=""/>

<folder description="Search Robots">
<useragent description="Googlebot 2.1" useragent="Mozilla/5.0
(compatible; Googlebot/2.1; +HTTP://www.google.com/bot.html)"
appcodename="" appname="" appversion="" platform="" vendor=""
vendorsub=""/>
<useragent description="Msnbot 1.1" useragent="msnbot/1.1
(+HTTP://search.msn.com/msnbot.htm)" appcodename="" appname=""
appversion="" platform="" vendor="" vendorsub=""/><useragent
description="Yahoo Slurp" useragent="Mozilla/5.0 (compatible; Yahoo!
Slurp; HTTP://help.yahoo.com/help/us/ysearch/slurp)" appcodename=""
appname="" appversion="" platform="" vendor="" vendorsub=""/>

</folder>

</useragentswitcher>
    
```

Έτσι, πετυχαίνουμε γρήγορη εναλλαγή του τύπου του user agent του πελάτη, για χρήση στον έλεγχο λειτουργίας της εφαρμογής. Το XML αυτό, αντιστοιχεί στο μενού εναλλαγής του user agent του πλοηγού που απεικονίζεται στο Σχήμα 11.

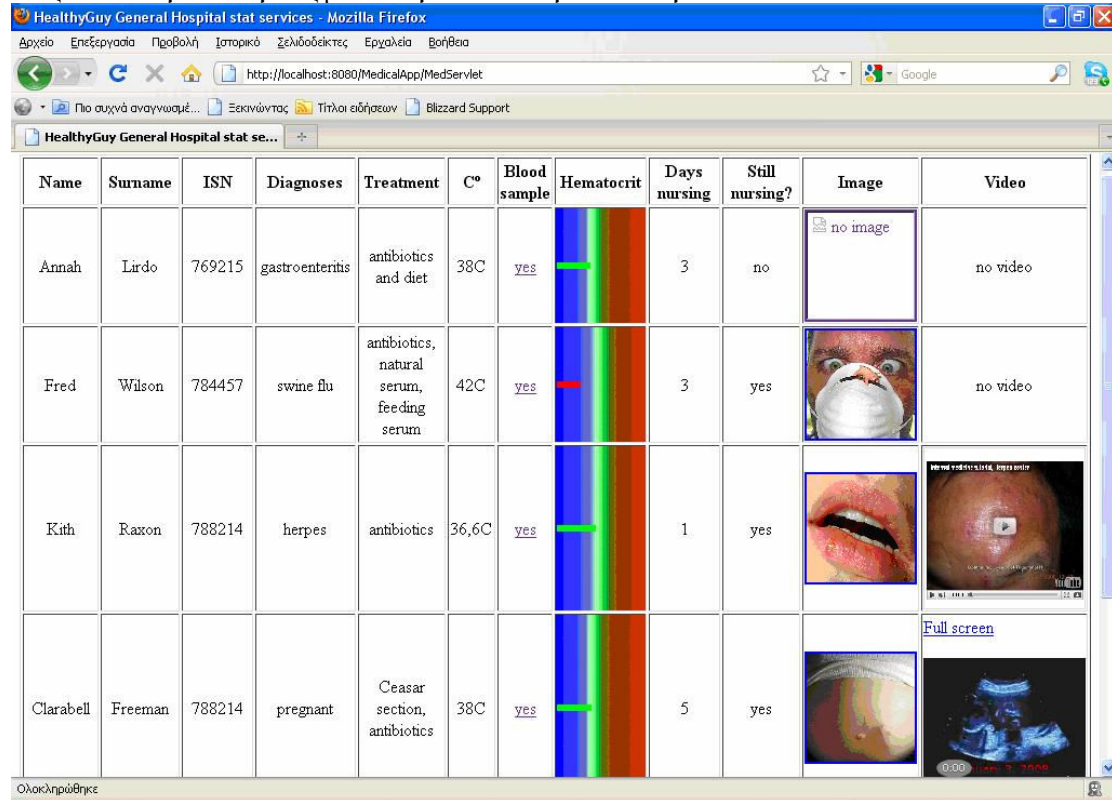


Σχήμα 11. Το Interface του πρόσθετου User agent switcher για τον πλοηγό Firefox

6.2: Επίτευξη ζητούμενου στόχου:

6.2.1: Ιστοσελίδες για H/Y

Το Σχήμα 12 απεικονίζει την παραγόμενη ιστοσελίδα στην περίπτωση που η τερματική συσκευή πελάτη είναι τύπου H/Y.



Name	Surname	ISN	Diagnoses	Treatment	C°	Blood sample	Hematocrit	Days nursing	Still nursing?	Image	Video
Annah	Lirido	769215	gastroenteritis	antibiotics and diet	38C	yes		3	no	no image	no video
Fred	Wilson	784457	swine flu	antibiotics, natural serum, feeding serum	42C	yes		3	yes		no video
Kith	Raxon	788214	herpes	antibiotics	36,6C	yes		1	yes		
Clarabell	Freeman	788214	pregnant	Ceasar section, antibiotics	38C	yes		5	yes		

Σχήμα 12. Παραγόμενη ιστοσελίδα στην περίπτωση που η τερματική συσκευή είναι τύπου H/Y

6.2.2: Ιστοσελίδες για PDA

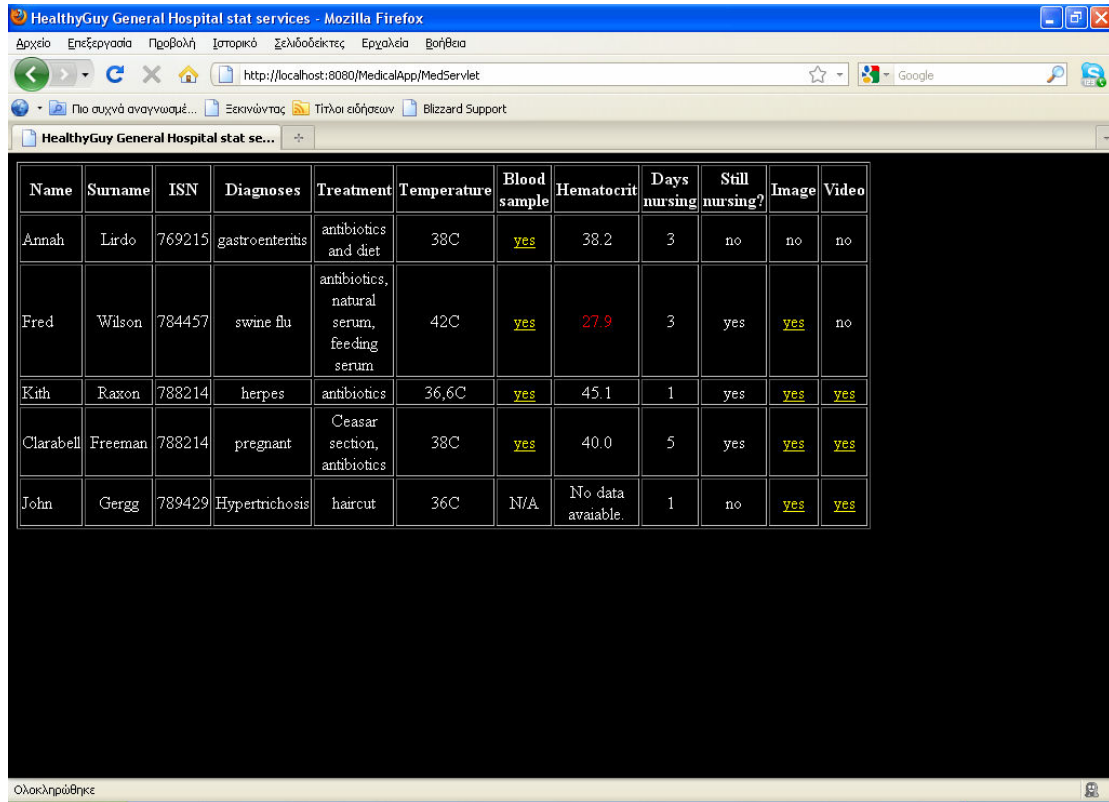
Το Σχήμα 13 απεικονίζει την παραγόμενη ιστοσελίδα στην περίπτωση που η τερματική συσκευή πελάτη είναι τύπου PDA.

