



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Πληροφορίες παιδικών ασθενειών και αναζήτηση φαρμακείου και παιδίατρου μέσω mobile και web εφαρμογής με χρήση web services.
Όνοματεπώνυμο Φοιτητή	ΚΟΥΓΙΟΥΜΤΖΗΣ ΒΑΣΙΛΕΙΟΣ
Πατρώνυμο	ΑΝΤΩΝΙΟΣ
Αριθμός Μητρώου	ΜΠΠΛ.12023
Επιβλέπων	Ευθύμης Αλέπης

Ημερομηνία Παράδοσης

Ιούνιος 2016

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Περιεχόμενα

Κεφάλαιο 1: Εισαγωγή.....	7
Λειτουργικές απαιτήσεις εφαρμογής.....	7
<i>Εμφάνιση πληροφοριών για παιδικές ασθένειες</i>	7
<i>Εύρεση κοντινού φαρμακείου – παιδίατρο</i> - Κάθε χρήστης της mobile εφαρμογής μπορεί να αναζητήσει κοντινό φαρμακείο η παιδίατρο.....	8
Μη λειτουργικές απαιτήσεις.....	8
Διάγραμμα περιπτώσεων χρήσης.....	9
Σχήμα : Διάγραμμα περιπτώσεων χρήσης.....	9
Κεφάλαιο 2 : Ανασκόπηση πεδίου	10
Εισαγωγή.....	10
Location-based services	11
Mobile Εφαρμογές στην Ελλάδα	13
Εφαρμογή Farmakeia	13
Εφαρμογή Doctoranytime.....	14
Κεφάλαιο 3 : Αρχιτεκτονική συστήματος	17
Εισαγωγή.....	17
Σχήμα : Σχεδιάγραμμα Υλοποίησης εφαρμογής	18
Γενικά θέματα αρχιτεκτονικής.....	19
Model–view–controller (MVC).....	19
Web Services	24
JSON format	24
Μέρος 1 ^ο : Web Services , Database Server και administration website	26
Εισαγωγή.....	26
MySQL Database	28
PHP Services	31
Μέρος 2 ^ο : Android Application	35
Δομή κώδικα εφαρμογής android	36
Offline λειτουργία και συγχρονισμός	39
Μέρος 3 ^ο : Web Application.....	42
Δομή web εφαρμογής.....	44
Εγκατάσταση web εφαρμογής.....	47

Κεφάλαιο 4: Παρουσίαση και χρήση εφαρμογής	51
Εφαρμογή Android.....	51
Κεντρικό μενού επιλογών	51
Επιλογή εμφάνισης κατηγοριών	52
Επιλογή ασθένειας.....	53
Λεπτομέρειες ασθένειας.....	54
Αναζήτηση συμπτώματος	55
Αναζήτηση Φαρμακείου – Γιατρού.....	56
Web εφαρμογή angular js.....	58
Κεφάλαιο 5: Συμπεράσματα και μελλοντικές επεκτάσεις	59
Συμπεράσματα.....	59
Μελλοντικές επεκτάσεις.....	60
Βιβλιογραφία	61
Παράρτημα	62
Εγκαταστάσεις εφαρμογής.....	62
Repositories κώδικα εφαρμογής στο GITHUB	62

Περίληψη

Η επιστήμη της πληροφορικής υπήρξε όλα αυτά τα χρόνια μια επιστήμη που προσέφερε κρίσιμες εφαρμογές σε όλους τους τομείς της ανθρώπινης δραστηριότητας. Ειδικότερα σε αυτή τη πτυχιακή εργασία επικεντρωθήκαμε στην εκμετάλλευση των πολλών δυνατοτήτων που προσφέρουν πλέον οι mobile και web εφαρμογές στην υποστήριξη της ιατρικής επιστήμης και συγκεκριμένα στους τομείς της πρόληψης και ενημέρωσης.

Σκοπός της πτυχιακής εργασίας είναι η ανάπτυξη μιας mobile εφαρμογής που προσπαθεί να καλύψει πληροφοριακές ανάγκες των χρηστών σχετικά με συχνές παιδικές ασθένειες ανά κατηγορία καθώς και χωρική αναζήτηση κοντινών φαρμακείων και παιδιάτρων εκμεταλλευόμενη Location-based services (LBS). Θα πρέπει να τονιστεί ότι η παρούσα διπλωματική εργασία δίνει το μεγαλύτερο βάρος στο τεχνικό κομμάτι της υλοποίησης και όχι τόσο στον εμπλουτισμό του πληροφοριακού περιεχομένου της εφαρμογής αφού αυτό θα απαιτούσε επιπρόσθετη μελέτη και επιμέλεια από αρμόδιο επιστήμονα εξειδικευμένο ιατρό σε θέματα παιδιατρικής. Συνεπώς σκοπός της εργασίας είναι η παρουσίαση της τεχνικής υποδομής μιας τέτοιας εφαρμογής από την σκοπιά της επιστήμης της πληροφορικής και πώς αυτή μπορεί να παρέχει χρήσιμες υπηρεσίες στην ιατρική επιστήμη.

Τεχνολογικά η εφαρμογή προσπαθεί να εκμεταλλευτεί και καταδείξει την χρησιμότητα των web services τα οποία είναι ικανά να “σερβίρουν” δεδομένα σε clients που έχουν αναπτυχθεί σε διαφορετικές πλατφόρμες και γλώσσες υλοποίησης. Συγκεκριμένα στην παρούσα εργασία οι clients που θα “καταναλώνουν” τα web services είναι μια mobile εφαρμογή android (java) και μια web εφαρμογή .NET5 / ANGULAR JS η οποία θα φιλοξενηθεί σε cloud περιβάλλον.

Η πτυχιακή χρησιμοποιώντας τις τελευταίες τάσεις και τεχνολογίες στην ανάπτυξη mobile και web εφαρμογών προσπαθεί να αναδείξει μια σύγχρονη και ολοένα αναπτυσσόμενη τεχνική ανάπτυξης εφαρμογών βασισμένες σε http web services που θα μπορούσε δυνητικά να χρησιμοποιηθεί για την δημιουργία οποιασδήποτε παρόμοιας εφαρμογής που καλύπτει πληροφοριακές ανάγκες σε οποιοδήποτε τομέα.

Τα web services είναι το παρόν και το μέλλον στην ανάπτυξη mobile και web εφαρμογών και θα αναλύσουμε στα επόμενα κεφάλαια τις τεχνικές για την δημιουργία της υποδομής μιας τέτοιας mobile και web εφαρμογής.

Abstract

Computer science has been over the years a science that offered critical applications in all areas of human activity. Especially in this dissertation we focus on exploiting the growing opportunities offered by mobile and web applications to support medical science particularly in the fields of prevention and information.

The aim of this dissertation is to develop a mobile application that tries to cover the information needs of users for common childhood diseases by category and spatial search for nearby pharmacies and pediatricians by exploiting Location-based services (LBS). It should be noted that this dissertation emphasizes on the technical part of implementation rather than the enrichment of the application's content as this would require additional study and diligence by a competent scientist specialist in pediatric subjects. Therefore the purpose of this study is the presentation of the technical infrastructure for such an application from the standpoint of computer science and how it can provide useful services in medical science.

