



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

**ΑΝΑΠΤΥΞΗ
ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΜΕ ΧΡΗΣΗ DJANGO ΚΑΙ ΡΥΘΜΟΝ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΘΟΔΩΡΗ ΓΡΕΑΣΙΔΗ

Βόλος, Απρίλιος 2009

Η σελίδα αυτή είναι σκόπιμα λευκή.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

**ΑΝΑΠΤΥΞΗ
ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΜΕ ΧΡΗΣΗ DJANGO ΚΑΙ ΡΥΘΗΟΝ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΘΟΔΩΡΗ ΓΡΕΑΣΙΔΗ

Επιβλέπων : Μανόλης Βάβαλης
Αναπληρωτής Καθηγητής Τ.Μ.Η.Υ.Τ.Δ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή τον Οκτώβριου 2009.

(Υπογραφή)

.....
Ηλίας Χούστης
Καθηγητής Τ.Μ.Η.Υ.Τ.Δ.

(Υπογραφή)

.....
Μανόλης Βάβαλης
Αναπληρωτής Καθηγητής Τ.Μ.Η.Υ.Τ.Δ.

Βόλος, Οκτώβριος 2009

(Υπογραφή)

.....

ΘΟΔΩΡΗΣ ΓΡΕΑΣΙΔΗΣ

Διπλωματούχος Μηχανικός Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων
Πανεπιστημίου Θεσσαλίας

© 2009 – All rights reserved

Περίληψη

Τα τελευταία χρόνια η μορφή του Internet καθώς και οι δυνατότητες που αυτό μπορεί να προσφέρει στους χρήστες του έχουν αλλάξει κατά πολύ. Νέες τεχνολογίες και σχεδιαστικές τεχνικές εμφανίστηκαν και στη συνέχεια υιοθετήθηκαν από τους σχεδιαστές ιστοσελίδων κάνοντας το διαδίκτυο διαδραστικό και επιτρέποντας τη μετέπειτα εμφάνιση των Web εφαρμογών. Η ταχύτητα των αλλαγών της μορφής του διαδικτύου, με έκανε να θέλω να εξετάσω τις τεχνολογίες που αξιοποιούνταν και πιο συγκεκριμένα, τις δυνατότητες που η πλατφόρμα Django⁴ προσφέρει στην ανάπτυξη υψηλής ποιότητας διαδικτυακών εφαρμογών.

Ψάχνοντας το είδος της εφαρμογής που θα στοχεύαμε, παρατηρήσαμε ότι τα προγράμματα οργάνωσης ιατρείων είναι σχετικά παροχειμένα, τόσο από άποψη περιβάλλοντος χρήστη όσο και από άποψη διασύνδεσης με το διαδίκτυο. Έτσι τελικά επέλεξα να αναπτύξω μια Web εφαρμογή που θα διευκολύνει την καθημερινή ζωή ενός ιατρού, προσφέροντάς του έναν αποτελεσματικό και γρήγορο τρόπο να οργανώσει τα ραντεβού του και να διαχειρίζεται το ιστορικό των ασθενών του.

Ο σκοπός αυτής της εργασίας είναι η εξερεύνηση και αξιολόγηση της πλατφόρμας Django⁴ μέσω της ανάπτυξης μιας Web εφαρμογής για τη διαχείριση των επαφών (ασθενών), περιστατικών και των ραντεβού ενός ιατρού. Πιο συγκεκριμένα η εφαρμογή που αναπτύχθηκε εστιάστηκε στην ειδικότητα του οδοντιάτρου, αλλά σχεδιάστηκε με γνώμονα την εύκολη επεκτασιμότητα και σε άλλες ιατρικές ειδικότητες. Έτσι, η αξιολόγηση των δυνατοτήτων των τεχνολογιών διαδικτύου που επικρατούν στις μέρες μας, προσομοιώνεται μέσα από ένα ρεαλιστικό σενάριο καθημερινής χρήσης.

Λέξεις Κλειδιά: << διαδικτυακά συστήματα, Web εφαρμογές, Django, διαχείριση ασθενών >>

Πίνακας περιεχομένων

1	Εισαγωγή στο Web και Web 2.0	4
1.1	Διαδραστικότητα (AJAX) στο Web.....	7
1.2	Το Web σαν λειτουργικό σύστημα	8
1.3	Cloud Computing.....	10
1.4	Το Web στις ιατρικές πρακτικές.....	11
2	Προδιαγραφές και Αρχιτεκτονική της υλοποίησης	13
2.1	Απαιτήσεις χρηστών	13
2.1.1	Διαχείριση ασθενών	13
2.1.2	Διαχείριση ραντεβού.....	14
2.1.3	Διαχείριση περιστατικών	14
2.1.4	Προστασία (προσωπικών) δεδομένων.....	14
2.2	Αρχιτεκτονική του συστήματος.....	14
3	Τεχνολογίες που χρησιμοποιήθηκαν	18
3.1	Python	18
3.1.1	Django.....	19
3.2	JavaScript.....	25
3.2.1	JQuery	26
3.3	AJAX	27
4	Υλοποίηση	28
4.1	Περιβάλλον χρήσης	28
4.2	Σενάριο χρήσης.....	32
5	Σύνοψη και μελλοντικές προοπτικές	38
6	Αναφορές	40

Πίνακας Εικόνων

Εικόνα 1-1: mashup λέξεων με τα χαρακτηριστικά του Web 2.0	5
Εικόνα 1-2: Χαρακτηριστικά και εφαρμογές του Web 2.0	6
Εικόνα 2-1: Η αρχιτεκτονική της υλοποίησης.	15
Εικόνα 2-2: Λειτουργίες και μεθοδολογία υλοποίησης της εφαρμογής.....	16
Εικόνα 2-3: Σχέσεις των αντικειμένων της βάσης δεδομένων.....	17
Εικόνα 3-1: Η αρχιτεκτονική μιας Django εφαρμογής.	24
Εικόνα 3-2: Η ροή πληροφορίας μια αίτησης.	25
Εικόνα 4-1: Η κεντρική ιστοσελίδα της εφαρμογής, με χρήση της μηνιαίας προβολής.	29
Εικόνα 4-2: Η κεντρική ιστοσελίδα της εφαρμογής, με χρήση της εβδομαδιαίας προβολής.	30
Εικόνα 4-3: Η γρήγορη επισκόπηση ενός ραντεβού.	31
Εικόνα 4-4: Η προβολή των λεπτομερειών ενός ραντεβού.....	31
Εικόνα 4-5: Η ιστοσελίδα εισαγωγής στοιχείων χρήστη	33
Εικόνα 4-6: Το gadget των ασθενών.....	34
Εικόνα 4-7: Η ιστοσελίδα εισαγωγής των στοιχείων ενός νέου ασθενή.....	34
Εικόνα 4-8: Η ιστοσελίδα δημιουργίας ενός ραντεβού.....	35
Εικόνα 4-9: Η δημιουργία ενός νέου περιστατικού.....	35
Εικόνα 4-10: Η τελική μορφή της προβολής Month View με το νέο ραντεβού	36
Εικόνα 4-11: Η τελική μορφή της προβολής Week View με το νέο ραντεβού.....	37

1

Εισαγωγή στο Web και Web 2.0

Η λειτουργία του Web όταν πρωτοεμφανίστηκε, αλλά και για αρκετά χρόνια αργότερα, μπορούσε να παρομοιαστεί με μια τεράστια ψηφιακή βιβλιοθήκη. Παρότι αυτό στις μέρες μας μπορεί να θεωρηθεί δεδομένο, στα πρώτα στάδια χρήσης του έφερε τρομερές αλλαγές, καθώς οι ερευνητικές ομάδες μπορούσαν πλέον να παρουσιάζουν άμεσα τα αποτελέσματα των ερευνών τους στο σύνολο της επιστημονικής κοινότητας. Με τον καιρό το Internet έγινε προσιτό στο ευρύ κοινό και κάποια στιγμή ήταν αναμενόμενο να έρθει κάποια αλλαγή.

Η αλλαγή αυτή ήταν το Web 2.0. Ο όρος Web 2.0 χρησιμοποιείται για να περιγράψει προηγμένες τεχνολογίες διαδικτύου και εφαρμογές αυτών, όπως τα blog, τα wiki, τα RSS και τα κοινωνικά δίκτυα. Καθώς ο σαφής διαχωρισμός Web 1.0 και Web 2.0 δεν είναι εύκολος, πολλές φορές προτιμάμε να εστιάσουμε στην παρουσία μερικών χαρακτηριστικών ή δυνατοτήτων όπως:

- το blogging και η δυνατότητα σχολιασμού από τους χρήστες
- η αξιοποίηση νέων τεχνολογιών όπως η AJAX
- οι Web εφαρμογές
- παροχή RSS και online RSS readers
- η δυνατότητα συγκέντρωσης πληροφορίας από άλλες πηγές (mashups)
- χρήση ανοιχτών API, προσφέροντας διαλειτουργικότητα μεταξύ site
- τα wiki και άλλες συμμετοχικές εφαρμογές

- οι εφαρμογές κοινωνικών δικτύων
- ευκολία στη δημιουργία και διαχείριση δεδομένων/περιεχομένου από απλούς χρήστες

Οι εφαρμογές του Web 2.0 και τα χαρακτηριστικά του παρουσιάζονται και γραφικά στις εικόνες Εικόνα 1-1 και Εικόνα 1-2.

Κάποιες άλλες φορές πάλι μπορούμε να πούμε ότι η έννοια του Web 2.0 εμπεριέχει τη χρήση τεχνολογιών που επιτρέπουν στα site να μοιράζονται και να διαχειρίζονται το περιεχόμενο άλλων sites, καθώς επίσης και τη χρήση “ετικετών” (“tags”) για την εύκολη αναζήτηση σχετικού περιεχομένου και αυτοματοποιημένη ειδοποίηση των χρηστών μέσω RSS. Ένα καλό παράδειγμα είναι η περίπτωση του Last FM⁶. Πρόκειται για ένα online ραδιοφωνικό σταθμό όπου κάθε χρήστης ακούει διαφορετικά τραγούδια και η επιλογή της λίστας τραγουδιών γίνεται ανάλογα με τις προτιμήσεις του χρήστη. Επιπλέον χρησιμοποιεί “ετικέτες” (“tags”) σε κάθε τραγούδι και συγκρότημα ώστε να γίνονται εύκολες συσχετίσεις και αναζητήσεις παρόμοιου περιεχομένου.

Αλλά το Web 2.0 μπορεί κάλλιστα να εμπεριέχει υπηρεσίες που μετατρέπουν μια κοινότητα σε περιεχόμενο, κάτι που παρατηρείται έντονα στα κοινωνικά δίκτυα. Έτσι, το περιεχόμενο του διαδικτυακού τόπου μπορεί να προστίθεται από την κοινότητα όπως στα wiki και τα forums, αλλά επιπλέον το μπορεί να είναι η ίδια κοινότητα, όπως στην περίπτωση του Facebook⁷.



Εικόνα 1-1: mashup λέξεων με τα χαρακτηριστικά του Web 2.0.

Εικόνα 1-2: Χαρακτηριστικά και εφαρμογές του Web 2.0¹.



Η γενιά των Web 2.0 ιστοσελίδων, δίνει έμφαση στον τρόπο που ο χρήστης αλληλεπιδρά με το site, τη βελτίωση του σχεδιασμού των ιστοσελίδων, τη δυνατότητα να επεμβαίνει στο περιεχόμενο των ιστοσελίδων, τη συμμετοχικότητα και τη δυνατότητα να δρα στο διαδίκτυο όπως δρα και στον υπολογιστή του.

Οι χρήστες με την εμφάνιση του Web 2.0, άρχισαν να μην παρακολουθούν απλές στατικές ιστοσελίδες, αλλά να επεμβαίνουν δυναμικά στο περιεχόμενό τους, έχοντας τη δυνατότητα ακόμα και να “ανεβάσουν” δικό τους περιεχόμενο. Πλέον μπορούσαν να δημοσιεύουν ειδήσεις αλλά και να γράφουν τη γνώμη τους για αυτά που διαβάζουν. Η επανάσταση αυτή στον τρόπο αλληλεπίδρασης με το Web, έφερε μια σημαντική αύξηση στον αριθμό των χρηστών του Internet οι οποίοι άρχισαν να αλληλεπιδρούν μεταξύ τους, μέσα από τα νεοδημιουργηθέντα forums. Έτσι καθώς οι χρήστες άρχισαν να αντιλαμβάνονται την ύπαρξη και άλλων χρηστών στο Internet, με παρόμοια ενδιαφέροντα με αυτούς, εμφανίστηκε η έννοια της κοινότητας (community).