Name	Surname	ISN	Diagnoses	Treatment	Temperature	blood sample	Hematocrit	Days nursing	Still nursing?	Image	Video
John	Gergg	789429	Hypertrichosis	haircut	36C	N/A	No data available.	1	no		yes
Annah	Lirido	769215	gastroenteritis	antibiotics and diet	38C	yes	L N H	3	no	no image	no
Fred	Wilson	784457	swine flu	antibiotics, natural serum, feeding serum	42C	yes	L N H	3	yes		no
Kith	Raxon	788214	herpes	antibiotics	36.6C	yes	L N H	1	yes		yes
Clarabell	Freeman	788214	pregnant	Ceasar section, antibiotics	38C	yes	L N H	5	yes		yes

Σχήμα 13. Παραγόμενη ιστοσελίδα στην περίπτωση που η τερματική συσκευή είναι τύπου PDA

6.2.3: Ιστοσελίδες για κινητά τηλέφωνα με πολυμεσικές δυνατότητες

Το Σχήμα 14 απεικονίζει την παραγόμενη ιστοσελίδα στην περίπτωση που η τερματική συσκευή πελάτη είναι κινητό τηλέφωνο με πολυμεσικές δυνατότητες.



Name	Surname	ISN	Diagnoses	Treatment	Temperature	Blood sample	Hematocrit	Days nursing	Still nursing?	Image	Video
Annah	Lirdo	769215	gastroenteritis	antibiotics and diet	38C	yes	38.2	3	no	no	no
Fred	Wilson	784457	swine flu	antibiotics, natural serum, feeding serum	42C	yes	27.9	3	yes	yes	no
Kith	Razon	788214	herpes	antibiotics	36,6C	yes	45.1	1	yes	yes	yes
Clarabell	Freeman	788214	pregnant	Ceasar section, antibiotics	38C	yes	40.0	5	yes	yes	yes
John	Gergg	789429	Hypertrichosis	haircut	36C	N/A	No data available.	1	no	yes	yes

Σχήμα 14. Παραγόμενη ιστοσελίδα για τερματική συσκευή πελάτη τύπου κινητό τηλέφωνο με πολυμεσικές δυνατότητες

6.2.4: Ιστοσελίδες για κινητά τηλέφωνα με περιορισμένες πολυμεσικές δυνατότητες

Το Σχήμα 15 απεικονίζει την παραγόμενη ιστοσελίδα στην περίπτωση που η τερματική συσκευή πελάτη είναι κινητό τηλέφωνο με περιορισμένες πολυμεσικές δυνατότητες.

Name	Surname	ISN	Diagnoses	Treatment	Temperature	Blood sample	Hematocrit	Days nursing	Still nursing?	Image
Annah	Lirdo	769215	gastroenteritis	antibiotics and diet	38C	yes	38.2	3	no	no
Fred	Wilson	784457	swine flu	antibiotics, natural serum, feeding serum	42C	yes	27.9	3	yes	yes
Kith	Razon	788214	herpes	antibiotics	36,6C	yes	45.1	1	yes	yes
Clarabell	Freeman	788214	pregnant	Ceasar section, antibiotics	38C	yes	40.0	5	yes	yes
John	Gergg	789429	Hypertrichosis	haircut	36C	N/A	No data available.	1	no	yes

Σχήμα 15. Παραγόμενη ιστοσελίδα για τερματική συσκευή πελάτη τύπου κινητό τηλέφωνο με περιορισμένες πολυμεσικές δυνατότητες

6.2.5: Ιστοσελίδες για κινητά τηλέφωνα χωρίς ή με πολύ περιορισμένες πολυμεσικές δυνατότητες

Το Σχήμα 16 απεικονίζει την παραγόμενη ιστοσελίδα στην περίπτωση που η τερματική συσκευή πελάτη είναι κινητό τηλέφωνο χωρίς ή με πολύ περιορισμένες πολυμεσικές δυνατότητες.