Technologically the application tries to exploit and demonstrate the usefulness of web services that are capable of "serving" data on clients that have been developed in different frameworks and programming languages. Specifically the clients that will "consume" web services is a mobile application android (java) and a web application .NET5 / ANGULAR JS which will be host at a cloud environment.

By using the latest trends and technologies in the development of mobile and web applications this dissertation presents a modern and increasingly growing way of creating applications based on http web services that could potentially be used for the creation of any similar application. Web services are the present and future in developing mobile and web applications and in the following chapters techniques will be presented to create the infrastructure for this type of applications.

Κεφάλαιο 1: Εισαγωγή

Στην παρούσα πτυχιακή γίνεται η ανάπτυξη της εφαρμογής “Sickid” σε mobile και web περιβάλλον. Στην εφαρμογή Sickid θα βρείτε πληροφορίες που αφορούν τη υγεία του παιδιού και συγκεκριμένα ασθένειες που συνήθως ταλαιπωρούν τα παιδιά χωρισμένες σε κατηγορίες .

Στο mobile κομμάτι της εφαρμογής μπορείτε επίσης να αναζητήσετε κάποια ασθένεια η σύμπτωμα και τέλος να αναζητήσετε κοντινό φαρμακείο η παιδίατρο με την βοήθεια τους gps και την χρήση ενσωματωμένου στην εφαρμογή google map με δεδομένα ιατρείων και φαρμακείων από τον server της εφαρμογής.

Η υλοποίηση της εφαρμογής συνδυάζει γνώσεις από πολλά πεδία της πληροφορικής και διαφορετικές τεχνολογίες και γλώσσες. Τεχνικά η εφαρμογή χρησιμοποιεί την τεχνολογία των web services ώστε να καλύψει τις ανάγκες της mobile εφαρμογής όσο και της web εφαρμογής σε δεδομένα. Η χρήση κοινών web services σε δυο εντελώς διαφορετικών τύπων client αποτελεί τον κύριο τεχνικό στόχο της εφαρμογής.

Ακολουθεί μια ανάλυση απαιτήσεων που περιγράφει τις ανάγκες που ευελπιστεί να καλύψει η εφαρμογή

Λειτουργικές απαιτήσεις εφαρμογής.

Εμφάνιση πληροφοριών για παιδικές ασθένειες

- Κάθε χρήστης της mobile και web εφαρμογής θα μπορεί να δει τις πληροφορίες σχετικά με την ασθένεια που επιθυμεί
- Κάθε χρήστης της mobile εφαρμογής θα μπορεί να αναζητά τη συγκεκριμένη ασθένεια μέσω κατηγοριών στις οποίες είναι καταχωρημένες η πιο άμεσα μέσω της επιλογής για αναζήτηση. Θα πρέπει να μπορεί να αναζητήσει συμπτώματα .

Εύρεση κοντινού φαρμακείου – παιδίατρο

- Κάθε χρήστης της mobile εφαρμογής μπορεί να αναζητήσει κοντινό φαρμακείο η παιδίατρο.
- Πρέπει να υπάρχει αυτόματος εντοπισμός της τοποθεσίας του χρήστη καθώς και χειροκίνητη επιλογή εντοπισμού.
- Πρέπει να υπάρχουν φίλτρα κατηγορίας την οποία αναζητεί
- Πρέπει να υπάρχει φίλτρο μέγιστης απόστασης αναζήτησης
- Πρέπει να παρέχονται οι κατάλληλες πληροφορίες στα αποτελέσματα αναζήτησης (τηλέφωνο ,διεύθυνση)

Μη λειτουργικές απαιτήσεις

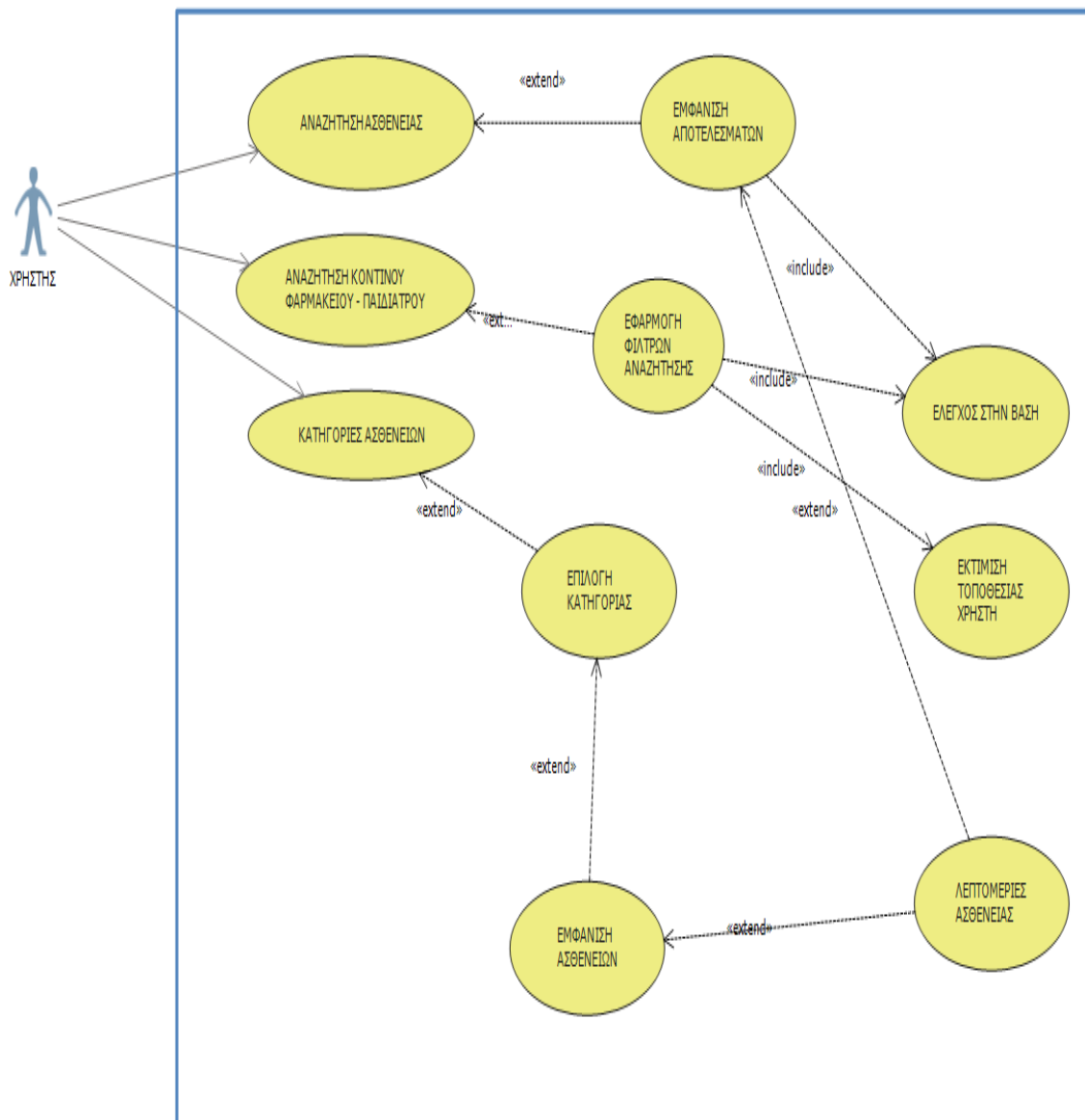
- Οι ασθένειες θα χωρίζονται σε κατηγορίες
- Η αναζήτηση στις ασθένειες θα αναζητά αποτελέσματα που ταιριάζουν μερικώς με το input του χρήστη όχι ολοκληρωτικά.
- Ο εντοπισμός τοποθεσίας του χρήστη θα πρέπει να γίνεται με όλους τους δυνατούς τρόπους (wifi, cellular network , GPS)
- Ο εντοπισμός τοποθεσίας θα πρέπει να είναι επαναλαμβανόμενος σε λογικό πλαίσιο χρόνου (interval) και να παύει εντελώς όταν η εφαρμογή είναι ανενεργή για να μην γίνεται κατασπατάληση μπαταρίας.
- Η εφαρμογή android πρέπει να διαθέτει τοπική βάση δεδομένων για δυνατότητα εμφάνισης των δεδομένων offline
- Σε περίπτωση μη αλλαγής των δεδομένων στο server να γίνεται χρήση της τοπικής βάσης ώστε να αποφεύγεται άσκοπη χρήση του δικτύου.