1.1 Διαδραστικότητα (AJAX) στο Web

Το Web 2.0 άλλαξε τη χρήση του Internet από μέσο ενημέρωσης σε μέσο επικοινωνίας και ανταλλαγής απόψεων, αλλά η αλλαγή του τρόπου λειτουργίας του είναι αποτέλεσμα της χρήσης μιας άλλης τεχνολογίας, της AJAX. Με την υιοθέτηση τεχνολογιών όπως η AJAX και την εκτενέστερη χρήση JavaScript², η διαδραστικότητα των ιστοσελίδων έγινε πιο έντονη.

Η AJAX είναι μια προγραμματιστική τεχνική για τη δημιουργία καλύτερων, γρηγορότερων και πιο διαδραστικών Web εφαρμογών. Η έννοια της AJAX (Asynchronous JavaScript and XML) είναι η ασύγχρονη μεταφορά δεδομένων μεταξύ client και Server, η οποία δεν χρειάζεται να περιορίζεται σε XML³. Μία από τις πρώτες εφαρμογές ήταν το 2005 το Google suggest, όπου καθώς ο χρήστης συμπλήρωνε το πεδίο αναζήτησης, στην ιστοσελίδα της Google, εμφανίζονταν προτάσεις για το κείμενο αναζήτησης που εισήγαγε. Έτσι ο χρήστης βλέπει το περιεχόμενο των ιστοσελίδων που επισκέπτεται να ανταποκρίνεται στις ενέργειες του. Οι ιστοσελίδες πλέον δεν ξαναφορτώνονταν ολόκληρες με κάθε ενέργεια του χρήστη, παρά αλλάζει το περιεχόμενο ενός μόνο κομματιού, χωρίς να επηρεάζεται η λειτουργία των

² Η JavaScript είναι μια γλώσσα προγραμματισμού, που είναι κομμάτι των περισσότερων browsers και επιτρέπει τη δημιουργία ιστοσελίδων που ανταποκρίνονται στις ενέργειες του χρήστη.

³ Η XML (Extensible Markup Language) αποτελεί ένα σύνολο κανόνων για την κωδικοποίηση δεδομένων σε ψηφιακά αρχεία.

υπολοίπων. Αυτό τις κάνει να μοιάζουν με τις εφαρμογές που κάθε χρήστης έχει συνηθίσει να τρέχει στον υπολογιστή του, μετατρέποντας το Internet σε πλατφόρμα εφαρμογών.

Ένα περίτρανο παράδειγμα μιας Web εφαρμογής⁴ μπορεί να αποτελέσει η υπηρεσία google maps που προβάλλει χάρτες, εικόνες καθώς και διαδρομές από κάθε περιοχή του κόσμου, ανταποκρινόμενο στην περιοχή που ο χρήστης βλέπει εκείνη τη στιγμή. Το επίπεδο διαδραστικότητας που η υπηρεσία google maps προσφέρει έγινε γρήγορα συγκρίσιμο με τις δυνατότητες της εφαρμογής google earth έχοντας σαν αποτέλεσμα, την προτίμηση της Web εφαρμογής από τον κόσμο.

1.2 Το Web σαν λειτουργικό σύστημα

Το Web ξεκίνησε σαν μέσο ενημέρωσης, είδαμε πως σταδιακά εξελίχθηκε σε μέσο επικοινωνίας και τελικά απέκτησε και μια νέα μορφή αλληλεπίδρασης με το χρήστη. Αυτή η νέα μορφή αλληλεπίδρασης έκανε τους πρωτοπόρους του Internet να δοκιμάσουν να δημιουργήσουν Web εφαρμογές. Από τους πρώτους τομείς δοκιμών ήταν τα Webmails, που γρήγορα αντέγραψαν κάθε λειτουργία των εφαρμογών email . Πλέον ο χρήστης έμπαινε σε μία και μόνο ιστοσελίδα, η οποία άλλαζε ταχύτατα περιεχόμενο χωρίς να ξαναφορτώνεται. Έτσι εμφανίστηκαν και εδραιώθηκαν οι Web εφαρμογές.

Το Web ήταν πλέον ένα “λειτουργικό σύστημα”, προσφέροντας εφαρμογές που δεν χρειάζονταν εγκατάσταση, ήταν προσβάσιμες από κάθε υπολογιστή και δεν στερούνταν χαρακτηριστικών σε σύγκριση με τις κλασσικές offline υλοποιήσεις. Αρχικά απλές, όπως ημερολόγια και στη συνέχεια πολύπλοκες, όπως επεξεργαστές εγγράφων. Με τον καιρό εμφανίστηκαν προσπάθειες προσομοίωσης ενός ολοκληρωμένου περιβάλλοντος γραφείου μέσω μιας Web εφαρμογής. Μια τέτοιου είδους εφαρμογή παρείχε πληθώρα εργαλείων όπως επεξεργαστές εγγράφων, προγράμματα επεξεργασίας εικόνας, δημιουργίας παρουσιάσεων, ακόμα και προγράμματα chat, Web browsers και παιχνίδια. Μια τέτοια εφαρμογή προσφέρει στο χρήστη αποθηκευτικό χώρο και ένα προσωποποιημένο περιβάλλον εργασίας, προσβάσιμο από οπουδήποτε υπάρχει μια σύνδεση στο Internet, χωρίς να τον απασχολεί με θέματα όπως οι εγκαταστάσεις προγραμμάτων.

Ένα Web OS ουσιαστικά είναι ένα γραφικό περιβάλλον, που προσφέρει στο χρήστη πρόσβαση σε εφαρμογές και δεδομένα που είναι όλα ή εν μέρει αποθηκευμένα στο Web. Ένα Web OS είναι ανεξάρτητο από το λειτουργικό σύστημα του υπολογιστή του χρήστη και όσο

⁴ Μια Web εφαρμογή, είναι μία εφαρμογή που ο χρήστης χρησιμοποιεί μέσα από τον browser του, επισκεπτόμενος συνήθως κάποια ιστοσελίδα.

οι δυνατότητές του αυξάνονται, με την προσθήκη συστήματος διαχείρισης εφαρμογών και αρχείων, τόσο η χρήση του επικαλύπτεται με αυτή των κλασικών λειτουργικών συστημάτων. Ένα Web OS είναι εστιασμένο στην παροχή Web εφαρμογών, αλλά λόγω των δυνατοτήτων του μπορεί να αποτελέσει λύση θα μετατρέψει τον υπολογιστή του χρήστη σε απλό τερματικό, εξαλείφοντας πλέον την ανάγκη εγκατάστασης λογισμικού. Για παράδειγμα με αυτό τον τρόπο μια εταιρία θα μπορούσε να μειώσει το κόστος υπηρεσιών του τμήματος τεχνικής υποστήριξης, καθώς οι υπολογιστές πλέον χρειάζονται απλά ένα browser, αποκτώντας παράλληλα ένα κεντρικό σημείο αναβαθμίσεων, ενημερώσεων και διορθώσεων.

1.3 Cloud Computing

Η τάση της χρήσης Web εφαρμογών, δημιούργησε την έννοια του “cloud computing”. Σύμφωνα με αυτό το μοντέλο οι πόροι που οι χρήστες χρησιμοποιούν, αντί να είναι συγκεντρωμένοι στον υπολογιστή τους (ή γενικότερα να τους ανήκουν), είναι διασκορπισμένοι σε όλο το Internet. Ο χρήστης δε γνωρίζει που εντοπίζονται οι πόροι, οι οποίοι δεν του ανήκουν αλλά και δεν τον απασχολεί η συντήρηση και διαχείρισή τους. Επιπλέον σε μια μεγαλύτερη γενίκευση του “Cloud Computing”, οι πόροι δε χρειάζεται καν να βρίσκονται στην ίδια τοποθεσία ή να είναι οι ίδιοι κάθε φορά που χρησιμοποιείται μια υπηρεσία.

Ακολουθώντας την ίδια λογική ούτε η υποδομή που ενώνει το χρήστη (ή τους χρήστες) με την εφαρμογή (ή και μεταξύ των χρηστών) του ανήκει ή τον απασχολεί η υλοποίηση και συντήρηση της. Έχοντας, λοιπόν, τους πόρους και την υποδομή να ακολουθούν αυτή τη λογική χρήσης, είναι δυνατή και η παροχή εφαρμογών ως υπηρεσιών. Έτσι, οι εφαρμογές δεν ανήκουν στο χρήστη, ο οποίος μπορεί να έχει πρόσβαση σε αυτές από παντού.

Συγκεντρώνοντας όλα τα παραπάνω μπορούμε να αντιληφθούμε το μοντέλο αυτό, σαν το “σύννεφο” που χρησιμοποιείται στα δίκτυα για να περιγράψει μια υποδομή με κάποιους πόρους και κάποια λειτουργικότητα που δε μας αφορά ο τρόπος λειτουργίας τους ή η υλοποίησή τους. Ο υπολογιστής του χρήστη, όσον αφορά εφαρμογές “Cloud Computing”, μπορεί να παρομοιαστεί με τερματικό πρόσβασης, από όπου ο χρήστης αποκτά πρόσβαση μέσω της υποδομής του σύννεφου στους πόρους και τις εφαρμογές που υπάρχουν εντός του.

Κύρια πλεονεκτήματα του “Cloud Computing” είναι:

- η υψηλή διαθεσιμότητα
- η επεκτασιμότητα
- η αμεσότητα και ευκολία στην πρόσβαση από παντού
- κεντρικό σημείο ενημερώσεων αναβαθμίσεων και συντήρησης των Web εφαρμογών

Πλέον ακολουθώντας το παραπάνω μοντέλο, οι χρήστες δε χρειάζεται να αγοράζουν πακέτα εφαρμογών που είτε δεν αξιοποιούν όλες τις δυνατότητες που έχουν είτε δεν τις χρησιμοποιούν και τόσο συχνά. Επιπρόσθετα δε χρειάζεται ούτε να ανησυχούν για την ασφάλεια ή την επάρκεια των δυνατοτήτων των πόρων που αξιοποιούνται για τη λειτουργία μιας εφαρμογής που χρησιμοποιούν.

Με το μοντέλο του “Cloud Computing”

- η χρήση των εφαρμογών
- η χρήση των επεξεργαστικών και αποθηκευτικών πόρων
- η χρήση της υποδομής πρόσβασης

είναι υπηρεσίες που μπορούν να προσφέρονται στο χρήστη ο οποίος όμως χρεώνεται μόνο για ό,τι υπηρεσίες και όσο τις χρησιμοποιεί.

Ένα απλό παράδειγμα είναι το google mail, όπου ούτε οι servers που αξιοποιούνται, ούτε τα αποθηκευτικά μέσα, ούτε η δικτυακή υποδομή ανήκουν στο χρήστη. Όμως αυτή η διασπορά των πόρων δεν στοιχίζει ούτε σε επιδόσεις, ούτε σε λειτουργικότητα και ο χρήστης μπορεί να χρησιμοποιεί όλες τις λειτουργίες που οι παραδοσιακοί mail clients προσφέρουν.

Ακολουθώντας την προσέγγιση του “Cloud Computing” μια εταιρία μπορεί να ενοικιάσει ένα server για να στεγάσει το δικτυακό της τόπο ή το ίδιο εύκολα να “ενοικιάσει” τις εφαρμογές ή και το λειτουργικό σύστημα που θα χρησιμοποιεί, ακολουθώντας τη λογική των Web OS. Με αυτό τον τρόπο όλοι οι υπάλληλοι της εταιρίας αποκτούν ένα ενιαίο περιβάλλον εργασίας του οποίου οι δυνατότητες μπορούν να ακολουθήσουν τις απαιτήσεις για την εξέλιξη της εταιρίας.

1.4 To Web στις ιατρικές πρακτικές

Ας αναφερθούμε τώρα στις μέχρι τώρα σχέσεις της ιατρικής κοινότητας με το διαδίκτυο και αξιολογεί το λογισμικό που χρησιμοποιούν.

Μέχρι τώρα η χρήση του Web από τους ιατρούς περιορίζεται κυρίως στην αναζήτηση πληροφοριών και επιπλέον βιβλιογραφίας. Ο αριθμός των online κοινοτήτων είναι περιορισμένος, ενώ αποτελείται κυρίως από φοιτητές και συνήθως παρέχει προτάσεις για βιβλιογραφία¹. Δεν υπάρχει λοιπόν, ακόμα, κάποια υποδομή για την ενεργοποίηση της ιατρικής κοινότητας καθώς η πιο συνήθης ενέργεια στο Web σχετικά με το επάγγελμά τους, είναι η ανάγνωση ιατρικών νέων.