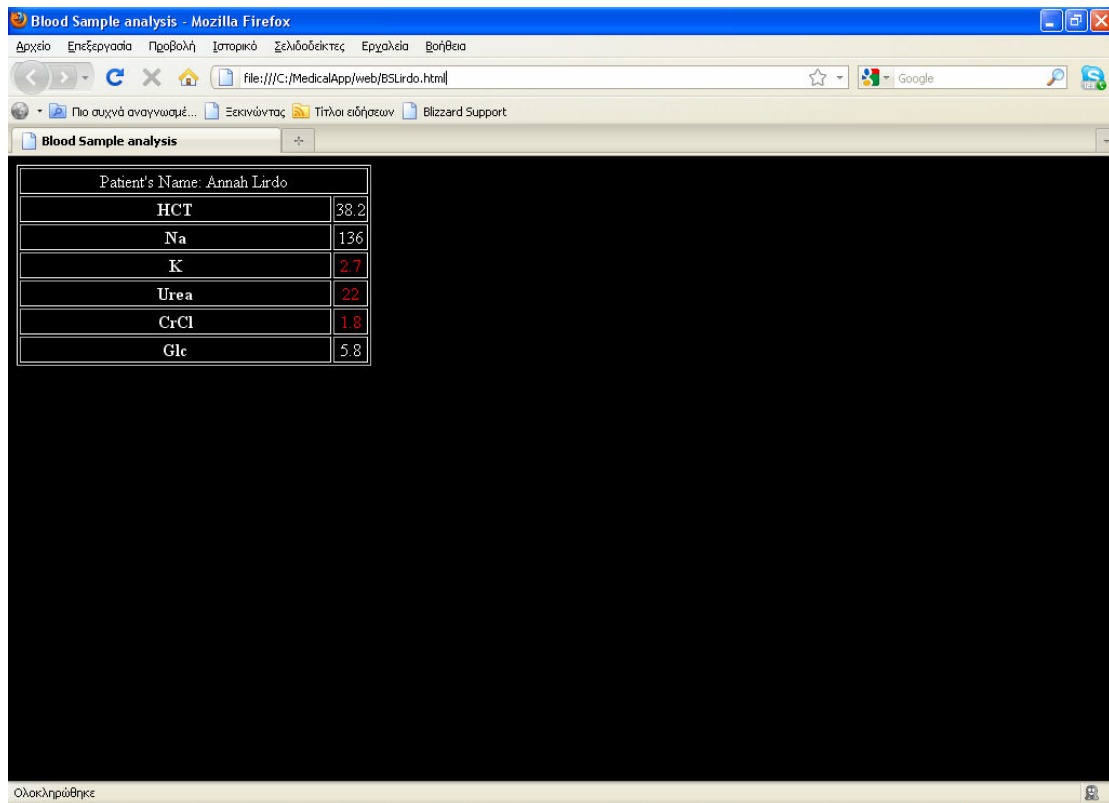
The screenshot shows a mobile browser interface with a table of patient data. The table has the following columns: Name, Surname, ISN, Diagnoses, Treatment, Temperature, Blood sample, Hematocrit, Days nursing, and Still nursing?. The data is as follows:

Name	Surname	ISN	Diagnoses	Treatment	Temperature	Blood sample	Hematocrit	Days nursing	Still nursing?
Annah	Lirdo	769215	gastroenteritis	antibiotics and diet	38C	yes	38.2	3	no
Fred	Wilson	784457	swine flu	antibiotics, natural serum, feeding serum	42C	yes	27.9	3	yes
Kith	Raxon	788214	herpes	antibiotics	36,6C	yes	45.1	1	yes
Clarabell	Freeman	788214	pregnant	Ceasar section, antibiotics	38C	yes	40.0	5	yes
John	Gergg	789429	Hypertrichosis	haircut	36C	N/A	No data available.	1	no

Σχήμα 16. Παραγόμενη ιστοσελίδα για τερματική συσκευή πελάτη τύπου κινητό τηλέφωνο χωρίς πολυμεσικές δυνατότητες

6.2.6: Αιματολογικός πίνακας ασθενή

Το Σχήμα 17 απεικονίζει τον αιματολογικό πίνακα μιας ασθενούς, συγκεκριμένα της Lirdo Annah, που πάσχει από γαστρεντερίτιδα. Παρατηρούμε πως οι τιμές που απέχουν από τα φυσιολογικά όρια, φαίνονται με κόκκινο χρώμα, όπως το Ποτάσιο (K), το Νάτριο (Na) και η Ουρία (Urea), λογικό μιας και η ασθενής έχει ακανόνιστες, συχνές κενώσεις και αφυδατώνεται γρήγορα. Το ιατρικό προσωπικό, παρακολουθώντας τις τιμές αν πλησιάζουν στα φυσιολογικά όρια, θα μπορούσε να εκτιμήσει αν βελτιώνεται η κατάστασή της, καθώς και να ορίσει η διαίτά της, όσο παραμένει στο νοσοκομείο, να είναι πλούσια σε κάλιο (κρεμμύδια, μπανάνες κ.α.) και σε υγρά (σούπες, χυμούς κ.α.).



Patient's Name: Annah Lirido	
HCT	38.2
Na	136
K	2.7
Urea	22
CrCl	1.8
Glucose	5.8

Σχήμα 17. Παραγόμενος πίνακας αιματολογικής εξέτασης της ασθενούς Lirido Annah.

6.3:Βελτιώσεις:

6.3.1:Τι μπορεί να γίνει έπειτα

Επόμενο βήμα ανάπτυξης της εφαρμογής αυτής θα ήταν να γίνει μια πιο λεπτομερής αναζήτηση στο αρχείο του WURFL, σχετικά με τις δυνατότητες των συσκευών, όπως περιγράφονται στην τεκμηρίωση, ώστε να συμπεριληφθούν κι αυτές ως μεταβλητές στον servlet.

Επιπλέον όροι επιλογής για τον τύπο της ιστοσελίδας που θα προβληθεί, μπορούν να εισαχθούν, για καλύτερο αποτέλεσμα. Σχετικά με το πολυμεσικό υλικό, αν κάποια συσκευή δεν μπορεί να αποδώσει συγκεκριμένα πολυμέσα, μπορούμε ή να υποβαθμίσουμε το επίπεδο της HTML απόδοσης της συσκευής, ή να προστεθούν XSLTs με τις αντίστοιχες μεθόδους μεταγλώττισης τους, που να μην περιέχουν τα συγκεκριμένα πρότυπα εικόνας και video.

Ανάλογα με τις δυνατότητες τήρησης αρχείου εικόνων και οπτικοακουστικού υλικού στον ηλεκτρονικό εξοπλισμό του νοσοκομείου, θα μπορούσε να γίνει μια επιλογή «επιστημονικών» ιστοσελίδων, που περιέχουν υψηλής ποιότητας πολυμεσικά στοιχεία για υποστήριξη τόσο του ιατρικού προσωπικού στη διάγνωση της ασθένειας ενός πάσχοντα, όσο των διοικητικών υπηρεσιών του νοσοκομείου, που ζητάνε αναφορικές λεπτομέρειες της διάγνωσης του ασθενή.