Διάγραμμα περιπτώσεων χρήσης

Το διάγραμμα περιπτώσεων χρήσης αναπαριστά τις λειτουργίες της εφαρμογής από την οπτική γωνία ενός χρήστη. Βοηθά στην κατανόηση του γενικού σκοπού της εφαρμογής και μας καθιστά ικανούς να αντιληφθούμε το μέγεθος της συνολικής εφαρμογής. Μια περίπτωση χρήσης αντιστοιχεί σε ένα συγκεκριμένο είδος χρήσης της εφαρμογής. Είναι μια εικόνα της λειτουργικότητας ενός συστήματος το οποίο ενεργοποιείται για να ανταποκριθεί σε έναν εξωτερικό ενεργοποιό (actor).

Έτσι λοιπόν, στην εφαρμογή αυτή, το διάγραμμα περιπτώσεων χρήσης είναι το παρακάτω:

Σχήμα : Διάγραμμα περιπτώσεων χρήσης



Κεφάλαιο 2 : Ανασκόπηση πεδίου

Εισαγωγή

Η επιστήμη της πληροφορικής είναι πλέον αναγκαία και για πολλούς άλλους τομείς εκτός από αυτόν των επιχειρήσεων. Πλέον η πληροφορική βρίσκει εφαρμογή σε επιστήμες όπως η Ιατρική και η Βιολογία.^[12]

Τα τελευταία χρόνια γίνεται ολοένα και πιο έντονη στο χώρο της υγείας η αναγκαιότητα της αναβάθμισης της ποιότητας των προσφερόμενων υπηρεσιών με ταυτόχρονη μείωση του κόστους τους. Για το σκοπό αυτό, πέρα από διαδικασίες επιχειρηματικού ανασχεδιασμού (business process reengineering) σημαντική βοήθεια έρχονται να προσφέρουν και οι τεχνολογίες τηλεπικοινωνιών και πληροφορικής. Έτσι, ολοένα και περισσότερα νοσοκομεία διεθνώς έχουν σε μεγάλο βαθμό υλοποιήσει ολοκληρωμένα πληροφοριακά συστήματα για την κάλυψη των αναγκών διαχείρισης τόσο διαχειριστικών και οικονομικών, όσο και ιατρικών δεδομένων. Τα πληροφοριακά αυτά συστήματα δίνουν την δυνατότητα αυτοματοποίησης μεγάλου αριθμού διαδικασιών (π.χ. χρεώσεων, παραγγελιοδοσίας / αποτελέσματα, κλπ.) ενώ ταυτόχρονα επιτρέπουν την μηχανογραφική διαχείριση του φακέλου του ασθενούς (Electronic Patient Record - EPR). Το τελευταίο επιτυγχάνεται μέσω των λεγόμενων Κλινικών Πληροφοριακών Συστημάτων (Clinical Information Systems) που δίνουν την δυνατότητα μηχανογραφικής παρακολούθησης του συνόλου της ιατρικής πληροφορίας ενός ασθενή συμπεριλαμβανομένων του ιστορικού, στοιχείων κλινικής εξέτασης, αποτελεσμάτων παρακλινικών εξετάσεων, στοιχεία απεικονιστικών εξετάσεων, πορίσματα και διαγνώσεις, κλπ. Σημαντική προϋπόθεση στην διαχείριση ευαίσθητων προσωπικών δεδομένων, όπως είναι τα ιατρικά δεδομένα, αποτελεί η διασφάλιση του ιατρικού απορρήτου και η προστασία των προσωπικών δεδομένων που επιτυγχάνεται μέσω συστημάτων και τεχνικών ιεραρχικής πρόσβασης στα δεδομένα σύμφωνα με το "ρόλο" κάθε χρήστη στην ροής εργασίας εντός του νοσοκομείου. Οι τεχνολογίες "έξυπνων καρτών" (smart-cards) τόσο επαγγελματικών (professional) όσο και καρτών υγείας (health cards) διασφαλίζουν την πρόσβαση στα προσωπικά δεδομένα μόνο κατόπιν εξουσιοδότησης του ίδιου του ασθενούς και μόνο στο ιατρικό και νοσηλευτικό προσωπικό που έχει την ευθύνη παρακολούθησης του ασθενούς.^[13]

Η συμβολή της επιστήμης της πληροφορικής στην ιατρική επιστήμη με τα παραδοσιακά ERP, MIS συστήματα έχουν βοηθήσει στο να καλύψουν τις ιδιαίτερες ανάγκες ενός νοσηλευτικού οργανισμού. Όμως τα τελευταία χρόνια η σημαντική αύξηση πωλήσεων των έξυπνων κινητών συσκευών (smartphones) και tablets ανοίγουν συνεχώς νέους ορίζοντες στην υποστήριξη της ιατρικής επιστήμης με μια μεγάλη ποικιλία νέων εφαρμογών. Οι mobile εφαρμογές έχουν γίνει κομμάτι της καθημερινότητας μας αλλά και ένα ισχυρό εργαλείο σε όλους τους τομείς που υποστηρίζονται από τις εφαρμογές πληροφορικής όπως είναι και η Ιατρική επιστήμη στην περίπτωση μας.

Οι κινητές συσκευές είναι πλέον αρκετά ισχυρές και εύχρηστες δίνοντας δυνατότητες για ανάπτυξης εφαρμογών με δυνατότητες σύνδεσης σε βάσεις δεδομένων για παροχή άμεσης πληροφόρησης καθώς επίσης και Location-based services (LBS) .

Location-based services

«Με τον όρο Location Based Services (LBS) περιγράφεται ένα ευρύτερο σύνολο εφαρμογών και τεχνολογιών που στοχεύουν στην παροχή προσωποποιημένων υπηρεσιών προς τελικούς χρήστες, βάσει της γεωγραφικής τους θέσης. Περιλαμβάνει και συνδέεται με μια πλειάδα σχετικών ερευνητικών περιοχών, όπως: profile matching, context management, geographic information systems, information retrieval και web services. Τα LBS αποτελούν μια σχετικά νέα κατηγορία συστημάτων για την παροχή υπηρεσιών προστιθέμενης αξίας. Έτσι, ο χρήστης της υπηρεσίας μπορεί να αναζητήσει ένα σημείο ενδιαφέροντος και να πάρει τα καταστήματα που είναι κοντά του, καθώς και τον τιμοκατάλογό τους, τόσο σε text μορφή (όπως σε μια κλασσική μηχανή αναζήτησης), όσο και σημειωμένα πάνω σε έναν χάρτη, με οδηγίες για το πώς θα πάει σε αυτά.