Λόγω της φύσης του κλάδου το μεγαλύτερο μέρος των προγραμμάτων περιορίζεται στην αρχειοθέτηση των ασθενών και του ιατρικού ιστορικού τους. Παρατηρείται, δε, μεγάλο ποσοστό απόλυτα εξειδικευμένων προγραμμάτων που απευθύνονται σε συγκεκριμένους κλάδους της ιατρικής. Προγράμματα σαν αυτά, πολλές φορές είναι σχεδιασμένα να συνεργάζονται με συγκεκριμένο ιατρικό εξοπλισμό προσφέροντας παράλληλα δυνατότητες καταγραφής ασθενών και περιστατικών. Τέτοιου είδους εξειδικευμένα προγράμματα που συνεργάζονται με ιατρικό εξοπλισμό χρησιμοποιούνται για παράδειγμα από πνευμονολόγους και καρδιολόγους και πολλές φορές παρέχονται από την εταιρία κατασκευής του εξοπλισμού.

Η απευθείας συνεργασία με ιατρικό εξοπλισμό είναι κάτι που μια Web εφαρμογή δύσκολα μπορεί να ανταγωνιστεί.

Καθώς η κύρια μάζα των ιατρών που χρησιμοποιούν τους υπολογιστές και το διαδίκτυο ανήκουν στην νέα γενιά ιατρών, είναι λογικό η αγορά εφαρμογών να μην έχει προλάβει να ανταποκριθεί. Τα υπάρχοντα προγράμματα, με εξαίρεση κάποιες πρόσφατες εφαρμογές^{2 3}, είναι αρκετά λιτά. Το περιβάλλον χρήσης συνήθως παραπέμπει σε περασμένες εποχές.

2

Προδιαγραφές και Αρχιτεκτονική της υλοποίησης

2.1 Απαιτήσεις χρηστών

Παρακάτω θα αναλύσουμε τις απαιτήσεις για μια ιατρική εφαρμογή διαχείρισης ραντεβού για οδοντιάτρους. Η στροφή μας προς αυτή τη συγκεκριμένη ειδικότητα έχει να κάνει με το γεγονός της κατά βάση σταθερής πελατείας ενός οδοντιάτρου. Έτσι λόγω της διαχρονικότητας των σχέσεων των ασθενών με τον οδοντίατρο τους, αποκτά μεγαλύτερη αξία η δυνατότητα της εύκολης παρακολούθησης των περιστατικών ενός ασθενή μέσα στην πάροδο του χρόνου.

2.1.1 Διαχείριση ασθενών

Ένα από τα κύρια χαρακτηριστικά μίας εφαρμογής για ιατρούς είναι η διαχείριση των ασθενών. Η εφαρμογή πρέπει να παρέχει εύκολη πρόσβαση στα στοιχεία κάθε ασθενή, συμπεριλαμβανομένου του ιατρικού του ιστορικού. Λόγω του πλήθους των ασθενών η εφαρμογή θα πρέπει να έχει προνοήσει για την εύκολη, εύστοχη και άμεση αναζήτηση των ασθενών.

2.1.2 Διαχείριση ραντεβού

Θέλουμε να ενσωματώσουμε μια ατζέντα στο πρόγραμμα για την εμφάνιση και τη διαχείριση των ραντεβού του ιατρού με τους ασθενείς. Πρέπει να είναι δυνατή η άμεση επισκόπηση των ραντεβού της εκάστοτε εβδομάδας και συνολικά του κάθε μήνα. Με αυτό τον τρόπο ο ιατρός θα μπορεί να επιβλέπει το φόρτο εργασίας της εκάστοτε ημέρας και θα διευκολύνεται να επιλέξει ημερομηνίες για νέα ραντεβού.

2.1.3 Διαχείριση περιστατικών

Είναι θεμιτό και όχι πολύ δύσκολο να κατηγοριοποιήσουμε τα ραντεβού σε περιστατικά. Έτσι μπορούμε να παρακολουθήσουμε την πορεία ενός περιστατικού, για το οποίο χρειάστηκαν περισσότερα του ενός ραντεβού, μέσα στο χρόνο. Συγκεντρωτικά, έτσι μπορούμε να εποπτεύσουμε όλα τα περιστατικά και τα ραντεβού που είχαμε με το συγκεκριμένο ασθενή, διατηρώντας με αυτό τον τρόπο ένα πρόσφατο ιατρικό ιστορικό.

2.1.4 Προστασία (προσωπικών) δεδομένων

Λόγω της φύσης των δεδομένων που η εφαρμογή διατηρεί, το θέμα της προστασίας τους είναι μεγάλης σημασίας. Στην παρούσα φάση υποθέτουμε ότι αρκεί να χρησιμοποιήσουμε κάποιες από τις ήδη υπάρχουσες τεχνολογίες.

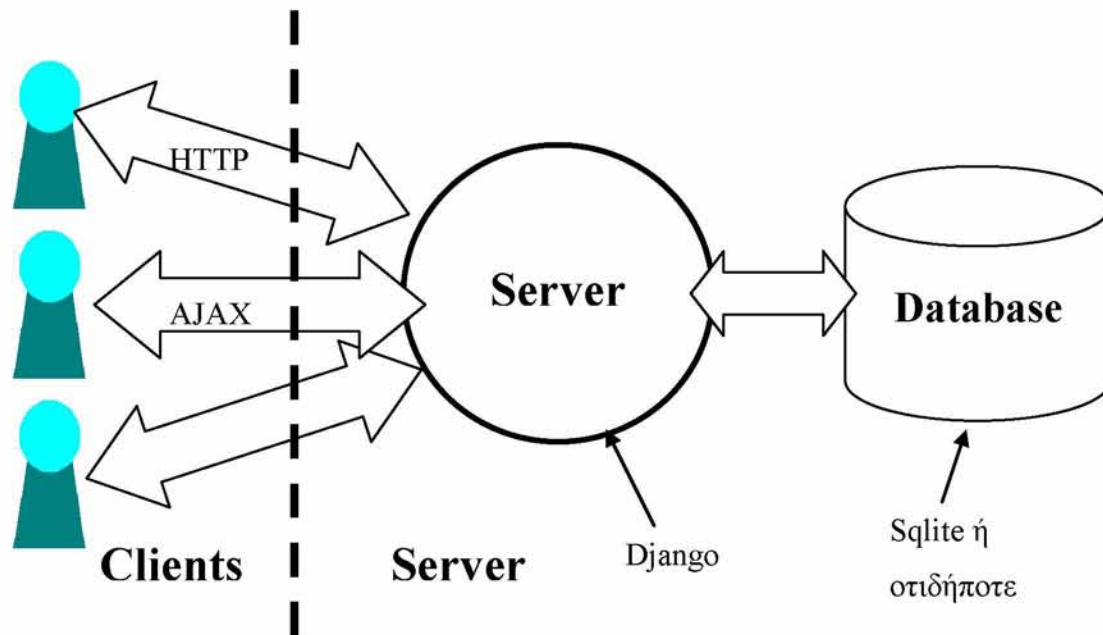
2.2 Αρχιτεκτονική του συστήματος

Περνάμε στην ανάλυση της αρχιτεκτονικής της υλοποίησής μας. Στην υλοποίηση μας χρησιμοποιούμε τεχνολογίες Web για να δημιουργήσουμε μια εφαρμογή που να καλύπτει τις προαναφερθείσες απαιτήσεις.

Σαν Web εφαρμογή, τα βασικά τμήματά της είναι δύο:

1. Ένας Web server που εξυπηρετεί τις αιτήσεις των χρηστών και
2. Μια βάση δεδομένων στην οποία ο Web server αποθηκεύει και ανακαλεί δεδομένα.

Η δική μας υλοποίηση όπως φαίνεται στην Εικόνα 2-1 χρησιμοποιεί στη μεριά του Server Django⁴ και python και η βάση δεδομένων είναι σε Sqlite. Λόγω της φύσης του Django για τη βάση δεδομένων θα μπορούσε να χρησιμοποιηθεί και σχεδόν οποιαδήποτε άλλη λύση όπως MySQL ή Oracle. Σε επόμενο κεφάλαιο θα αναφερθούμε σε αυτό το θέμα εκτενέστερα. Οι χρήστες με τη χρήση κάποιου browser θα μπορούν να έχουν πρόσβαση στις υπηρεσίες της εφαρμογής. Για τη μεταφορά των δεδομένων θα χρησιμοποιείται αρχικά το απλό πρωτόκολλο HTTP, για την αρχική φόρτωση ιστοσελίδων, ενώ στη συνέχεια το περιεχόμενο της ιστοσελίδας θα ανανεώνεται μέσα από τη χρήση ασύγχρονων μεταφορών AJAX.



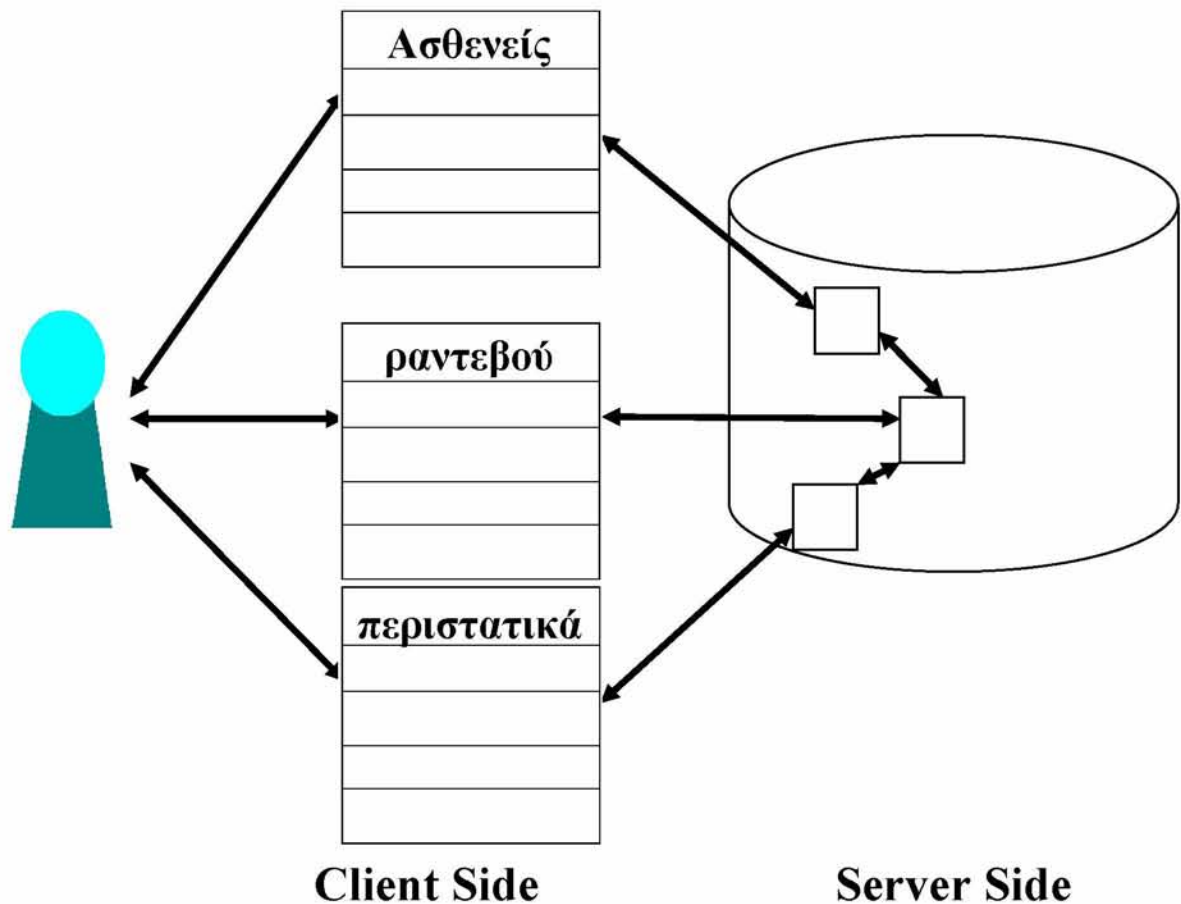
Εικόνα 2-1: Η αρχιτεκτονική της υλοποίησης.

Κάθε ιατρός έχει ένα προσωπικό όνομα χρήστη και κωδικό με τα οποία έχει πρόσβαση στη Web εφαρμογή. Με αυτό τον τρόπο τα δεδομένα του εκάστοτε χρήστη δε θα είναι προσβάσιμα από τρίτους. Η πρόσβαση στην εφαρμογή θα μπορεί να γίνει από οπουδήποτε μέσω ενός οποιουδήποτε browser, αρκεί να υπάρχει πρόσβαση στο διαδίκτυο.