6.3.2:Εισαγωγή νέων στοιχείων

Η μηχανοργάνωση διαδραματίζει σημαντικό ρόλο στην εκπλήρωση της αποστολή ενός νοσοκομείου να προλαμβάνει και να θεραπεύει τις ασθένειες των πολιτών. Βασικός παράγοντας της προσπάθειας αυτής, είναι η ταχεία και τεκμηριωμένη πληροφόρηση του ιατρικού προσωπικού σε θέματα διάγνωσης, εποπτείας και καταγραφής των στοιχείων ενός ασθενούς. Λόγω του ότι ολοένα νέες ανάγκες προκύπτουν στη λειτουργία του νοσοκομείου και καινούργιες τεχνολογίες εντάσσονται στην ίαση και υποστήριξη των ασθενών, ένα σύστημα μηχανοργάνωσης πρέπει να είναι ευέλικτο και επεκτάσιμο ώστε να μπορεί να εντάξει καινούργια δεδομένα ασθενών, χωρίς να συμβιβάζεται με τη διάθεση λιγότερων πόρων συστήματος από αυτούς που απαιτούνται.

Όπως με το παράδειγμα του πεδίου ER (Εντατική Μονάδα) στο XML των ασθενών, που υπάρχει για μελλοντική χρήση, έτσι πρέπει να σκέφτονται οι μηχανικοί πληροφορικής, που θα αναπτύξουν περαιτέρω την παρούσα εφαρμογή δυναμικής παρουσίασης ιστοσελίδων, ώστε να μην στενέψουν ή να στερήσουν την ευελιξία και ελευθερία της ριζικής αλλαγής του τρόπου εισαγωγής και επεξεργασίας των δεδομένων ασθενών.

Οι τεχνολογίες ανοιχτού λογισμικού δεν πρέπει να εξαιρούνται από τη μελέτη μηχανοργάνωσης ενός νοσοκομείου, για τον λόγο που μπορούν να προσφέρουν ευελιξία, ακόμα και στην καρδιά της εφαρμογής. Ακόμη, όμως, κι αν χρησιμοποιηθούν εφαρμογές που προστατεύουν τον κώδικα τους από παραμετροποίηση, θα πρέπει να υπάρχει γνώση στον τρόπο λειτουργίας τους, ώστε να αποφεύγονται ατέλειες και προβλήματα στην εύρυθμη λειτουργία των πληροφοριακών συστημάτων του νοσοκομείου.

ΜΕΡΟΣ 3^ο

ΣΥΝΟΨΗ

- Κεφάλαιο 7^ο: Προβλήματα
- Κεφάλαιο 8^ο: Περαιτέρω παραδείγματα

Κεφάλαιο 7^ο - Προβλήματα

7.1:Ελλείψεις του WURFL:

7.1.1:Το αιώνιο πρόβλημα της δωρεάν, ανοιχτής κοινότητας

Η προσπάθεια του WURFL εξαρχής βασίστηκε, και συνεχίζει να βασίζεται, στο μεράκι και τη συνεχή προσπάθεια ατόμων, σε όλη τη διαδικτυακή κοινότητα, που αφιερώνουν χρόνο και προσπάθεια, στην ανάπτυξή του.

Επειδή έχουμε συνηθίσει, τα τελευταία χρόνια, να θεωρούμε δεδομένη την αξιοπιστία των εφαρμογών και των τεχνολογιών που έχουν πίσω τους τη διαρκή υποστήριξη μιας εταιρίας, η οποία αφιερώνει χρηματικό κεφάλαιο για αυτή τη δουλειά, απαιτούμε και από τις προσπάθειες τέτοιων ελεύθερων κοινοτήτων να αποδίδουν καρπούς, αντάξιους, αν όχι καλύτερους, από αυτούς των εταιριών. Κάτι τέτοιο δεν μπορεί να συμβαίνει συνέχεια, όμως, αφού δεν υπάρχει χρηματική υποστήριξη, τις περισσότερες φορές, των ατόμων αυτών, ώστε να εκμεταλλεύονται το χρόνο τους και την προσπάθειά τους, χωρίς απόσπαση από την ανάγκη κάλυψης των βιοποριστικών τους αναγκών.

Θα πρέπει να το έχουμε υπ' όψιν αυτό, στη συνέχεια αυτού του μέρους της εργασίας μας, γιατί η κριτική αντιμετώπιση των τεχνολογιών που χρησιμοποιήθηκαν στην παρούσα εφαρμογή, θα είναι μονόπλευρη, αν κρίνουμε, καθαρά με τεχνοκρατικά κριτήρια.

7.1.2:Capabilities

Οι δυνατότητες των συσκευών που παρουσιάζονται στο WURFL, πολλές φορές είναι ελλιπείς, ή και ψευδείς. Αυτό γίνεται εξαιτίας λαθών μετάφρασης των UAProf, ή λανθασμένης οργάνωσης του αρχείου XML. Επιπλέον, ασαφείς οδηγίες δίνονται πολλές φορές, ειδικά από συσκευές που δεν είναι ευρέως γνωστές στην αγορά, ή χρησιμοποιείται το σύστημα της απόκρυψης λειτουργιών μιας συσκευής, για επιχειρηματικούς κυρίως λόγους. Παράλληλα, νέες δυνατότητες προστίθενται στις ήδη υπάρχουσες ενός μοντέλου συσκευής, στα πλαίσια ανανέωσης των τεχνολογιών, με αποτέλεσμα να προωθείται στην αγορά η συσκευή αυτή ως νέο μοντέλο, αλλά διατηρώντας τον ίδιο user agent, γεγονός που μπορεί να έχει ως συνέπεια, να μην προβλέπονται στο αρχείο wurfl οι πλήρεις δυνατότητες που ενδεχομένως έχει η συσκευή.

Η λίστα των δυνατοτήτων στη Deprecated ομάδα, είναι μια ένδειξη πως δεν έχει αποφασιστεί ακόμα, η τελική μορφή προβολής των στοιχείων μιας συσκευής. Αλλαγές στην δομή οργάνωσης αφήνουν

μετέωρα τα μέλη της κοινότητας, στο να προσαρμόσουν τις πληροφορίες, όπως πιστεύει ο καθ' ένας τους ότι είναι σωστότερο.