Αυτή η σύγκλιση των μηχανών αναζήτησης, των συστημάτων γεωγραφικών πληροφοριών, αλγορίθμων για την επεξεργασία του προφίλ ενός χρήστη (π.χ. συσκευή, προσωπικά ενδιαφέροντα, γεωγραφική και σχετική θέση) και ταιριάσματός του (matching) με έναν πάροχο μιας υπηρεσίας, αφενός θα απασχολήσει την αγορά για τα επόμενα χρόνια, αφετέρου αποτελεί μια σημαντική ευκαιρία για την ολοκλήρωση υφιστάμενων τεχνολογιών και την ανάδειξη νέων κατευθύνσεων για την έρευνα στην ακαδημαϊκή κοινότητα.

Γενικά, ένα σύστημα LBS παρέχει απαντήσεις στις ακόλουθες ερωτήσεις:

- Που βρίσκομαι; Η γνώση της θέσης του χρήστη είναι θεμελιώδους σημασίας για την επιλογή των κοντινότερων σε αυτόν σημείων ενδιαφέροντος, πληροφοριών και υπηρεσιών. Η θέση ενός χρήστη μπορεί να απαντηθεί με τρεις τρόπους: (α) από το δίκτυο της κινητής τηλεφωνίας (cell-id των σταθμών βάσης - BTS) με ακρίβεια 50-200m (β) από το σύστημα GPS με ακρίβεια 4-40m και (γ) με beacons μικρής εμβέλειας (π.χ. bluetooth).

- Τι βρίσκεται κοντά μου; Η απάντηση στην ερώτηση μπορεί να είναι τόσο απλή όσο ένας χάρτης που απεικονίζει τα γειτονικά οικοδομικά τετράγωνα, ή περισσότερο σύνθετη: εύρεση ατόμων/οργανισμών/συσκευών που προσφέρουν υπηρεσίες/πληροφορίες για τις οποίες ενδιαφερόμαι.

- Πως θα πάω εκεί; Είναι επίσης εξαιρετικά σημαντικό, πέρα από το ταίριασμα παρόχου και καταναλωτή μιας υπηρεσίας, να δίνονται οδηγίες στο χρήστη για την ακριβή θέση της υπηρεσίας (π.χ. ένα φαρμακείο, πολυκατάστημα, ασύρματος εκτυπωτής, κτλ). Επίσης, ο χρήστης περιμένει επιπλέον πληροφορίες για το πώς μπορεί να κάνει χρήση των υπηρεσιών.

»^[14]

Η ανάγκη μεταδόσεως ιατρικών πληροφοριών, με εικόνες, δεδομένα και φωνή εξ αποστάσεως, μέσω διαφόρων τηλεπικοινωνιακών μέσων, έχει αναγνωριστεί προ πολλού. Το θέμα αποκτά νέο ενδιαφέρον εξαιτίας της προόδου που έχει επιτευχθεί στην τεχνολογία των τηλεπικοινωνιών και της πληροφορικής. Με βάση τη χρήση τηλεπικοινωνιακών και πληροφοριακών συστημάτων

και τη μετατροπή ιατρικής πληροφορίας σε ηλεκτρονική μορφή, διακρίνονται οι παρακάτω κύριες κατευθύνσεις υπηρεσιών και εφαρμογών:

Τηλεκπαίδευση, που καλύπτει τις ανάγκες του ενεργού ιατρικού και παραϊατρικού προσωπικού για συνεχή ενημέρωση σε διάφορους τομείς της ιατρικής. Επιπλέον εξασφαλίζεται εκπαίδευση του υγιούς πληθυσμού μέσω προγραμμάτων Αγωγής Υγείας, με σκοπό να διαμορφωθούν νέοι τρόποι συμπεριφοράς, όχι μόνο για την πρόληψη των νοσημάτων, αλλά και για την προστασία και προαγωγή της υγείας.

Τηλεδιάγνωση, που καλύπτει την από απόσταση μελέτη από ειδικούς των αποτελεσμάτων των ιατρικών εξετάσεων (ακτινογραφίες, εργαστηριακά ευρήματα κλπ) και τη σύνταξη σχετικών αναφορών.

Τηλεθεραπεία, που καλύπτει την από απόσταση παρακολούθηση ασθενών, όπου ο ασθενής επισκεπτόμενος την πλησιέστερη προς τον τόπο διαμονής του ιατρική μονάδα μπορεί να τυγχάνει ιατρικής φροντίδας από απομακρυσμένο ιατρικό κέντρο ως προς την πάθησή του.

Τηλεσυμβουλευτική, που καλύπτει την ανάγκη ανταλλαγής απόψεων καθώς και την οργάνωση συμβουλίων ειδικών ιατρών για την αντιμετώπιση συγκεκριμένων σύνθετων καταστάσεων όπου απαιτείται η ταυτόχρονη μελέτη της κατάστασης του ασθενούς από ειδικούς διαφορετικών ειδικοτήτων. ^[15]

Mobile Εφαρμογές στην Ελλάδα

Από τα παραπάνω γίνεται πλέον σαφές ότι mobile εφαρμογές που εκμεταλλεύονται Ιατρικά δεδομένα σε συνδυασμό με γεωγραφικά δεδομένα μπορούν να προσφέρουν χρήσιμες υπηρεσίες στον τομέα της άμεσης διάγνωσης και εύρεσης βοήθειας για τον χρήστη της εφαρμογής εξειδικευμένο ιατρό η και μη.

Δημοφιλής mobile και web εφαρμογές στον τομέα της ιατρικής πληροφορικής αποτελούν κυρίως εφαρμογές εύρεσης φαρμακείων και ιατρών με δυνατότητα κλεισίματος ακόμα και ραντεβού.

Μια δημοφιλή εφαρμογή με 100.000 και πλέον εγκαταστάσεις είναι το “Farmakeia”

Εφαρμογή Farmakeia



Από την περιγραφή της εφαρμογής:

«Ειδική εφαρμογή, iFarmakeia, για όλα τα Smartphones (iphone, android) με περισσότερους από 100.000 χρήστες. Η εφαρμογή κατατάσσεται 1η στην κατηγορία της (utilities) και βρίσκεται στην 1η δεκάδα των Top25 εφαρμογών στην Ελλάδα στο AppStore.

Εφημερεύοντα Φαρμακεία από Αθήνα, Πειραιά, Θεσσαλονίκη, Πάτρα, Ηράκλειο, Σέρρες και 33 ακόμη πόλεις.

Παρουσιάζονται ανά πάσα στιγμή τα εφημερεύοντα φαρμακεία που είναι ανοικτά την συγκεκριμένη ώρα (και όχι όλα τα εφημερεύοντα φαρμακεία) για να διευκολύνεται το κοινό στην

αναζήτηση του πλησιέστερου.