Ένας χρήστης μπορεί ακόμα να έχει πρόσβαση στα ραντεβού του ακόμα και από ένα κινητό ενώ είναι στο δρόμο. Αυτό γίνεται εφικτό, καθώς τα SmartPhones⁸, όπως το Apple iPhone⁹ και το HTC Magic¹⁰, ενσωματώνουν ένα mini Web browser. Καθώς τα SmartPhones καταλαμβάνουν όλο και μεγαλύτερο μερίδιο της αγοράς, διαμόρφωσαν την αγορά ώστε το δίκτυο 3G να μπορεί πλέον να προσφέρει αξιοπρεπείς, για την πλοήγηση στο διαδίκτυο, ταχύτητες. Πέραν αυτού, οι εταιρίες τείνουν να ενσωματώνουν στα καινούργια τους μοντέλα και δυνατότητες σύνδεσης μέσω WiFi¹¹, επιτρέποντας στους χρήστες να αξιοποιήσουν και τα τυχόν υπάρχοντα σημεία πρόσβασης WiFi (WiFi Hotspots¹²).

Ο χρήστης, όπως φαίνεται στην Εικόνα 2-2, θα έχει, μέσω του browser του, τη δυνατότητα να εκτελεί άμεσα τρεις βασικές λειτουργίες:

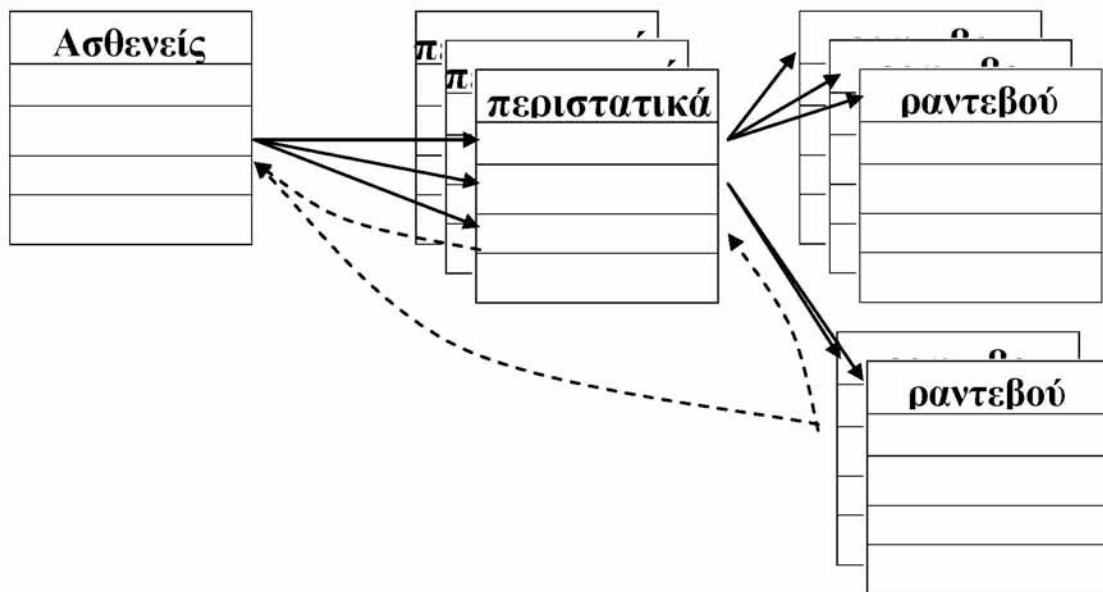
1. να δημιουργεί και να βρίσκει εγγραφές ασθενών
2. να διαχειρίζεται ραντεβού
3. και να διαχειρίζεται περιστατικά



Εικόνα 2-2: Λειτουργίες και μεθοδολογία υλοποίησης της εφαρμογής.

Η βάση δεδομένων θα αναλάβει την αποθήκευση των παραπάνω πληροφοριών σε διαφορετικούς πίνακες, συσχετισμένους μεταξύ τους. Έτσι κάθε ιατρός θα έχει μία λίστα με τους δικούς του ασθενής όπου κάθε ασθενείς έχει κάποια περιστατικά. Με τη σειρά τους τα περιστατικά αποτελούνται από ένα ή περισσότερα ραντεβού που πραγματοποιήθηκαν μέχρι την ολοκλήρωση του περιστατικού.

Επιπλέον υλοποιούνται και ανάστροφες σχέσεις, ώστε να μπορούμε να βρούμε για παράδειγμα ποιόν ασθενή αφορά ένα περιστατικό ή ένα ραντεβού. Το παραπάνω φαίνεται στην Εικόνα 2-3. Στη δική μας περίπτωση, βέβαια, οι ανάστροφες αυτές σχέσεις υλοποιούνται αυτόματα από το framework Django⁴, το οποίο θα χρησιμοποιήσουμε στην ανάπτυξη της Web εφαρμογής μας.



Εικόνα 2-3: Σχέσεις των αντικειμένων της βάσης δεδομένων.

Η πρόσβαση στα δεδομένα θα επιτρέπεται μόνο στον ιδιοκτήτη/δημιουργό των εγγραφών. Αυτό γίνεται εφικτό διατηρώντας σε κάθε πίνακα της βάσης δεδομένων μια παράμετρο που υποδηλώνει τον ιατρό/χρήστη τον οποίο δημιούργησε κάθε εγγραφή. Η προσθήκη αυτή από μόνη της δεν είναι αρκετή και κρίνεται απαραίτητος ο προσεκτικός προγραμματισμός όλων των ενεργειών του εξυπηρετητή ώστε να εκτελούνται όλοι οι απαραίτητοι έλεγχοι δικαιωμάτων.

3

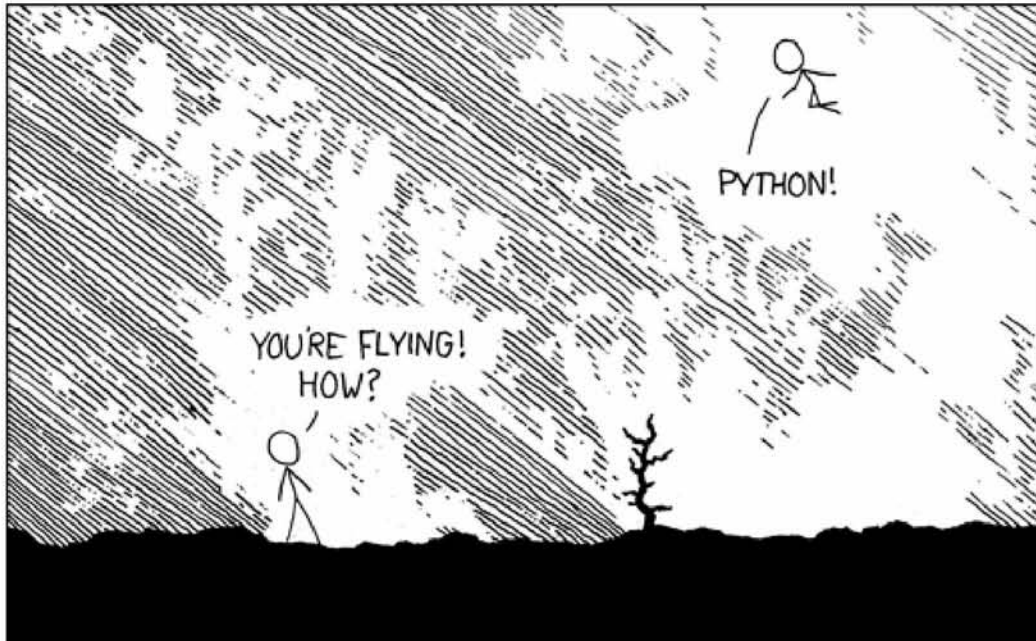
Τεχνολογίες που χρησιμοποιήθηκαν

Σε αυτό το κεφάλαιο θα ασχοληθούμε με τις τεχνολογίες που χρησιμοποιήθηκαν και επέτρεψαν την ολοκλήρωση αυτής της Web εφαρμογής.

3.1 Python

Η Python είναι μια scripting γλώσσα προγραμματισμού υψηλού επιπέδου, βασισμένη στη C, με αρκετά χρόνια εξέλιξης. Η Python παρέχει ένα περιβάλλον κονσόλας (shell), το οποίο δίνει τη δυνατότητα στον προγραμματιστή να εκτελεί και να δοκιμάζει κάθε εντολή πριν την ενσωματώσει σε κάποιο πρόγραμμα. Αυτή η δυνατότητα κάνει την Python εύκολη στην εκμάθηση αλλά επιπλέον διευκολύνει την επίλυση προβλημάτων μέσω δοκιμών (trial and error). Οι βιβλιοθήκες που παρέχονται έχουν μεγάλο εύρος εφαρμογών. Ο προγραμματιστής μπορεί εύκολα να επεξεργαστεί εικόνα και ήχο και να ταξινομήσει μια λίστα με μία γραμμή κώδικα.

Όλα τα παραπάνω έκαναν την Python μια δυνατή γλώσσα προγραμματισμού, εύκολης και άμεσης υλοποίησης, με πολλούς οπαδούς.



[Πηγή:xkcd.com](http://xkcd.com)

3.1.1 Django

Το Django⁴ είναι ένα framework για τη γρήγορη ανάπτυξη υψηλής ποιότητας Web εφαρμογών. Πρωτοεμφανίστηκε τον Ιούλιο του 2005 και η τρέχουσα έκδοση του είναι η 1.1. Αρχικά αναπτύχθηκε για να εξυπηρετήσει τις ανάγκες των διαδικτυακών μέσων ενημέρωσης αλλά τελικά εξελίχθηκε σε κάτι πιο γενικό και με πάρα πολλές δυνατότητες. Η Web εφαρμογή μας υλοποιήθηκε στην έκδοση 1,0 του Django καθώς αυτή ήταν η πιο νέα διαθέσιμη έκδοση στην αρχή, αλλά και κατά τη μεγαλύτερη διάρκεια της υλοποίησης.

Σαν framework ορίζουμε μια βασική εννοιολογική δομή που χρησιμοποιείται για να λύσει ή να αντιμετωπίσει σύνθετα ζητήματα. Με άλλα λόγια είναι ένα πλαίσιο εργασίας που μέσω ενός πλάνου και διάφορων προτεινόμενων ενεργειών μας βοηθά να αντιμετωπίσουμε μια γκάμα πολύπλοκων προβλημάτων.

Το Django σχεδιάστηκε με δύο κεντρικούς γνώμονες

- τα απαιτητικά χρονοδιαγράμματα υλοποίησης (deadlines)
- και τη δημιουργία κομψών & υψηλής απόδοσης online εφαρμογών

ενώ παράλληλα εστιάζει στην αρχή DRY (Do not Repeat Yourself ή Μην επαναλαμβάνεις τον εαυτό σου).

Το Django παρέχει βιβλιοθήκες για λειτουργίες, που είναι κοινότητες στην πλειονότητα, των Web εφαρμογών. Με αυτό τον τρόπο επιταχύνεται σημαντικά η ολοκλήρωση ενός project αφού ξεφεύγουμε από τα μονότονα κομμάτια και υλοποιούμε την κυρίως εφαρμογή.

Παράλληλα όμως η αξιοποίηση των βιβλιοθηκών προσφέρει σταθερότητα και ασφάλεια καθώς πάρα πολλά κρίσιμα κομμάτια των Web εφαρμογών είναι έτοιμα. Έτσι ο προγραμματιστής έχει χρόνο να ασχοληθεί με πραγματικά προβλήματα της εφαρμογής αλλά και να κάνει βελτιστοποιήσεις, αντί να απασχολείται με λειτουργίες που είναι κοινότητες σε όλες τις Web εφαρμογές.

Τα κύρια χαρακτηριστικά που ένας προγραμματιστής θα ήθελε να ξέρει για το Django είναι τα παρακάτω:

- Object-relational mapper
- Automatic admin interface
- Elegant URL design
- Template system
- Cache system
- Internationalization

Αλλά ας αφήσουμε την ορολογία και ας δούμε τι είναι και τι μας προσφέρει το κάθε χαρακτηριστικό.

Object-relational mapper.

Ίσως η μεγαλύτερη καινοτομία του Django είναι η ιδέα της δημιουργίας ενός wrapper που να αποκρύπτει τη βάση δεδομένων από τον προγραμματιστή. Υποστήριξη παρέχεται προς ώρας για τις εξής βάσεις δεδομένων: postgresql_psycopg2, postgresql, mysql, sqlite3 και oracle. Αυτός ο mapper επιπλέον αλλάζει τον τρόπο επικοινωνίας με τη βάση δεδομένων, από επερωτήσεις ‘SELECT ... FROM ... WHERE’, σε απλή διαχείριση αντικειμένων.