7.1.3:Patch

Ευτυχώς, ανά τακτά διαστήματα, ενημερωμένη έκδοση του αρχείου που έχει αποφασιστεί από το μεγαλύτερο μέρος της κοινότητας ως τεκμηριωμένο, ανακοινώνεται στην κεντρική ιστοσελίδα. Το πρόβλημα συνεχίζει και υπάρχει όμως, στο ότι μια εφαρμογή που κάνει χρήση αυτού του αρχείου, πρέπει να μεριμνά στο να το ενημερώνει συχνά.

7.2:User agents και κακή χρήση αυτών:

7.2.1: Robots, web crawlers, downloading tools, linkers, web filterers και spammers

Παράλληλα με τους χρηστικούς user agents που μελετήσαμε στα προηγούμενα κεφάλαια, υπάρχουν ειδικού τύπου user agents που χρησιμοποιούνται από αυτοματοποιημένες διαδικτυακές εφαρμογές για την συλλογή στοιχείων, την ενημέρωση των μηχανών αναζήτησης ή για κακές, μη αποδεκτές προθέσεις.

Αυτοί οι user agents είναι ικανοί να ξεγελάσουν τους διάφορους servers, καθώς περιέχουν πολλές φορές παρόμοιες πληροφορίες με τους user agents του WURFL. Ένα σύστημα πρέπει να διαθέτει αποδοτικά συστήματα απομάκρυνσης ή απαγόρευσης πρόσβασης τέτοιων εφαρμογών, που ζητάνε πρόσβαση στις πληροφορίες, χωρίς να αποκλείει, όμως, τους χρήστες στους οποίους απευθύνεται. Μια τέτοια προσπάθεια απαιτεί λεπτομερή μελέτη και γνώση επί του θέματος της αναγνώρισης των user agents και δεν πρέπει να υλοποιείται επιπόλαια σε μια εφαρμογή.

7.2.2:Παρόμοιες κινητές συσκευές

Στην οργάνωση του WURFL, πολλές φορές συναντάμε παρόμοιους user agents για έναν τύπο συσκευής. Αυτό γίνεται γιατί κάθε μοντέλο κινητής ασύρματης συσκευής, έχει μια σειρά χαρακτηριστικών που προστίθενται σε κάθε νέα έκδοση της συσκευής, όπως περιγράψαμε παραπάνω. Παρ' όλη την προσπάθεια που κάνουν τα μέλη της κοινότητας του WURFL για πρόβλεψη αυτού του ενδεχομένου, είτε με νέες ομάδες είτε με συγκεκριμένες αναγνωριστικές δυνατότητες αυτών, πολλές εταιρίες κατασκευής κινητών συσκευών μπλέκουν τόσο πολύ τα πράγματα, που πια είναι αδύνατον να τηρηθεί μια συνοχή στην καταγραφή.

7.2.3: Έλλειψη ασφάλειας

Το αρχείο του WURFL είναι διαθέσιμο σε οποιονδήποτε το ζητήσει για μεταφόρτωση. Ένα μεγάλο θέμα που εγείρεται είναι να μελετήσει κάποιος τους διάφορους user agents, ώστε να τους χρησιμοποιήσει σε συστήματα πιστοποίησης μέσω αυτών, για σκοπούς παραπλάνησης του συστήματος και πρόσβασης σε πληροφορίες που δεν είναι εξουσιοδοτημένος. Έχουμε το παράδειγμα του user agent switcher, του προγράμματος που χρησιμοποιούμε, για αλλαγή του εμφανιζόμενου user agent του πλοηγού, στα πλαίσια εξομοίωσης της εφαρμογής MedApp.

Λόγω της φύσης της ιδέας κοινοποίησης και συγγραφής του WURFL αρχείου (ανοιχτό λογισμικό), η λύση να υλοποιηθεί με κλειστό κώδικα είναι μη αποδεκτή. Όσοι κατασκευάζουν συστήματα προβολής πληροφορίας που βασίζονται στο WURFL, πρέπει να βρουν τρόπους πιστοποίησης του τελικού χρήστη, για το αν είναι εξουσιοδοτημένος να δει την αιτηθείσα πληροφορία, ανεξάρτητους από την λειτουργία του WURFL.

7.2.4: Συμπεράσματα

Στα πλαίσια της δυναμικής προσαρμογής του περιεχομένου ιστοσελίδων, η λύση του WURFL είναι γεμάτη υποσχέσεις. Στην παρούσα φάση, δεν μπορεί να γίνει καλή χρήση του, λόγω της κατακερματισμένης πληροφορίας που δίνει για τις κινητές συσκευές. Εντούτοις, είναι αρκετά λειτουργικό, καθώς υποστηρίζεται από τα μέλη της κοινότητας, ώστε να ανανεώνονται οι εγγραφές σε τακτά χρονικά διαστήματα, τέτοια που να προλαμβάνουν τις εξελίξεις της τεχνολογίας.

Επίσης, λόγω του XML χαρακτήρα που έχει για την δομή των δυνατοτήτων των καταχωρημένων συσκευών, είναι άμεσα και εύκολα επεξεργάσιμο, σε περίπτωση που αλλάξουν τα πρότυπα ή θέλουμε να εισάγουμε νέα, ίσως και δικά μας.

Στον τομέα της πιστοποίησης και ασφάλειας, είναι προτιμότερο να δοθεί έμφαση στις παράπλευρες τεχνολογίες που λειτουργούν παράλληλα με το πρωτόκολλο HTTP, παρά να αναπτυχθεί κάποιο πρότυπο πιστοποίησης μέσω των user agents, λόγω του ότι οι πληροφορίες αυτές, εξυπηρετούν συγκεκριμένες ομάδες συσκευών και όχι μεμονωμένες συσκευές.

Κεφάλαιο 8^ο – Περαιτέρω παραδείγματα

8.1:Υπηρεσίες:

8.1.1:Μετακινήσεις

Οι εταιρίες, καθώς και οι δημόσιες υπηρεσίες, θα πρέπει να εναρμονιστούν στους καιρούς μας, αναγνωρίζοντας ότι πολύς κόσμος χρειάζεται ενημέρωση, σχετικά με την ποιότητα και την κάλυψη των αποστάσεων, που μπορούν να προσφέρουν τα μέσα μαζικής μεταφοράς τους.