Επίσης, δίνεται η δυνατότητα ταξινόμησης με απόσταση (απαραίτητη προϋπόθεση είναι να έχετε ενεργοποιημένες τις υπηρεσίες τοποθεσίας) από το σημείο ή ανά περιοχή.

Η εφαρμογή λειτουργεί και offline αρκεί ο χρήστης να έχει ενημερώσει την λίστα της ημέρας.

Επιπλέον, κάθε φαρμακείο αποτυπώνεται με τα στοιχεία επικοινωνίας του και σε χάρτη (google maps) για τον ευκολότερο εντοπισμό του.

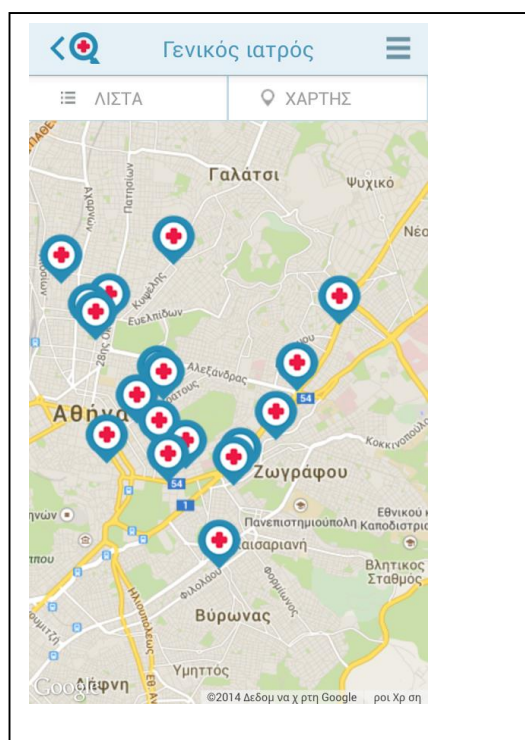
Επίσης, προβάλλονται μεγάλες προσφορές σε προϊόντα από Φαρμακεία για να εξοικονομήτε στις αγορές σας.»

Από την περιγραφή της εφαρμογής βλέπουμε πόσο χρήσιμη έως και σωτήρια μπορεί να καταστεί σε περίπτωση εκτατής ανάγκης για εύρεση του πιο κοντινού εφημερεύοντος φαρμακείου.

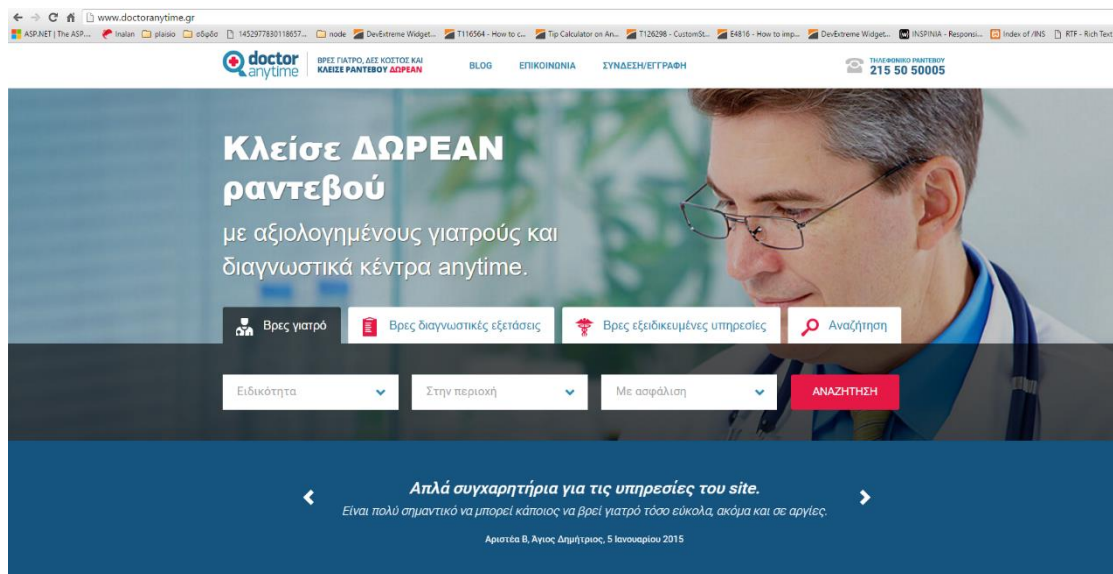
Στη συνέχεια η εφαρμογή Doctoranytime στοχεύει στην ανεύρεση γιατρού και κλείσιμο ραντεβού μέσω mobile αλλά και web εφαρμογής.

Εφαρμογή Doctoranytime

Mobile εφαρμογή Doctoranytime



Web εφαρμογή Doctoranytime



Από την περιγραφή της εφαρμογής:

«Το Doctoranytime έχει αλλάξει τον τρόπο με τον οποίο χιλιάδες χρήστες κλείνουν ραντεβού στον γιατρό τους και βρίσκεται συνεχώς δίπλα σε όποιον χρειάζεται ιατρική φροντίδα προσφέροντας μία καλύτερη κοινωνία υγείας και μία πιο εύκολη καθημερινότητα.

Κάνε εγγραφή δωρεάν και αποκτήσε πρόσβαση στις υπηρεσίες του Doctoranytime από κάθε κινητή συσκευή.

Χαρακτηριστικά:

1. Βρες αξιολογημένους γιατρούς και κλείσε ραντεβού σε 1'!
2. Επιλέγεις την ειδικότητα, την περιοχή, την ασφάλιση (ΕΟΠΥΥ / Με Ειδική Τιμή 20€/ Ιδιωτικά).
3. Βλέπεις την εμπειρία του γιατρού, το κόστος επίσκεψης, τη διαθεσιμότητα και κλείνεις ραντεβού σε πραγματικό χρόνο.
4. Η εφαρμογή θα κρατήσει το ιατρικό ιστορικό σου για να θυμάσαι τότε πηγες σε κάθε γιατρό.

Στο Doctoranytime, μπορείς να βρεις ιδιώτες ή ΕΟΠΥΥ γιατρούς, ακόμα και γιατρούς που συνεργάζονται με την ιδιωτική σου ασφάλεια.

Ψάξε για γιατρό κοντά σου ή σε οποιαδήποτε περιοχή θέλεις και κλείσε ραντεβού οποιαδήποτε στιγμή.

Βρες τον κατάλληλο γιατρό για σένα κρίνοντας από τις υπηρεσίες που παρέχει στο ιατρείο του, την εμπειρία και την εξειδίκευσή του και τις αξιολογήσεις ασθενών που τον έχουν επισκεφθεί. Μπορείς να μάθεις άμεσα το κόστος της επισκέψης ή οποιασδήποτε ιατρικής πράξης.

Το να βρεις γιατρό και να κλείσεις ραντεβού δεν ήταν ποτέ τόσο εύκολο!»

Όπως βλέπουμε πρόκειται για μια χρήσιμη εφαρμογή η οποία αξιοποιώντας LBS φέρνει σε επαφή ασθενή και ιατρό μέσα από ένα εύχρηστο mobile και web περιβάλλον.