Ο προγραμματιστής αντί να ορίζει τη μορφή των πινάκων της βάσης δεδομένων, ορίζει την μορφή αντίστοιχων αντικειμένων Python τα οποία θα αποτελέσουν τα μοντέλα για την αυτόματη δημιουργία των πινάκων από το Django. Τα μοντέλα αυτά εξορισμού ορίζονται στο αρχείο models.py ενός project. Εκτός από τον τρόπο δημιουργίας αλλάζει και ο τρόπος των επερωτήσεων. Ένας προγραμματιστής πλέον μπορεί να γράψει εντολές python της μορφής Cases.objects.filter(id=’2’), οι οποίες του επιστρέφουν αντικείμενα python τα οποία μπορεί περαιτέρω να φιλτράρει ή να διαχειριστεί. Παράλληλα όμως παρέχεται και υποστήριξη για κανονικές επερωτήσεις στη βάση δεδομένων, που όμως και πάλι επιστρέφουν αντικείμενα python.

Έτσι ο προγραμματιστής αποφεύγει τις περίπλοκες και αλυσιδωτές, πολλές φορές, επερωτήσεις και χρησιμοποιεί αντικείμενα και λίστες αντικειμένων/αποτελεσμάτων. Η χρήση αντικειμένων προγραμματιστικά είναι πιο οικεία από τις επερωτήσεις. Αυτό συμβαίνει λόγω της χρήσης αντικειμένων από όλες τις σύγχρονες γλώσσες προγραμματισμού και λόγω του ότι η νέα αυτή μορφή αλληλεπίδρασης με τη βάση δεδομένων είναι υψηλότερου επιπέδου και πολύ πιο κοντά στη φυσική γλώσσα από τις επερωτήσεις. Παράλληλα ο κώδικας γίνεται πιο ευανάγνωστος, πιο σύντομος και πιο κατανοητός, πράγμα που εκτός από το να μειώνει το χρόνο ανάπτυξης μειώνει και τα λάθη.

Λόγω της ύπαρξης του mapper ο προγραμματιστής δε χρειάζεται να ξέρει τον τρόπο σύνταξης εντολών και επερωτήσεων σε πολλά είδη βάσεων δεδομένων. Αρκεί να ξέρει να κάνει αναζητήσεις μέσω του Django. Παύει να μας απασχολεί η βάση δεδομένων που ο πελάτης είναι διατεθειμένος να χρησιμοποιήσει αφού η σχεδίαση είναι ανεξάρτητη του είδους της βάσης δεδομένων. Έτσι είναι δυνατό, για την ανάπτυξη της εφαρμογής να χρησιμοποιηθεί πχ sqlite, ενώ στην τελική υλοποίηση να χρησιμοποιείται mysql.

Τα χαρακτηριστικά όμως που παρέχονται από τον mapper δε σταματούν εδώ. Συνοπτικά, θα αναφέρουμε ότι το Django αναλαμβάνει να δημιουργήσει αυτόματα πίνακες συσχετίσεων για τις σχέσεις “πολλών με πολλά” (many-to-many) και μας παρέχει έναν εύκολο τρόπο ανάστροφης αναζήτησης ξένων κλειδιών. Παράλληλα παρέχει μια προγραμματιστική διεπαφή (API), ώστε να είναι δυνατή η χρήση του mapper και “έξω” από το Django.

Automatic admin interface

Καθώς σχεδόν κάθε site χρειάζεται ένα περιβάλλον διαχειριστή, οι σχεδιαστές του Django σκεφτήκαν να υπεραπλουστεύσουν το σχεδιασμό του. Ένας διαχειριστής χρειάζεται, κατά βάση, άμεση πρόσβαση στους πίνακες της βάσης δεδομένων. Έτσι με το Django μπορούμε μέσα από τη δήλωση των μοντέλων να επιλέξουμε ποιοι πίνακες και με ποιον τρόπο θα εμφανίζονται στο περιβάλλον του διαχειριστή. Δε χρειάζεται να σχεδιάσουμε το περιβάλλον του, το οποίο έχει δική του ιστοσελίδα login, καθώς χρησιμοποιεί ένα εξορισμού template για το περιβάλλον χρήσης. Σε περίπτωση όμως που θέλουμε να επέμβουμε, μπορούμε να το κάνουμε εύκολα μέσω ενός συστήματος παράκαμψης (override) αρχείων HTML.

Το παραπάνω μοντέλο σχεδίασης μοιάζει με τη λογική των “συστημάτων διαχείρισης περιεχομένου” (CMS) αλλά γίνεται μέσα από σύνταξη σε αρχεία κώδικα python. Η αυτοματοποιημένη αυτή τεχνική πάντως, είναι αρκετά αποδοτική, σε βαθμό που γίνονται προσπάθειες μέσα από τη χρήση προγραμματιστικής διεπαφής (API) του mapper του Django, να χρησιμοποιηθεί το περιβάλλον διαχειριστή και σε άλλα project που είναι γραμμένα σε άλλες γλώσσες όπως php και Java.

Elegant URL design

Αυτό το χαρακτηριστικό μας βάζει στη λογική της χρήσης, ενός συστήματος επιλογής της ενέργειας που ο Server θα εκτελέσει, με βάση τη μορφή του url (url dispatcher) ή ακόμα και αλυσιδωτών dispatchers. Η ρύθμιση του dispatcher γίνεται μέσω του αρχείου urls.py και κάθε εντολή έχει τη μορφή “url, συνάρτηση που θα κληθεί”. Παράλληλα ο dispatcher έχει δυνατότητες αναγνώρισης κανονικών εκφράσεων και παραμέτρων μέσα στο url, ενώ οι εκφράσεις έχουν φθίνουσα προτεραιότητα ταιριάσματος.

Η χρήση ενός url dispatcher προσφέρει αδιαφάνεια προς το χρήστη για το τι εκτελείται στο Server, πώς λέγεται και πού βρίσκεται. Επιπλέον, αιτήσεις με διαφορετικά urls μπορεί να εξυπηρετούνται από συναρτήσεις που βρίσκονται σε διαφορετικά αρχεία κώδικα python (το κεντρικό αρχείο συναρτήσεων εξυπηρέτησης ονομάζεται views.py).

Έτσι, το Django μας βάζει να σκεφτούμε λίγο παραπάνω εξαρχής την μορφή των urls που θα χρησιμοποιήσουμε μειώνοντας όμως τον κίνδυνο μίας μελλοντικής ολικής ανασχεδίασης. Επιπλέον οι αλυσιδωτοί dispatcher μπορούν να διευκολύνουν ακόμα και το διαμοιρασμό της σχεδίασης τμημάτων μιας Web εφαρμογής από πολλά μέλη μιας ομάδας ανάπτυξης.

Template system

Παλαιότερα, με τη χρήση γλωσσών όπως η perl και η php, η σύνταξη του κώδικα εξυπηρέτησης μίας αίτησης γινόταν μαζί με τον κώδικα HTML που θα εμφανιζόταν σαν αποτέλεσμα. Έτσι είχαμε ουσιαστικά ένα αλφαριθμητικό, στο οποίο σε κάποια σημεία προσθέταμε διαφορετικά κομμάτια κειμένου, ανάλογα με τις παραμέτρους των αιτήσεων.

Το Django ήρθε να αλλάξει αυτόν το δύσκολο στην ανάγνωση, κατανόηση και διόρθωση του, τρόπο σχεδιασμού. Αυτή η αλλαγή έγινε με τη δημιουργία ενός εξελιγμένου συστήματος προτύπων (template system). Το Django όρισε μια φιλική προς τους σχεδιαστές (designer-friendly) γλώσσα για το σχεδιασμό προτύπων (templates) ώστε να διαχωρίσει τη σχεδίαση του site από το περιεχόμενο και των κώδικα σε python. Ένα template είναι ουσιαστικά ένα αρχείο HTML (ή XML, JSON ή και οτιδήποτε άλλο) στο οποίο υπάρχουν κάποιες “ετικέτες” (“tags”). Όταν κληθεί μια συνάρτηση python για την εξυπηρέτηση μιας αίτησης, τελικά προκαλεί την “κλήση” ενός προτύπου (template) με μια λίστα από παραμέτρους. Αυτές οι παράμετροι τοποθετούνται στο πρότυπο (template), στα σημεία όπου υπάρχουν οι “ετικέτες” (“tags”) με το αντίστοιχο όνομα.

Επιπλέον, μέσα στο πρότυπο (template) είναι δυνατή η περιορισμένη χρήση εντολών διακλάδωσης, επαναλήψεων, και διαχείρισης του περιεχομένου των αντικειμένων που ελήφθησαν σαν παράμετροι. Υπάρχει, δε, δυνατότητα κληρονομικότητας αλλά και απλής προσάρτησης άλλων προτύπων (templates). Με αυτό τον τρόπο ο προγραμματιστής σχεδιάζει για παράδειγμα μία μόνο φορά την επικεφαλίδα HTML (header) ή το μενού πλοήγησης που θα χρησιμοποιείται σε όλο το site και απλά κληρονομεί ή εισάγει αυτά τα templates σε άλλα. Έχουμε, λοιπόν, μια ιεράρχηση που σαν αποτέλεσμα έχει την εύκολη και άμεση αλλαγή ενός στοιχείου που βρίσκεται σε μόνο ένα αρχείο, αλλά κληρονομείται από πολλά άλλα πρότυπα (templates). Έτσι, για παράδειγμα, αντί να αλλάζουμε το μενού πλοήγησης από όλες τις ιστοσελίδες μας, (πράγμα που θα ήταν απαραίτητο αν χρησιμοποιούσαμε πχ Java ή php), θα κάνουμε την αλλαγή μόνο στο αρχείο που περιέχει το μενού και το οποίο εισάγεται σε όλες τις ιστοσελίδες.

Cache system

Οι πόροι ενός εξυπηρετητή χρησιμοποιούνται συνεχώς για να εξυπηρετήσουν αιτήσεις που πολλές φορές επαναλαμβάνονται. Αυτό προκαλεί μείωση της απόκρισης του εξυπηρετητή σε νέες αιτήσεις. Το Django, αναγνωρίζοντας αυτό το πρόβλημα ενσωματώνει ένα έξυπνο σύστημα αποθήκευσης πρόσφατα χρησιμοποιημένων δεδομένων (caching). Έτσι, πέραν του caching, που πιθανώς εκτελεί το λειτουργικό σύστημα και ο δίσκος, το ίδιο το Django προσφέρει μια ακόμα βελτιστοποίηση.

Ο προγραμματιστής μπορεί επιπλέον να καταδείξει κάποια δεδομένα που θεωρεί ότι το caching τους είναι σημαντικό, αλλά όπως όλα στο Django, το caching system μπορεί να παραμετροποιηθεί ή ακόμα και να αντικατασταθεί και από άλλο.

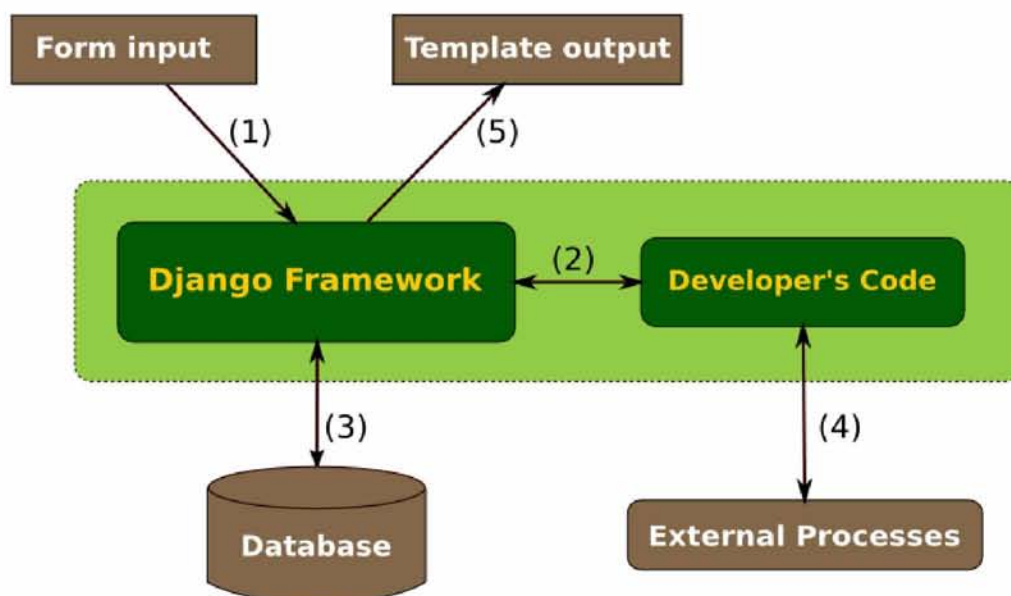
Internationalization

Στα templates του Django μπορεί να αξιοποιηθεί το πρότυπο αυτόματης μετάφρασης i18n. Όταν μια Django εφαρμογή λαμβάνει μια αίτηση μπορεί να την αναλύσει και να δει τη γλώσσα του browser που έκανε την αίτηση. Έπειτα, αν χρησιμοποιείται το πρότυπο i18n, μπορεί να αξιοποιήσει αυτήν την πληροφορία για να μεταφράσει αυτόματα τη ιστοσελίδα με την οποία θα απαντήσει. Ο προγραμματιστής μπορεί με χρήση συγκεκριμένων “ετικετών” (“tags”) μέσα στα πρότυπα (templates) να μεταφράσει επιλεκτικά συγκεκριμένες περιοχές κειμένου της ιστοσελίδας.