Με την χρήση των τεχνολογιών που κάνει το μεγαλύτερο μέρος των ατόμων, που χρησιμοποιεί συχνά υπηρεσίες μετακίνησης, θα μπορούσαν να γίνουν εφαρμογές που, να δίνουν στίγμα στους χρήστες για το που βρίσκεται το όχημα ή σκάφος που επιθυμούν να ταξιδέψουν, να παρέχουν πληροφορίες για το κόμιστρο, τα δρομολόγια, τη διάρκεια ταξιδιού κ.α., όλα αυτά εμπλουτισμένα με πολυμεσικές πληροφορίες που θα παραμετροποιούνται ανάλογα με την τερματική συσκευή που χρησιμοποιεί ο πελάτης των μετακινήσεων.

Επίσης, λόγω του ότι σε ένα ταξίδι, ο χρόνος μετακίνησης χαρακτηρίζεται από τους επιβάτες ως «νεκρός» χρόνος, θα μπορούσαν τα οχήματα ή σκάφη, να παρέχουν δικούς τους, ασύρματους Servers, για την κοινοποίηση στοιχείων όσων αφορά το ταξίδι, την πρόσβαση στο διαδίκτυο, ή σε διαφημίσεις και υπηρεσίες τηλεθέασης κ.α.

8.1.2:Εξυπηρέτηση του πολίτη

Στους δαιδαλώδεις δρόμους του λαβύρινθου της γραφειοκρατίας, η σωστή ενημέρωση και η έγκαιρη εκκαθάριση των εκκρεμοτήτων του πολίτη, είναι βασικά στοιχεία για την γρήγορη και αποτελεσματική εξυπηρέτησή του. Σε αυτό, μπορεί να συμβάλει το κράτος, με το να στέλνει στις τερματικές συσκευές των πολιτών που εισέρχονται σε μια δημόσια υπηρεσία, πλήρες οργανόγραμμα των υπαλλήλων, με τις αρμοδιότητες τους, για την αποφυγή σύγχυσης και ταλαιπωρίας των πολιτών. Επίσης, το ωράριο εργασίας των υπαλλήλων, όπως κι έναν υποτυπώδη «πίνακα ανακοινώσεων», για ανάρτηση πληροφοριών που μπορούν να χρειαστούν οι πολίτες.

8.1.3:Κοινοποίηση πολιτογραφικών στοιχείων

Μια χώρα, οφείλει να κοινοποιεί τις δράσεις της και τις αποφάσεις τις για κοινά θέματα ενδιαφέροντος, όπως άμυνας, οργάνωσης, διοίκησης, οικονομίας, εργασίας, υπηρεσιών κ.α. Η διαφορά με τις παλιές εποχές, με τη μονόδρομη πληροφόρηση που

πραγματοποιούταν μέσω τεχνολογιών όπως η τηλεόραση και το ραδιόφωνο, είναι ότι οι πολίτες μιας χώρας θέλουν να συμβάλουν άμεσα στα κοινά του κράτους τους, χωρίς γεωγραφικούς και τεχνολογικούς περιορισμούς. Για να το κάνουν αυτό, χρειάζονται μια αμφίδρομη πληροφόρηση, υπηρεσίες που να γνωστοποιούν στοιχεία του κράτους και να προσφέρουν πληροφορίες σχετικά με την δομή και λειτουργία της χώρας τους.

Εκτός από τους Η/Υ, που μέχρι τώρα είναι λίγο προβληματική η πρόσβαση στο διαδίκτυο, λόγω του τεχνολογικού και οικονομικού θέματος που εγείρει η εγκατάσταση μιας DSL γραμμής σε μια οικία, φτηνές μαζικές ασύρματες τεχνολογίες μπορούν να εντάξουν το μεγαλύτερο μέρος των πολιτών στην δράση της συμμετοχής στα κοινά. Βέβαια, πολλές φορές, διατάγματα, νόμοι, οργανογράμματα, οδικοί και τοπογραφικοί χάρτες, έντυπα, κοινοποιήσεις, διαγωνισμοί και άλλα στοιχεία, είναι καλύτερα να προβάλλονται ως πολυμεσικό υλικό, παρά ως κείμενο. Η δυνατότητα να μπορούν όλοι να έχουν πρόσβαση στο υλικό αυτό, ανεξαρτήτως των δυνατοτήτων της τερματικής τους συσκευής, είναι βασική στην όλη προσπάθεια.

8.2: Ψυχαγωγία:

8.2.1: Συλλογικές προσομοιώσεις (Collective simulations)

Μέσω των διαφόρων πλαισίων εργασίας πάνω σε γνωστικά αντικείμενα όπως η μετάδοση και εμπέδωση γνώσης με την χρήση συλλογικών προσομοιώσεων, ειδικά σε μικρές ηλικίες, μπορεί να γίνει πολύ πιο εύκολη, αν εισαχθούν τεχνολογίες, προσιτές σε πολύ νεαρά άτομα, που να έχουν δυνατότητα να απεικονίζουν πολυμεσικές πληροφορίες.

Μια πολύ καλή προσπάθεια είναι η ένταξη των παιδιών των χωρών του τρίτου κόσμου, στην κοινωνία της πληροφορικής, με τη χρήση πολύ φτηνών laptops, ειδικά σχεδιασμένων να πληρούν ανάγκες για τα σκληροτράχηλα εκείνα μέρη, που και οι στοιχειώδεις ανάγκες, πολλές φορές, ικανοποιούνται δύσκολα ή καθόλου.

Τα παιδιά αυτά, λαμβάνουν μέρος σε παιχνίδια συλλογικών προσομοιώσεων, αντλώντας στοιχεία από το διαδίκτυο ή τα μέσα αποθήκευσης που τους δίνονται, σχετικά με θέματα οικονομίας, διοίκησης, οικολογικά θέματα, κοινωνικά κ.α. Κατόπιν, οι παιδαγωγοί θέτουν ρόλους στα μέλη της ομάδας τα οποία αρχίζουν να αλληλεπιδρούν, ώστε να γίνει κοινωνική εκπαίδευση των παιδιών, στο τι αρμοδιότητες και τι υποχρεώσεις φέρει κάθε μέλος αυτής της μικρής, κοινωνικής προσομοίωσης.

Παράλληλα, γνώσεις θεωρητικής και πρακτικής φύσεως μπορούν να αποκομισθούν από τους μαθητές, μέσω αυτών των παιγνίων. Οι μαθητές καλούνται να αναλάβουν ρόλους για θέματα που μπορεί να μην έχουν την δυνατότητα να ασχοληθούν ποτέ, αλλά θα κάνουν κτήμα τον τρόπο αντιμετώπισης τέτοιων θεμάτων, που θα τους βοηθήσει να δρουν με διαύγεια και σύνεση στα καθημερινά τους προβλήματα.