Η εφαρμογή «Sickid» που αναπτύσσεται σε αυτήν την πτυχιακή προσπάθει να καλύψει μια πιο συγκεκριμένη ανάγκη η οποία επικεντρώνεται στην υγεία του παιδιού προσφέροντας όχι μόνο LBS αναζήτησης ιατρού και φαρμακείου αλλά και πληροφοριακά δεδομένα ασθενειών σε ένα απλό και εύχρηστο mobile και web περιβάλλον.

Κεφάλαιο 3 : Αρχιτεκτονική συστήματος

Εισαγωγή

Η υλοποίηση της υποδομής της εφαρμογής αποτελείται από τρία μεγάλα μέρη τα οποία έχουν αναπτυχθεί με διαφορετικά εργαλεία και εξυπηρετούν διαφορετικό σκοπό.

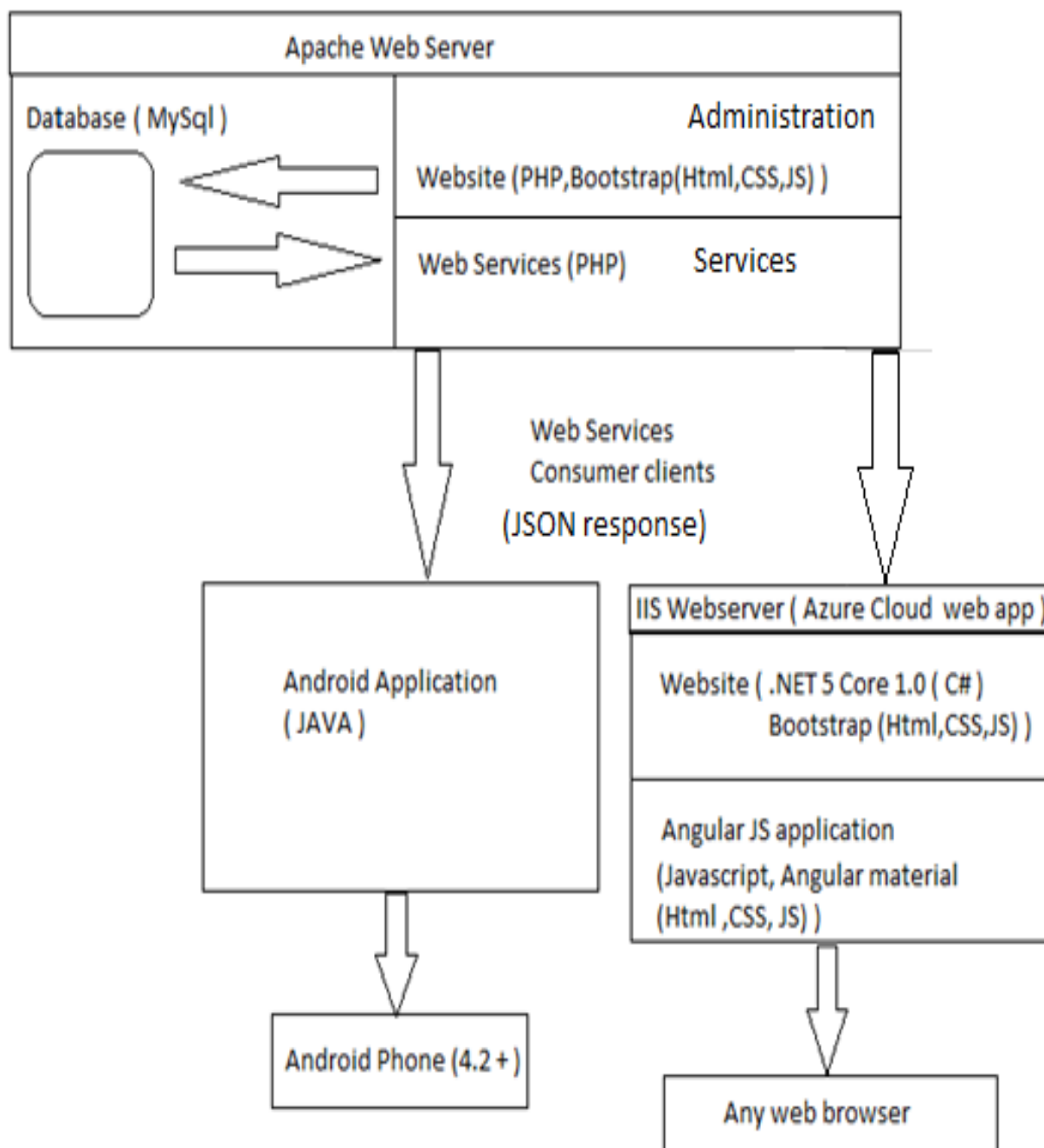
Το πρώτο μέρος της υλοποίησης αποτελεί την δημιουργία της εφαρμογής web η οποία φιλοξενεί τα web services την βάση δεδομένων της εφαρμογής και το περιβάλλον στο οποίο γίνεται η διαχείριση των δεδομένων της εφαρμογής (CRUD) . Τα δεδομένα αυτά αποτελούν τις κατηγορίες ασθενειών , τις ασθένειες και τα γεωγραφικά δεδομένα των φαρμακείων και παιδιάτρων.

Το δεύτερο μέρος της υλοποίησης αποτελεί την δημιουργία της mobile εφαρμογής η οποία καταναλώνει τα web services ώστε να δείχνει τις πληροφορίες των ασθενειών και να προσφέρει στο χρήστη δυνατότητες χωρικής αναζήτησης με την βοήθεια του gps .

Το τρίτο μέρος της υλοποίησης αποτελεί την δημιουργία ενός portal σε .NET5 που φιλοξενεί την angularJS web εφαρμογή η οποία καταναλώνει τα ίδια web services με την mobile εφαρμογής ώστε σε ένα πιο άνετο web περιβάλλον να δείχνει τις πληροφορίες των ασθενειών.

Συνολικά λοιπόν η υλοποίηση της εφαρμογής βασίζεται στα παρακάτω

Σχήμα : Σχεδιάγραμμα Υλοποίησης εφαρμογής



Γενικά θέματα αρχιτεκτονικής

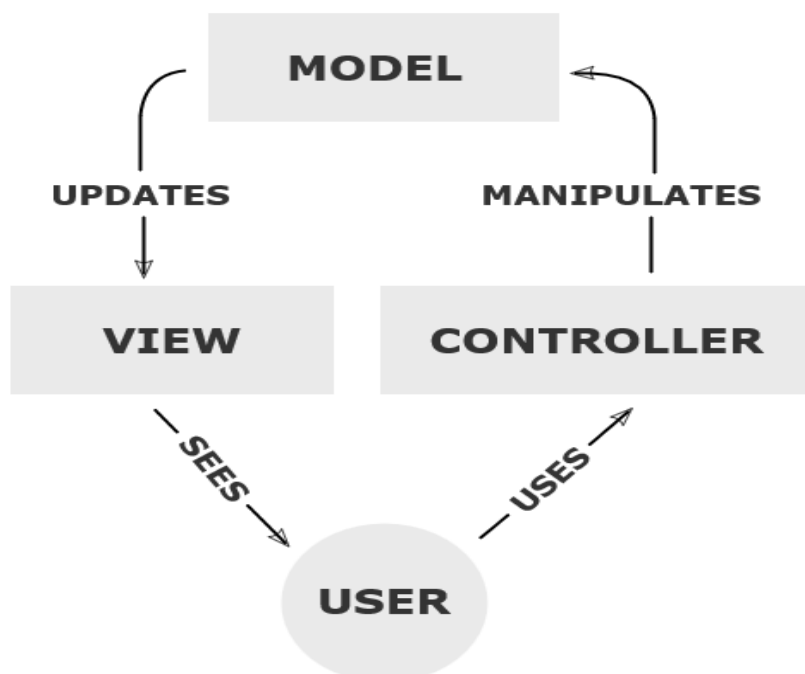
Πριν ξεκινήσουμε την ανάλυση των λεπτομερειών της υλοποίησης του κάθε μέρους της εφαρμογής κρίνω σκόπιμο να αναφερθούμε σε κάποιους γενικούς όρους στους οποίους θα συναντήσουμε αργότερα όπως το MVC pattern και τα web services.