Αξίζει να αναφέρουμε ότι το Django δουλεύει, στο μεγαλύτερο μέρος του, μέσα από ένα αρχείο `manage.py` με εντολές όπως:

- `python manage.py syncdb`
 - για τη δημιουργία των πινάκων της βάσης δεδομένων από τα δηλωμένα μοντέλα
- `python manage.py runserver`
 - για να ξεκινήσει τον ενσωματωμένο mini Web Server

ενώ χρησιμοποιεί το αρχείο `settings.py` σαν ένα κεντρικό σημείο καθολικών δηλώσεων και ρυθμίσεων. Στην Εικόνα 3-1 βλέπουμε πιο εποπτικά τα διακριτά μέρη μίας Django εφαρμογής.



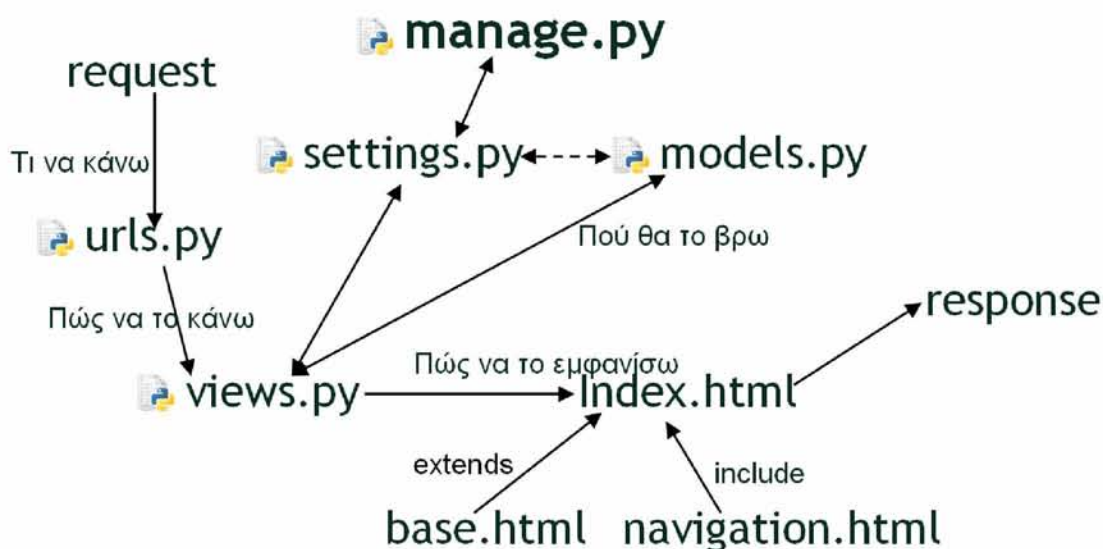
Εικόνα 3-1: Η αρχιτεκτονική μιας Django εφαρμογής.

Έτσι, βλέπουμε το Django σαν ένα ολοκληρωμένο framework με πολλές καινοτομίες και διευκολύνσεις για τον προγραμματιστή. Επιπλέον, παρατηρούμε ότι χαρακτηρίζεται από σωστή οργάνωση.

Έτσι, το Django διαχωρίζει:

- ✓ Τι να κάνω urls
- ✓ Πώς να το κάνω views
- ✓ Πού θα το βρω models
- ✓ Πώς να το εμφανίσω templates

ώστε να διευκολύνει την ανάπτυξη, την κατανόηση και την επιδιόρθωση λαθών των Web εφαρμογών. Πιο παραστατικά μπορούμε να δούμε στην Εικόνα 3-2.



Εικόνα 3-2: Η ροή πληροφορίας μια αίτησης.

Ο λόγος που επιλέξαμε τη χρήση Django έναντι άλλων τάσεων των ημερών, όπως κάποιο CMS, είναι ότι τα CMS δεν είναι ισχυρά εργαλεία ανάπτυξης εξειδικευμένων Web εφαρμογών. Από την άλλη με το Django, όπως είδαμε, ο προγραμματιστής έχει τον απόλυτο έλεγχο κάθε λεπτομέρειας και κάθε λειτουργίας απολαμβάνοντας παράλληλα μια σειρά από διευκολύνσεις και αυτοματοποιημένες λειτουργίες, όπως έτοιμο μηχανισμό συντήρησης της κατάστασης της πλοήγησης των χρηστών (session maintenance) με δικαιώματα.

3.2 JavaScript

Η JavaScript είναι άλλη μια scripting γλώσσα αντικειμενοστραφούς προγραμματισμού που εμφανίστηκε το 1995 και ενσωματώθηκε στους browser εκείνης της εποχής ώστε να κάνει πιο ζωντανές τις στατικές ιστοσελίδες εκείνης της εποχής. Έτσι ο προγραμματιστής

μπορούσε να κάνει την ιστοσελίδα να ανταποκρίνεται στις κινήσεις του χρήστη. Ένα σύνηθες παράδειγμα είναι μια περιοχή της ιστοσελίδας που αλλάζει χρώμα, είτε ανά τακτά χρονικά διαστήματα, είτε αποκρινόμενη στην μετακίνηση του ποντικιού.

Η JavaScript, όντας η μόνη γλώσσα προγραμματισμού που οποιοσδήποτε browser καταλάβαινε, χωρίς να χρειάζεται κάποιο plug-in, έγινε η βάση για τις εξελίξεις που ακολούθησαν.

3.2.1 JQuery

Καθώς ο καιρός περνούσε, οι προγραμματιστές άρχισαν να χειρίζονται τη JavaScript με πιο σύνθετο τρόπο, καταφέροντας να προσφέρουν πολύπλοκα εφέ στις ιστοσελίδες. Έτσι, εμφανίστηκαν μερικά frameworks που διευκόλυναν τον προγραμματισμό σε JavaScript προσφέροντας πολλές βιβλιοθήκες και απλοποιώντας τον τρόπο που οι πιο συνήθειες λειτουργίες προγραμματίζονται. Κάποια από τα υπάρχοντα frameworks είναι το prototype¹³, τα MooTools¹⁴ και το Dojo Toolkit¹⁵. Το JQuery⁵ έχει μια μεγάλη ενεργή κοινότητα χρηστών οι οποίοι αναπτύσσουν αξιόλογα plug-ins, με ολοκληρωμένες και σύνθετες πολλές φορές λειτουργίες, μέσω του συστήματος plugins του JQuery¹⁶, προσθέτοντας έτσι χαρακτηριστικά πρόσθετης αξίας στο JQuery⁵. Λόγω αυτής της μεγάλης κοινότητας χρηστών, το JQuery⁵ είναι το framework που έχει τη μεγαλύτερη αποδοχή από τους προγραμματιστές. Έτσι και εμείς λόγω των δυνατοτήτων του και της κοινότητας το προτιμήσαμε έναντι άλλων.

Παρακάτω αναφέρουμε τα plugins τα οποία χρησιμοποιήθηκαν από την εφαρμογή μας:

- [jQuery UI¹⁷](#)
 - Αποτελεί μια συλλογή από πολλές μικροεφαρμογές (gadgets) και επιπλέον διευκολύνει πολύ την υλοποίηση πολύπλοκων εφέ.
- [FullCalendar¹⁸](#)
 - Προσφέρει τη λειτουργία ενός μηνιαίου ημερολογίου.
 - Επιπλέον βοηθήσαμε στην ανάπτυξη και ενσωμάτωση τριών νέων λειτουργιών.
- [jQuery week-calendar¹⁹](#)
 - Προσθέτει τη λειτουργία μιας εβδομαδιαίας ατζέντας.
- [jQuery Autocomplete²⁰](#)
 - Υλοποιεί ένα σχήμα προτάσεων για την αυτόματη συμπλήρωση πεδίων.
- [Ajax upload²¹](#)

- Παρέχει τη δυνατότητα για ασύγχρονη αποστολή αρχείων από τον browser.
- [TableSorter²²](#)
 - Παρέχει ένα εύκολο τρόπο ταξινόμησης

3.3 *AJAX*

Είναι συντομογραφία του Asynchronous JavaScript and XML. Στην πραγματικότητα, είναι μια προγραμματιστική τεχνική κατά την οποία ο browser του χρήστη που επισκέπτεται μια ιστοσελίδα, συνεχίζει να επικοινωνεί με το Server, ακόμα και αφού ολοκληρωθεί η φόρτωση της ιστοσελίδας. Έτσι μπορεί να στείλει και να λάβει νέα δεδομένα, τα οποία στη συνέχεια μπορεί να προβάλει στο χρήστη, χωρίς ο χρήστης να αλλάζει ιστοσελίδα, αλλά απλά αλλάζοντας το περιεχόμενο της ήδη φορτωμένης ιστοσελίδας.

Όπως φανερώνει και το όνομα αυτής της τεχνολογίας/πρακτικής, αρχικά η μορφή των δεδομένων που μεταφέρονταν ήταν σε XML. Πλέον, είναι αρκετά σύνηθες να χρησιμοποιούνται αντικείμενα JSON (JavaScript Object Notation) αντί για XML. Έτσι, αποφεύγουμε τη διαδικασία της ανάγνωσης και ερμηνεύσης (parsing) των δεδομένων που η χρήση XML απαιτεί, λαμβάνοντας τα δεδομένα σαν αντικείμενο JavaScript. Όπως και τα XML, έτσι και τα αντικείμενα JSON μπορεί να περιέχουν μια πολύπλοκη δομή δεδομένων, παρέχοντας όμως στον προγραμματιστή ένα πιο άμεσο και συνήθη τρόπο πρόσβασης σε αυτά.

Για τους σκοπούς της παρούσας διπλωματικής, χρησιμοποιούνται μόνο μεταφορές αντικειμένων JSON αντί για XML.

4

Υλοποίηση

Σε αυτό το κεφάλαιο θα δούμε το αποτέλεσμα αυτή της εργασίας μαζί με ένα σενάριο χρήσης.

4.1 Περιβάλλον χρήσης

Μέσα από τη χρήση των παραπάνω τεχνολογιών τελικά καταλήξαμε να έχουμε αναπτύξει μια ολοκληρωμένη Web εφαρμογή που κάλυπτε όλες τις προδιαγραφές. Εδώ να σημειώσουμε για να καλύψουμε τις προδιαγραφές ασφάλειας υποθέτουμε ότι μπορούμε να χρησιμοποιήσουμε το πρωτόκολλο https.

Έτσι η κεντρική ιστοσελίδα της Web εφαρμογή μας, που πήρε το όνομα Appointments Plus, απέκτησε τη μορφή της Εικόνα 4-1.

Η κεντρική ιστοσελίδα παρέχει άμεση πρόσβαση σε όλες τις σημαντικές πληροφορίες που ένας ιατρός χρειάζεται. Το μεγαλύτερο κομμάτι καταλαμβάνει το μηνιαίο ημερολόγιο, στο οποίο φαίνονται όλα τα ραντεβού του μήνα. Τα ανολοκλήρωτα ραντεβού έχουν μπλε χρώμα, ενώ τα ολοκληρωμένα πράσινο. Η τρέχουσα ημέρα τονίζεται με γέμισμα κίτρινου χρώματος στο αντίστοιχο τετράγωνο. Στην περίπτωση της εικόνας η τρέχουσα μέρα είναι Τετάρτη 30 Σεπτεμβρίου. Επιπλέον τα ραντεβού στοιβάζονται σε κάθε μέρα με βάση την ώρα έναρξής τους.

Εικόνα 4-1: Η κεντρική ιστοσελίδα της εφαρμογής, με χρήση της μηνιαίας προβολής.

Δεξιότερα, μπορούμε να δούμε δύο gadgets που διευκολύνουν, το πρώτο στην αναζήτηση των πελατών και το δεύτερο στην αναζήτηση περιστατικών. Αν κάποιος κάνει κλικ σε κάποια εγγραφή, τότε το περιεχόμενο της ιστοσελίδας θα αλλάξει προβάλλοντας πλέον τα στοιχεία του επιλεγμένου ασθενή ή τις λεπτομέρειες του περιστατικού. Και στα δύο gadgets τα αποτελέσματα προβάλλονται ταξινομημένα δίνοντας παράλληλα στο χρήστη να αλλάξει τη μορφή της.