Εκτός βέβαια από τους μαθητές των σχολείων του τρίτου κόσμου, ανάλογες ενέργειες γίνονται και σε μαθητές των αναπτυσσόμενων, καθώς και των αναπτυγμένων χωρών. Αν και οι περισσότερες συλλογικές προσομοιώσεις στοχεύουν περισσότερο σε κοινωνικές έρευνες, εντούτοις πολλά σχολεία προσπαθούν να υιοθετήσουν ένα τέτοιο μοντέλο μάθησης, στα πλαίσια της ευρύτερης γνώσης που μπορεί να δοθεί σε ένα μαθητή, στα πλαίσια του σχολείου.[27]

Οι κινητές συσκευές, με τις τελευταίες εξελίξεις στην τεχνολογία της διεπαφής της συσκευής με τον χρήστη, έχουν γίνει πολύ φιλικές στο θέμα διαχείρισης τους. Οι μαθητές μπορούν να λαμβάνουν εύκολα το πολυμεσικό υλικό που απαιτείται για τη διενέργεια κάποιου πειράματος, ή στα πλαίσια ενός παιχνιδιού, αρκεί αυτό να απεικονίζεται σωστά στην τερματική τους συσκευή.

8.2.2: «Κυνήγι θυσσαυρού» και λοιπά αστικά παιχνίδια

Κατά καιρούς, βλέπουμε φαινόμενα οργάνωσης παιγνίων, που καλούν και άτομα από ομάδες μεγαλύτερης ηλικίας από αυτές που φανταζόμαστε πως θα μπορούσαν να λάβουν μέρος. Χαρακτηριστικό παράδειγμα, ένα από τα τελευταία τέτοια παιχνίδια, το Urban Dead. Πληροφορίες για αυτό βρίσκονται στη διεύθυνση ιστοσελίδας [HTTP://www.urbandead.com/](http://www.urbandead.com/), καθώς και τα λειτουργικά του στοιχεία.

Σε αυτό το παιχνίδι, αποφασίζεται από την εφαρμογή που περιέχει ο ιστοχώρος ποιος θα είναι αρχικά ο νεκροζώντανος, από μια ομάδα παιχτών, που θα ξεκινά να μολύνει τα υπόλοιπα μέλη. Το πεδίο δράσης των παιχτών είναι όλη η πόλη που διαμένουν οι παίχτες και απώτερος σκοπός είναι να εντοπιστεί και να σκοτωθεί ο αρχικός νεκροζώντανος με χρήση πιστολιού νερού ή έκρηξης νερόβομβας. Ο αρχικός νεκροζώντανος μολύνει τους παίκτες με απλό άγγιγμα, οπότε μετατρέπονται και αυτοί σε νεκροζώντανους με ικανότητα μόλυνσης, αλλά με τον όρο να φέρουν μια λουρίδα ρούχου στο κεφάλι, για άμεση αναγνώριση. Η έκβαση του παιχνιδιού μπορεί να έχει δύο αποτελέσματα, ή να «σκοτωθεί» ο αρχικός νεκροζώντανος, ή να μολυνθούν όλοι οι παίχτες μιας ομάδος.

Αυτό το παιχνίδι απαιτεί ενημέρωση στο σύστημα του για το ποιες αλλαγές γίνονται στην κατάσταση των παιχτών, οπότε μια τεχνολογία που μπορεί να δίνει πολυμεσικό υλικό σε κινητές συσκευές, μπορεί να διευκολύνει τη διαδικασία, καθώς να εισάγει και άλλα

χαρακτηριστικά που μπορούν να το κάνουν μοναδικό στο είδος του, όπως η επίλυση γρίφων που θα παρουσιάζονται σε πολυμεσική μορφή, ο εντοπισμός στον χάρτη των παιχτών κ.α.

Άλλο παράδειγμα παιχνιδιού, είναι το γνωστό «κυνήγι θυσασαυρού». Σε αυτό, μια σειρά από στοιχεία, μουσικά, εικαστικά, λογοτεχνικά κ.α. απαιτούν από τους παίκτες, να τα συνδυάσουν με χαρακτηριστικές περιοχές της πόλης που διαμένουν, ώστε να εντοπίσουν ένα αντικείμενο ή πρόσωπο, που ζητείται από το σενάριο που τρέχει, σαν κεντρική ιστορία του παιχνιδιού. Αυτό το παιχνίδι, θα μπορούσε να στέλνει τα στοιχεία, ως πολυμεσικά μηνύματα στις συσκευές των παιχτών, παρά να εκπέμπονται ή να δίνονται σε κοινούς χώρους, γεγονός που συνήθως καθιστά προβληματική την άμεση και έγκυρη πληροφόρηση των παιχτών.

8.2.3: Προσωπικά παιχνίδια

Στις συσκευές αυτές, μελετώνται υλοποιήσεις διαδικτυακών παιχνιδιών για τερματικές συσκευές με δυνατότητες σταθερών Η/Υ. Με την κατάλληλη παραμετροποίηση, όλοι μπορούν να απολαύσουν ένα παιχνίδι σε μια διαδικτυακή κοινωνία, χωρίς να απαιτείται η πρόσβαση σε υπολογιστή μεγάλων δυνατοτήτων.

Παράλληλα, παιχνίδια σε τεχνολογίες Flash, Shockwave κ.α., μπορούν να γνωστοποιηθούν σε κοινό που διαθέτει συσκευές με τέτοιες δυνατότητες, εξαιρώντας τους χρήστες συσκευών χωρίς αυτές τις δυνατότητες. Για τους τελευταίους, εναλλακτικές λύσεις ή χαμηλής απεικονιστικής ποιότητας εκδόσεις των παιχνιδιών, μπορούν να προταθούν, μέσω μιας δυναμικά παραγόμενης ανακοίνωσης.

8.3: Μηχανογράφηση:

Οι παραμετροποιήσεις σε πακέτα ψηφιακής οργάνωσης εταιριών και ιδρυμάτων, καθώς και η παραγωγή λογισμικού για τις ανάγκες ενός ιδρύματος ή εταιρίας, πρέπει να έχουν γνώση πάνω στις δυνατότητες του εξοπλισμού που θα χρησιμοποιηθεί απ' αυτή. Αν είναι να περιέχονται στον εξοπλισμό μιας επιχείρησης κινητές συσκευές, διαφορετικές από τα laptop και τα netbook, τότε πρέπει να υπάρξει πρόβλεψη και γι' αυτούς τους χρήστες στη μελέτη της εφαρμογής.