Όπως θα παρατηρήσουμε στην δομή των εφαρμογών ακολουθείτε το MVC (Model View Controller) pattern.

Model-view-controller (MVC)

“Το **Model-view-controller** (σε συντομογραφία αναφέρεται ως **MVC**) είναι ένα μοντέλο αρχιτεκτονικής λογισμικού το οποίο χρησιμοποιείται για την δημιουργία περιβαλλόντων αλληλεπίδρασης χρήστη. Στο μοντέλο αυτό η εφαρμογή διαιρείται σε τρία διασυνδεδεμένα μέρη ώστε να διαχωριστεί η παρουσίαση της πληροφορίας στον χρήστη από την μορφή που έχει αποθηκευτεί στο σύστημα. Το κύριο μέρος του μοντέλου είναι το αντικείμενο *Model* το οποίο διαχειρίζεται την ανάκτηση/αποθήκευση των δεδομένων στο σύστημα. Το αντικείμενο *View* χρησιμοποιείται μόνο για να παρουσιάζεται η πληροφορία στον χρήστη (π.χ. με γραφικό τρόπο). Το τρίτο μέρος είναι ο *Controller* ο οποίος δέχεται την είσοδο και στέλνει εντολές στο αντικείμενο *Model* και στο *View*.” [\[1\]](#) [\[2\]](#)[\[3\]](#)

Mvc pattern diagram



«Εκτός από το να διαιρείται η εφαρμογή σε τρία μοντέλα, η σχεδίαση *model-view-controller* ορίζει και τις αλληλεπιδράσεις των μοντέλων [\[4\]](#)

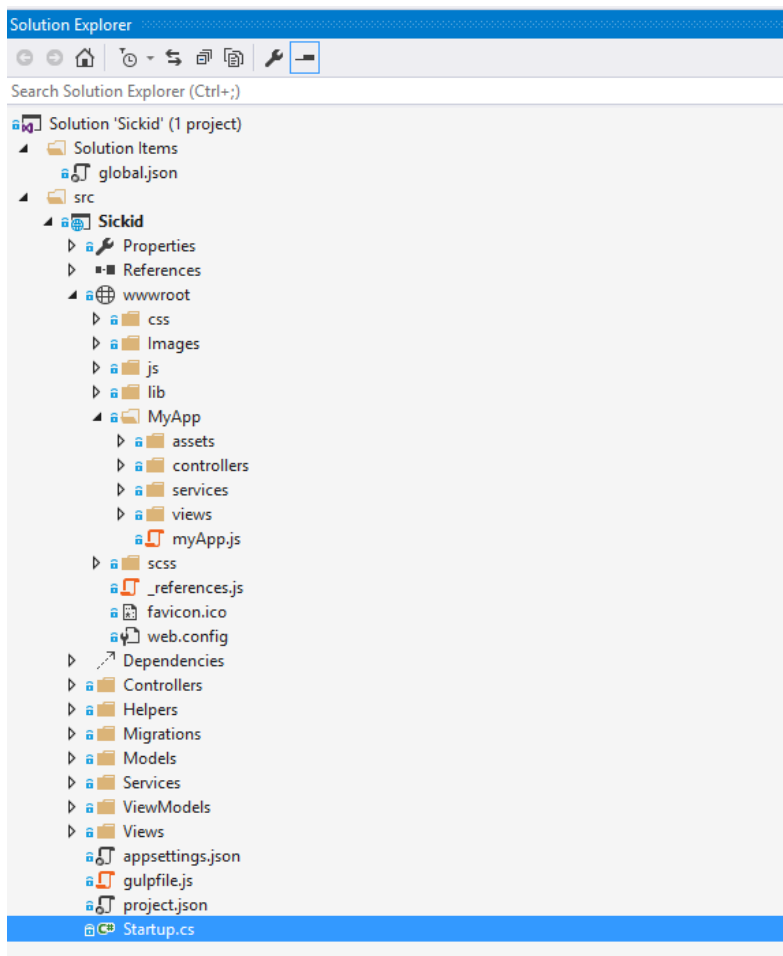
- Ο **controller** μπορεί να στέλνει εντολές στο μοντέλο και να ενημερώνει την κατάσταση του μοντέλου. Μπορεί επίσης να στέλνει εντολές ώστε να γίνει η αντίστοιχη αναπαράσταση των δεδομένων του μοντέλου μέσω του *View*.
- Το **model** ενημερώνει τις αντίστοιχες αναπαραστάσεις *views* και τους *controllers* όταν υπάρχει αλλαγή στα δεδομένα. Αυτή η ενημέρωση επιτρέπει στα *views* να ενημερώνουν την γραφική απεικόνιση.
- Το **view** αναπαριστά με γραφικό τρόπο την πληροφορία που περιέχει το *model* δημιουργώντας γραφική παρουσίαση στο χρήστη.»

Μπορούμε να παρατηρήσουμε τον διαχωρισμό αυτόν στο τρόπο που είναι δομημένα τα project των εφαρμογών.

Δομή εφαρμογής Android

Controler	24/2/2016 4:20 μμ	File folder	
Model	24/2/2016 4:20 μμ	File folder	
parse	24/2/2016 4:20 μμ	File folder	
utilities	24/2/2016 4:20 μμ	File folder	
CatFirstListActivity.java	7/6/2015 11:09 μμ	Visual Studio Code	19 KB
CatSecondListActivity.java	7/6/2015 11:33 μμ	Visual Studio Code	23 KB
ContentActivity.java	23/8/2015 12:32 πμ	Visual Studio Code	15 KB
GetSpinnerDataParser.java	16/9/2014 2:17 μμ	Visual Studio Code	3 KB
JSONParser.java	16/9/2014 1:18 μμ	Visual Studio Code	3 KB
MainActivity.java	22/3/2015 6:37 μμ	Visual Studio Code	4 KB
MapActivity.java	22/3/2015 6:37 μμ	Visual Studio Code	20 KB
PreferencesActivity.java	22/3/2015 6:37 μμ	Visual Studio Code	2 KB
SearchResultsActivity.java	10/5/2015 2:44 μμ	Visual Studio Code	6 KB
SpinnerCategory.java	16/9/2014 2:17 μμ	Visual Studio Code	1 KB

Δομή Web εφαρμογής .NET5 / Angular



«Βασικά Πλεονεκτήματα MVC.

Χτίζοντας μία εφαρμογή με MVC έχουμε τα εξής βασικά πλεονεκτήματα:

Διαχωρισμός Προβλημάτων (Separation of Concerns).