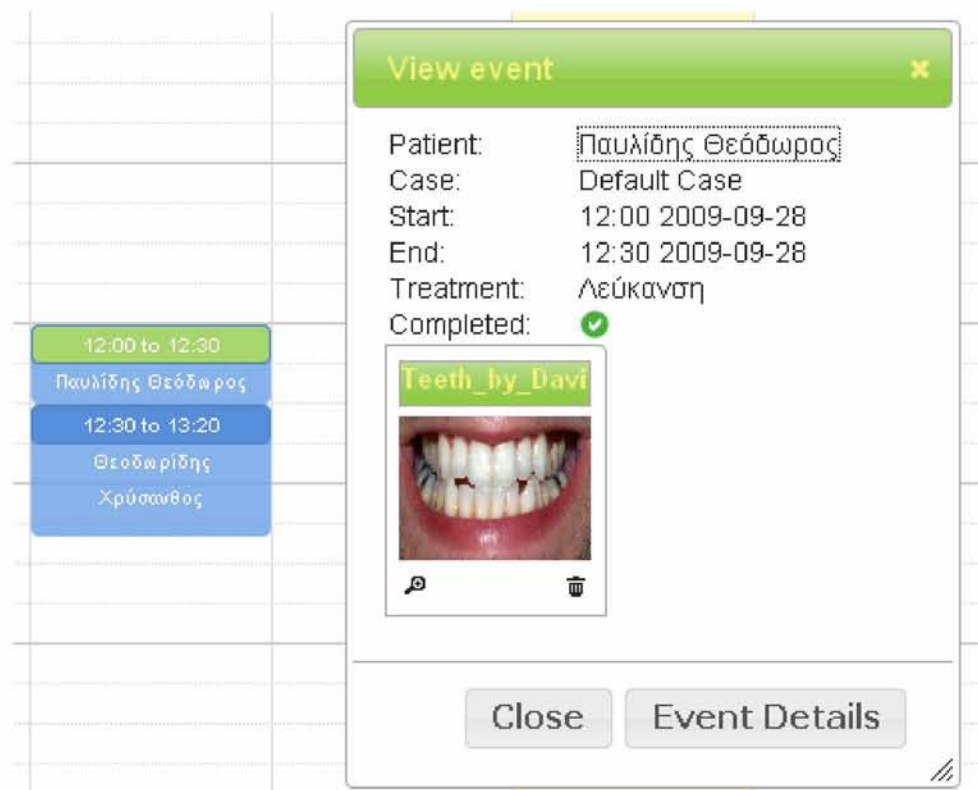
Μπορεί εύκολα κανείς χρησιμοποιώντας το πλήκτρο “Week View” (το οποίο τονίζουμε με κόκκινο περίγραμμα) να αλλάξει τον τρόπο προβολής των ραντεβού, από τη μορφή μηνιαίου ημερολογίου, σε μορφή εβδομαδιαίας ατζέντας. Σε αυτή τη μορφή, όπως φαίνεται και στην Εικόνα 4-2, τα ραντεβού έχουν ύψος τέτοιο ώστε να ταιριάζει με την ώρα έναρξης και λήξης του ραντεβού.

Και στις δύο μορφές προβολής, ο χρήστης μπορεί να μετακινήσει τα ραντεβού με απλό drag&drop. Στην περίπτωση της εβδομαδιαίας προβολής, ο χρήστης μπορεί να αυξομειώσει και το ύψος από τα πλαίσια των ραντεβού, αλλάζοντας έτσι με εύκολο και άμεσο τρόπο την ώρα έναρξης και λήξης του ραντεβού.

Εικόνα 4-2: Η κεντρική ιστοσελίδα της εφαρμογής, με χρήση της εβδομαδιαίας προβολής.

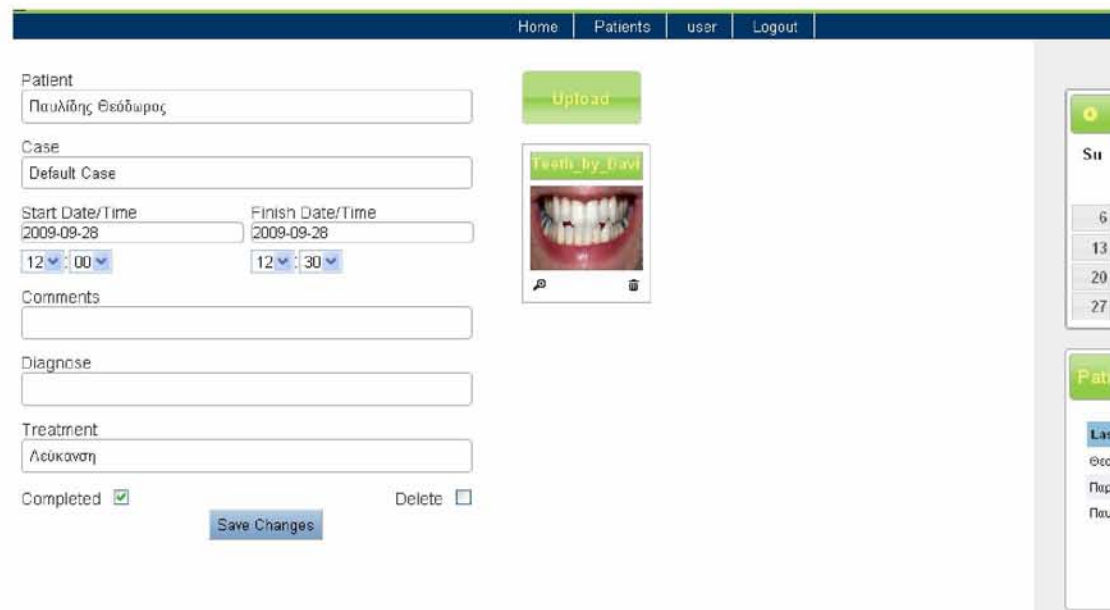
Η εφαρμογή προσφέρει μια γρήγορη σύνοψη του ραντεβού αν κάποιος κάνει κλικ σε κάποιο από αυτά με τη μορφή ενός μενού διαλόγου όπως αυτό που φαίνεται στη Εικόνα 4-3.

Στη σύνοψη αυτή, αναγράφονται μόνο τα μη κενά πεδία, ενώ τα πεδία του ασθενή και του περιστατικού λειτουργούν σαν “ετικέτες” (“tags”), δηλαδή μπορούμε να κάνουμε κλικ και να μας εμφανίσουν περισσότερες πληροφορίες είτε για τον ασθενή είτε για το περιστατικό. Στο τέλος της σύνοψης εμφανίζονται μικρογραφίες από το ψηφιακό περιεχόμενο που τυχόν έχουμε προσθέσει στο συγκεκριμένο ραντεβού, όπως πχ ψηφιακές φωτογραφίες και ακτινογραφίες..



Εικόνα 4-3: Η γρήγορη επισκόπηση ενός ραντεβού.

Αν κάποιος επιλέξει “Event Details” τότε το περιεχόμενο της ιστοσελίδας αλλάζει στη μορφή της Εικόνα 4-4.



Εικόνα 4-4: Η προβολή των λεπτομερειών ενός ραντεβού.

Εδώ ο χρήστης μπορεί να αλλάξει μια πληθώρα πεδίων, όπως είναι η διάγνωση και η φαρμακευτική αγωγή. Δεξιά, δίνεται η δυνατότητα στο χρήστη να κάνει upload ψηφιακό περιεχόμενο που μπορεί να σχετίζεται με το ραντεβού καθώς και να δει μικρογραφίες από το περιεχόμενο που πιθανόν έχει ήδη κάνει upload. Οι νέες φωτογραφίες έπειτα εμφανίζονται με τη μορφή μικρογραφιών λίγο χαμηλότερα, μαζί με τις πιθανόν προϋπάρχουσες, ενώ προφέρεται και η δυνατότητα μεγέθυνσης.

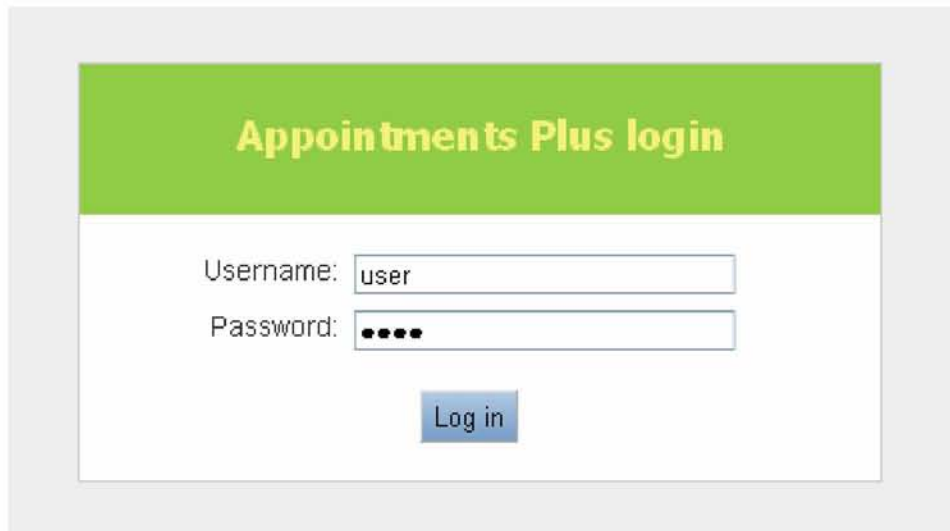
Πατώντας κανείς το πλήκτρο “Save Changes” αποθηκεύονται οι αλλαγές και επιστρέφει στη μηνιαία προβολή. Αν κανείς δε θέλει να αποθηκεύσει τις αλλαγές μπορεί να χρησιμοποιήσει ένα από τα πλήκτρα “Month View” ή “Week View”.

Συνοψίζοντας, σε αυτό το κεφάλαιο είδαμε πώς ο χρήστης χρησιμοποιεί την πλειονότητα των λειτουργιών αυτής της εφαρμογής. Σημειώνουμε δε, ότι για την ολοκλήρωση των παραπάνω ενεργειών δεν χρειάστηκε να αλλάξει ή να ξαναφορτωθεί η κεντρική ιστοσελίδα. Αυτό είναι εφικτό με την έντονη χρήση των τεχνολογιών JavaScript, JQuery και AJAX.

4.2 Σενάριο χρήσης

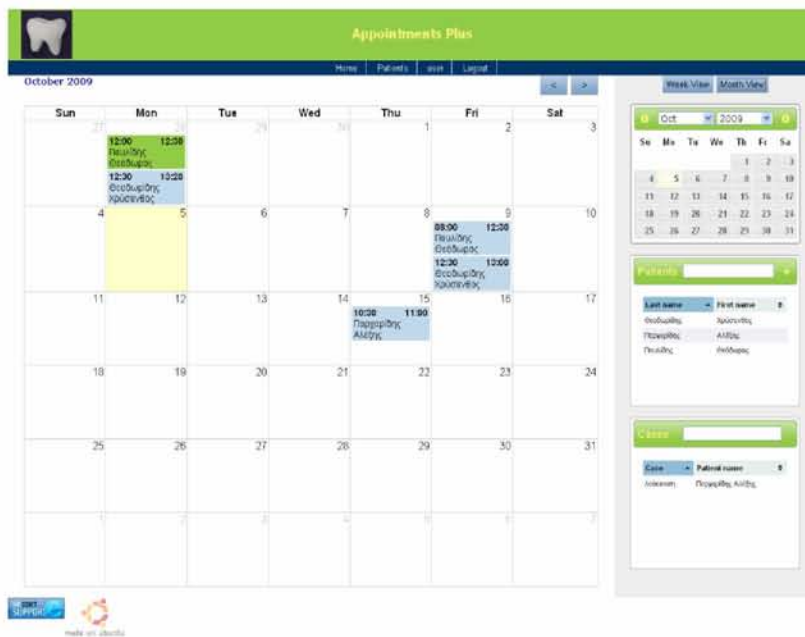
Σε αυτό το κεφάλαιο θα δούμε και ένα ολοκληρωμένο σενάριο χρήσης. Θα ξεκινήσουμε από τη ιστοσελίδα εισόδου, έπειτα θα δημιουργήσουμε ένα νέο ασθενή και τέλος θα δημιουργήσουμε ένα νέο ραντεβού.

1) Όταν κανείς πρωτοεπισκέπτεται το δικτυακό τόπο της εφαρμογής, συναντά μια ιστοσελίδα της μορφής της Εικόνα 4-5 που τον παροτρύνει να εισέλθει στην εφαρμογή δίνοντας το προσωπικό του όνομα χρήστη και τον κωδικό του. Τα ονόματα χρήστη εκδίδονται από τον διαχειριστή της εφαρμογής.