Λόγω των περιορισμένων δυνατοτήτων τους, το ζητούμενο είναι να μπορούν, να αποδώσουν πλήρεις και τεκμηριωμένες πληροφορίες στους χρήστες τους. Αν για κάποιο λόγο δεν μπορεί μια συσκευή, εξαιτίας των δυνατοτήτων της, να αποδώσει μια πληροφορία ή να την εμφανίζει κατακερματισμένη, μπορεί να δημιουργηθεί μεγάλο πρόβλημα από την απόδοση της εφαρμογής στο ενεργητικό της επιχείρησης.

Θα πρέπει να αποφασιστεί ο τύπος της πληροφορίας πως θα δομείται και πως η εφαρμογή θα αξιοποιεί τα δεδομένα που της δίνονται από μια βάση δεδομένων. Αν είναι σε μορφή κειμένου, πρέπει να δοθεί προσοχή στην ποιότητα απόδοσης των μορφοποιήσεων του κειμένου, αν, δηλαδή, η συσκευή αναγνωρίζει ειδικούς χαρακτήρες, αν περιέχεται εφαρμογή ανάγνωσης της συγκεκριμένης μορφής του κειμένου, αν επιδέχεται τροποποίησης, αν η οθόνη της συσκευής είναι σε θέση να αποδώσει ολόκληρο ή όχι το κείμενο, καθώς και το χρωματικό φάσμα που ενδεχομένως χρησιμοποιείται, κλπ.

Παράλληλα, πληροφορία μπορεί να δίδεται σε πολυμεσική μορφή, οπότε πρέπει να γνωρίζουμε αν μια συσκευή μπορεί να απεικονίζει εικόνες, να αναπαράγει μουσική ή video, να αναγνωρίζει ειδικές μορφές πολυμέσων, όπως streaming video κ.α.

Συνήθως, μια πληροφορία δομείται βάση μιας σηματοδοτικής γλώσσας, όπως η HTML, η XML κ.α. Βασικό είναι, να γνωρίζουμε αν η γλώσσα αυτή, υποστηρίζεται από τις συσκευές που θα συμπεριλάβουμε στην μελέτη του εξοπλισμού της επιχείρησης, καθώς και να γνωρίζουμε την ποιότητα της δυνατότητας αυτής.

Αφού μελετηθεί η προβολή της πληροφορίας, πρέπει να αποφασιστεί αν η συσκευή θα εκτελεί ολόκληρη την εφαρμογή ή συγκεκριμένο μέρος αυτής ειδικά διαμορφωμένο για αυτήν, ή καθόλου και απλά θα προβάλλει την πληροφορία. Η πρόβλεψη αυτή μπορεί να επιλύσει επεξεργαστικά προβλήματα, καθώς και να κάνει οικονομία χρόνου στις λειτουργίες της εφαρμογής.

Πολλές τέτοιες συσκευές σήμερα, έχουν τη δυνατότητα να φιλοξενούν δημοφιλή λειτουργικά συστήματα, όπως τα Windows CE. Βασιζόμενη εκεί, η μηχανοργάνωση μπορεί να αναπτύξει κοινή εφαρμογή για όλες τις τερματικές συσκευές της επιχείρησης, χωρίς προβλήματα προτυποποίησης και ασυμβατότητας. Διαφορετικά, θα πρέπει να κατανέμει τους επεξεργαστικούς ρόλους των συσκευών στο σύνολο της εφαρμογής. Τέλος, όπως είπαμε προηγουμένως, μια τέτοια συσκευή μπορεί απλά να προβάλλει πληροφορία.

Παράρτημα

Βιβλιογραφία:

- [1].[HTTP://www.developershome.com/wap/detection/](http://www.developershome.com/wap/detection/)
- [2].O'Reillys Java in a nutshell
- [3].O'Reillys Java language reference
- [4].O'Reillys Java servlet programming
- [5].O'Reillys HTML the definitive guide
- [6].O'Reillys CGI programming
- [7].O'Reillys HTML & XHTML the definitive guide
- [8].O'Reillys Cascading Style Sheets the definitive guide
- [9].O'Reillys Information architecture of the word wide web
- [10].O'Reillys Sax2
- [11].O'Reillys XML in a nutshell
- [12].O'Reillys Java & XML 2nd edition
- [13].O'Reillys Java & XSLT
- [14].O'Reillys XSLT
- [15].O'Reillys XML schema
- [16].[HTTP://stackoverflow.com/](http://stackoverflow.com/)
- [17].[HTTP://www.w3schools.com/](http://www.w3schools.com/)
- [18].Java - How to program, sixth edition - H. M. Deitel - Deitel & Associates, Inc., P. J. Deitel - Deitel & Associates, Inc
- [19].Java software solutions, 4th edition – Wesley Addison
- [20].[HTTP://java.sun.com/developer/Books/javaserverpages/cservletsjsp/index.html](http://java.sun.com/developer/Books/javaserverpages/cservletsjsp/index.html)
- [21].[HTTP://www.w3.org/](http://www.w3.org/)
- [22].[HTTP://www.w3.org/Mobile/](http://www.w3.org/Mobile/)
- [23].[HTTP://www.openmobilealliance.org/default.aspx](http://www.openmobilealliance.org/default.aspx)
- [24].[HTTP://www.3gpp.org/](http://www.3gpp.org/)
- [25].[HTTP://www.ietf.org/](http://www.ietf.org/)
- [26].[HTTP://wurfl.sourceforge.net/](http://wurfl.sourceforge.net/)
- [27]. Mr. Vetro: A Collective Simulation Framework - Alexander Repenning, Andri Ioannidou
- [28].[HTTP://www.gnu.org/licenses/gpl-3.0.txt](http://www.gnu.org/licenses/gpl-3.0.txt)
- [29].[HTTP://www.sun.com/cddl/cddl.html](http://www.sun.com/cddl/cddl.html)
- [30].[HTTP://www.wapforum.org/what/technical_1_0.htm](http://www.wapforum.org/what/technical_1_0.htm)
- [31].[HTTP://freemainsourcesoftware.org/index.php?title=Main_Page](http://freemainsourcesoftware.org/index.php?title=Main_Page)