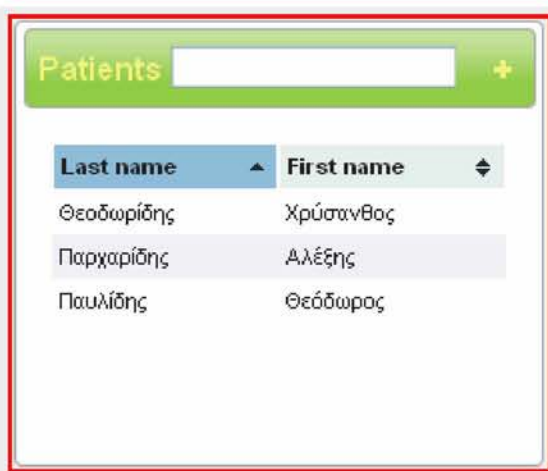
Εικόνα 4-5: Η ιστοσελίδα εισαγωγής στοιχείων χρήστη.

Τελικά αφού ο χρήστης εισάγει τα στοιχεία του, βρίσκεται στην κεντρική ιστοσελίδα, όπως την είδαμε και πιο πάνω στην Εικόνα 4-1.



**Μικρογραφία της
Εικόνα 4-1.**

2) Στο δεξιό μέρος μπορούμε να διακρίνουμε ένα πλαίσιο με ασθενείς όπως της Εικόνα 4-6. Αυτό το gadget προσφέρει στο χρήστη λειτουργίες άμεσης αναζήτησης και δημιουργίας ασθενών. Η αναζήτηση ενός παλαιότερου ασθενή γίνεται μέσω του πεδίου κειμένου (η αναζήτηση είναι τύπου as you type), είτε με βάση το όνομα είτε το επίθετο του ασθενή. Στο δικό μας σενάριο χρήσης θα χρησιμοποιήσουμε το πλήκτρο + για να προσθέσουμε ένα νέο ασθενή.



Εικόνα 4-6: Το gadget των ασθενών.

3) Το περιεχόμενο της ιστοσελίδας τότε θα αλλάξει εμφανίζοντας την μία φόρμα πληροφοριών όπως της Εικόνα 4-7.

Εικόνα 4-7: Η ιστοσελίδα εισαγωγής των στοιχείων ενός νέου ασθενή.

Συμπληρώνουμε τα πεδία και πατάμε το πλήκτρο “Save Patient” οπότε αποθηκεύεται ο νέος μας ασθενής και επανεμφανίζεται η προβολή των ραντεβού του μήνα.

Σημειώνουμε ότι για τη συμπλήρωση της ημερομηνίας γέννησης ο χρήστης μπορεί να χρησιμοποιήσει το αναδυόμενο ημερολόγιο.

4) Κάνοντας κλικ σε κενό σημείο οποιασδήποτε ημέρας, είτε βρισκόμαστε σε μηνιαία προβολή είτε σε εβδομαδιαία, το περιεχόμενο της ιστοσελίδας αλλάζει στο παρακάτω πλαίσιο παιδιών. Αρχίζοντας να πληκτρολογούμε το όνομα του ασθενή, εμφανίζονται

προτεινόμενες καταχωρήσεις ασθενών, όπως φαίνεται και στην Εικόνα 4-8. Επιλέγουμε την καταχώρηση που δημιουργήσαμε στο προηγούμενο βήμα και προχωρούμε στο πεδίο των περιστατικών.

Patient
θ
Παυλίδης Θεόδωρος
Θεοδωρίδης Χρύσανθος
Γρεσσιδής Θεόδωρος

Start Date/Time: 2009-10-05 11:00
Finish Date/Time: 2009-10-05 11:45

Comments
Diagnose
Treatment

Completed

Save Appointment

Εικόνα 4-8: Η ιστοσελίδα δημιουργίας ενός ραντεβού.

Όμοια και εδώ υπάρχουν προτάσεις αυτόματης συμπλήρωσης, αλλά για το σκοπό του σεναρίου μας, αρκεί να επιλέξουμε την πρόταση “New Case”, όπως στην Εικόνα 4-9, και έπειτα να συμπληρώσουμε το όνομα που θέλουμε να δώσουμε στο νέο περιστατικό.

Case
t|
Default Case
New Case

Case
Enter a name for the new case here...

Case
καθαρισμός

Εικόνα 4-9: Η δημιουργία ενός νέου περιστατικού.

Αν δε θέλαμε να δημιουργήσουμε ανεξάρτητο περιστατικό για το ραντεβού, θα μπορούσαμε είτε να επιλέξουμε την πρόταση “Default Case” είτε να αφήσουμε το πεδίο κενό, οπότε και υπονοείται το περιστατικό “Default Case”.

Επιλέγοντας την ώρα έναρξης, την εκτιμωμένη ώρα λήξης του ραντεβού και όποιο άλλο πεδίο κρίνουμε απαραίτητο, πατάμε το πλήκτρο “Save Appointment” και δημιουργείται με αυτό τον τρόπο το ραντεβού.

5) Μπορούμε πλέον να δούμε το νεοδημιουργηθέν μας ραντεβού μέσα και από τις δύο μορφές προβολής. Παρακάτω παραθέτουμε την προβολή “Month View” στην Εικόνα 4-10 και “Week View” στην Εικόνα 4-11.

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28 12:00 12:30 Παυλίδης Θεόδωρος 12:30 13:20 Θεοδωρίδης Χρυσάνθος	29	30	1	2	3
4	5	6	7	8	9 08:00 12:30 Παυλίδης Θεόδωρος 12:30 13:00 Θεοδωρίδης Χρυσάνθος	10 11:00 11:45 Γρεσσιδής Θεόδωρος
11	12	13	14	15 10:30 11:00 Παρχαρίδης Αλέξης	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Εικόνα 4-10: Η τελική μορφή της προβολής Month View με το νέο ραντεβού.

	Sunday Oct 04, 2009	Monday Oct 05, 2009	Tuesday Oct 06, 2009	Wednesday Oct 07, 2009	Thursday Oct 08, 2009	Friday Oct 09, 2009	Saturday Oct 10, 2009
8AM						08:00 to 12:30 Παράκληση Θεσσαλονίκης	
9AM							
10AM							
11AM							11:00 to 11:45 Γραπείο της Επικοινωνίας
12PM							
1PM						12:30 to 13:00 Θεοφιλιώδης	
2PM							
3PM							

Εικόνα 4-11: Η τελική μορφή της προβολής Week View με το νέο ραντεβού.

5

Σύνοψη και μελλοντικές προοπτικές

Αρχικά αξιολογήσαμε το βαθμό εξοικείωσης των ιατρών με το Web και κατά πόσο αυτό έχει επηρεάσει την πρακτική της ιατρικής. Είδαμε ότι ο κύριο λόγος ενασχόλησης των ιατρών με το Web είναι η πληροφόρηση, καθώς δεν υπάρχουν υποδομές, ώστε η υπάρχουσα κοινότητα να ενεργοποιηθεί. Η ομάδα των φοιτητών ιατρικής όμως είναι πολύ πιο ενεργή διαδικτυακά, κάτι που προδίδει τις τάσεις που θα επικρατήσουν τα επόμενα χρόνια στην πρακτική της ιατρικής.

Έπειτα αναφερθήκαμε στις εφαρμογές που χρησιμοποιούν οι ιατροί στις μέρες μας και εντοπίσαμε το κενό της αγοράς σε αυτό τον τομέα, αξιολογώντας παράλληλα τι πλεονεκτήματα προσφέρει η δική μας εφαρμογή.

Τα πλεονεκτήματα που αυτή μπορεί να αποτελέσουν ισχυρά κίνητρα υιοθέτησης της εφαρμογής. Κάποια από αυτά είναι:

1. Δυνατότητα πρόσβασης στο σύνολο των λειτουργιών της εφαρμογής, από οποιοδήποτε σημείο υπάρχει σύνδεση στο διαδίκτυο. Έτσι ο χρήστης δεν χρειάζεται να αντιγράψει και να συγχρονίζει τις επαφές και τα ραντεβού του μεταξύ πολλαπλών υπολογιστών.
2. Εξαιλείει την ανάγκη λήψης και διατήρησης αντιγράφων ασφαλείας από τη μεριά του χρήστη, καθώς όλες οι πληροφορίες βρίσκονται στο Server της εφαρμογής. Έτσι ο χρήστης αποφεύγει και τη διαδικασία αναβάθμισης των αποθηκευτικών του μέσων όσο τα δεδομένα της εφαρμογής αυξάνονται.

3. Δημιουργία μιας online κοινότητας ιατρών, όπου οι ιατροί μπορούν να ανταλλάσσουν οποιουδήποτε είδους πληροφορίες. Οι ιατροί αυτοί σαν χρήστες της ίδιας εφαρμογής και πιθανόν ίδια ειδικότητας, αποτελούν μια κοινότητα η οποία μπορεί πλέον να έρθει σε επαφή και να ενεργοποιηθεί.

Είδαμε το περιβάλλον χρήστη καθώς και ένα σενάριο λειτουργίας αυτής της Web εφαρμογής, η οποία προσφέρει την πλειονότητα των λειτουργιών της, κάνοντας αποκλειστική χρήση της τεχνολογίας AJAX. Υπήρξε, δε, ενθουσιασμός με το Django και την ανάπτυξη με αυτό λόγω της καλής δομής του και την ευκολία εκμάθησης προγραμματισμού με αυτό. Το υπάρχων documentation κρίνεται αρκετά επαρκές και το tutorial αναπτύσσει τους τομείς που κανείς θα ήθελε να μάθει πρώτα ώστε να παράγει γρήγορα κάτι που να λειτουργεί. Μετά την πρώτη ιστοσελίδα η πολυπλοκότητα των λειτουργιών, που αναλύονται μεγάλωνει γραμμικά, οπότε σταδιακά ο χρήστης εξοικειώνεται και με πολύ προχωρημένες λειτουργίες του Django χωρίς πρόβλημα. Λόγω της πολύ καλής δομής του και του διαχωρισμού των λειτουργιών αλλά και λόγω της παροχής έτοιμων βιβλιοθηκών για πολύπλοκα θέματα, το Django ίσως μπορεί να θεωρηθεί και αρκετά καλό διδασκτικό εργαλείο. Πιο συγκεκριμένα, κάνει εύκολα κατανοητό τον τρόπο λειτουργίας μιας Web εφαρμογής αλλά και κατατοπίζει τον προγραμματιστή στην αρχιτεκτονική του βοηθώντας τον να αναπτύξει γρήγορα και δομικά σωστά μια Web εφαρμογή.

Τα περιθώρια εξέλιξης της εφαρμογής προφανώς είναι ανεξάντλητα, τόσο λόγω των πολλών ειδικοτήτων των ιατρών, όσο και λόγω των προσωπικού τρόπου άσκησης της ιατρικής από τον κάθε ιατρό. Παρακάτω θα αναφέρουμε μερικούς τομείς που μπορούν άμεσα να βελτιωθούν.

1. Δυνατότητα εισαγωγής εικόνας ασθενή.
2. Δυνατότητα επιλογής χρώματος ραντεβού, ώστε να γίνεται καλύτερη οργάνωση όσον αφορά το φόρτο εργασίας.
3. Συγκεντρωτικό menu ρυθμίσεων της εφαρμογής, για κάθε χρήστη.
4. Δυνατότητα εντοπισμού και επικοινωνίας μεταξύ ιατρών.
5. Προτυποποίηση φόρμας ιστορικού ασθενή.
6. Δυνατότητα κοινοποίησης του ιστορικού ασθενών σε άλλους ιατρούς, ώστε να επιταχυνθεί η διαδικασία της παραπομπής.
7. Δημιουργία forum ιατρών.
8. Εισαγωγή στη δεξιά μεριά της κεντρικής ιστοσελίδας ενός gadget με ιατρικά νέα, για την εξυπηρέτηση περισσότερων αναγκών από μία μόνο Web εφαρμογή.

6

Αναφορές

1. <http://www.medicalstudents.in/>
2. <http://www.bsoft.gr/bsDent.asp?lang=el>
3. <http://www.dentalwin.gr/>
4. <http://www.djangoproject.com/>
5. <http://jquery.com/>
6. <http://www.last.fm/>
7. <http://www.facebook.com/>
8. <http://en.wikipedia.org/wiki/Smartphones>
9. <http://www.apple.com/iphone/>
10. <http://www.htc.com/www/product/magic/>
11. <http://en.wikipedia.org/wiki/Wifi>
12. http://en.wikipedia.org/wiki/Hotspot_%28Wi-Fi%29
13. <http://prototypejs.org/>
14. <http://dojotoolkit.org/>
15. <http://mootools.net/>
16. <http://plugins.jquery.com/>
17. <http://jqueryui.com/>

18. <http://arshaw.com/fullcalendar/>
19. <http://www.redredred.com.au/projects/jquery-week-calendar/>
20. <http://bassistance.de/jquery-plugins/jquery-plugin-autocomplete/>
21. <http://valums.com/ajax-upload/>
22. <http://www.compulsivoco.com/2008/08/tablesorter-filter-results-based-on-search-string/>