Αυτό είναι και το πιο βασικό πλεονέκτημα του MVC. Ουσιαστικά δημιουργείται μία εφαρμογή η οποία έχει τρία επίπεδα, το επίπεδο των *models*, το επίπεδο των *controllers* και το επίπεδο των *views* -που θα αναλυθούν παρακάτω- και το κάθε επίπεδο επιτελεί ξεχωριστό έργο και ταυτόχρονα συνεργάζεται με τα άλλα επίπεδα. Μία σωστή MVC εφαρμογή είναι εκείνη που τα

τρία επίπεδα είναι ξεκάθαρα καθορισμένα και δεν συμπλέκονται. Για παράδειγμα είναι λάθος στο επίπεδο των *View* να υπάρχει κώδικας που «μιλάει» με την βάση δεδομένων και «τραβάει» δεδομένα.

Επεκτασιμότητα.

Το δεύτερο πλεονέκτημα της *MVC* αρχιτεκτονικής είναι πολύ σημαντικό επίσης. «Επεκτασιμότητα» είναι η δυνατότητα που διαθέτει μία εφαρμογή, κατά την οποία μπορούμε μελλοντικά να προσθέσουμε λειτουργίες σε αυτή ή να αλλάξουμε κάποιες από τις ήδη υπάρχουσες λειτουργίες και να έχουμε άλλα αποτελέσματα. Για να το δούμε εντελώς απλά και κατανοητά, η πλατφόρμα *WordPress* είναι επεκτάσιμη με τη χρήση των διάφορων *plugins* διότι προσθέτουμε στις ήδη υπάρχουσες λειτουργίες και άλλες λειτουργίες. Τα προγράμματα που είναι φτιαγμένα με *MVC* αρχιτεκτονική έχουν βασικό χαρακτηριστικό ότι είναι επεκτάσιμα.

Ελεγχιμότητα (Testability)

Αυτό είναι ένα πολύ κρίσιμο χαρακτηριστικό. Οι *MVC* εφαρμογές έχουν την δυνατότητα να είναι ελέγξιμες και με τον τρόπο αυτό συντηρούνται πιο εύκολα. Ας κάνουμε ένα απλό παράδειγμα. Έστω ότι έχουμε μία εφαρμογή η οποία διαθέτει μία λειτουργία *login*, δηλαδή ζητά από τον χρήστη να πληκτρολογήσει κάποια στοιχεία σε μία φόρμα και εν συνέχεια τον εισάγει μέσα στο σύστημα. Αυτή τη λειτουργία την ελέγχει κάποιος *loginController* ο οποίος περιέχει κώδικα που διαχειρίζεται τα δεδομένα αυτά που εισήχθησαν από τον χρήστη. Αυτός ο *controller* θεωρείται μία «μονάδα» ή αλλιώς *unit*. Στα *MVC frameworks* μπορούμε με πολλή ευκολία να γράψουμε απλό κώδικα *-tests* με τον οποίο τεστάρουμε αυτόν τον *controller* αλλά και κάθε μία από τις λειτουργίες του. Παίρνουμε τα αποτελέσματα και βλέπουμε η συγκεκριμένη μονάδα της εφαρμογής μας λειτουργεί σωστά.

«Καθαρά» URLs.

Τα περισσότερα *MVC frameworks* για *web applications* δίνουν τη δυνατότητα να έχουμε «καθαρά» *urls*. Ας κάνουμε ένα παράδειγμα. Έστω ότι έχουμε ένα *blog* και πατάμε το *link* για να διαβάσουμε ένα άρθρο. Ένα τυπικό *URL* θα μπορούσε να ήταν

http://example.com/article/Page.aspx?action=show&art_id=236.

http://example.com/article/Page.aspx?action=show&art_id=236.

Με το MVC μπορούσαμε να έχουμε το εξής

http://example.com/articles/nice-link

http://example.com/articles/nice-link

Βλέπουμε λοιπόν ότι με το MVC μπορούμε να έχουμε πιο όμορφα και εύληπτα από το χρήστη αλλά και την μηχανή αναζήτησης URLs.

Για να δημιουργήσουμε μία εφαρμογή σε MVC μπορούμε να χρησιμοποιήσουμε κάποιο framework, ανάλογα τη γλώσσα προγραμματισμού που θα χρησιμοποιήσουμε, το οποίο framework κάνει τη διαδικασία δημιουργίας του application πιο γρήγορη, πιο ασφαλή-διότι σε ένα framework υπάρχουν πολλές λειτουργίες by default, δεν χρειάζεται να γράψουμε κώδικα- και πιο ευχάριστη

[5]»

Η web εφαρμογή έγινε χρησιμοποιώντας το νέο MVC 6 framework της Microsoft. Το MVC6 είναι κομμάτι του .NET 5 framework και αποτελεί το πιο σύγχρονο τρόπο που προσφέρει το .NET για δημιουργία ενός web application που ακολουθεί πιστά το MVC pattern.

Web Services

Τα web services είναι μια τεχνολογία που επιτρέπει στις εφαρμογές να επικοινωνούν μεταξύ τους ανεξαρτήτως πλατφόρμας και γλώσσας προγραμματισμού. Ένα web service είναι μια διεπαφή λογισμικού (software interface) που περιγράφει μια συλλογή από λειτουργίες οι οποίες μπορούν να προσεγγιστούν από το δίκτυο μέσω πρότυπων μηνυμάτων XML. Χρησιμοποιεί πρότυπα βασισμένα στη γλώσσα XML για να περιγράψει μία λειτουργία (operation) προς εκτέλεση και τα δεδομένα προς ανταλλαγή με κάποια άλλη εφαρμογή. Μια ομάδα από web services οι οποίες αλληλεπιδρούν μεταξύ τους καθορίζει μια εφαρμογή web services. [6]

Η Microsoft μέσα από το MSDN της καταλήγει ότι όλα τα web services έχουν τρία (3) κοινά χαρακτηριστικά [7]:

- *Τα web services εκθέτουν χρήσιμη λειτουργικότητα σε χρήστες του διαδικτύου μέσα από ένα πρότυπο δικτυακό πρωτόκολλο. Στις περισσότερες περιπτώσεις αυτό το πρωτόκολλο είναι το SOAP (Simple Object Access Protocol).*
- *Τα web services παρέχουν ένα τρόπο να περιγράψουν τις διεπαφές τους με αρκετή λεπτομέρεια ώστε να επιτρέψουν στο χρήστη τους να χτίσει μια εφαρμογή πελάτη η οποία να επικοινωνήσει μαζί τους. Η περιγραφή συνήθως παρέχεται σε ένα έγγραφο XML το οποίο ονομάζεται έγγραφο WSDL (Web Services Description Language).*
- *Τα web services καταχωρούνται ώστε οι δυνητικοί χρήστες να μπορούν να τα βρουν εύκολα. Αυτό γίνεται με το UDDI (Universal Discovery Description and Integration).*

Τα web services λοιπόν αποτελούν μία αρχιτεκτονική κατανεμημένων συστημάτων κατασκευασμένη από πολλά διαφορετικά υπολογιστικά συστήματα τα οποία επικοινωνούν μέσω του δικτύου ώστε να δημιουργήσουν ένα σύστημα. Αποτελούνται από ένα σύνολο από πρότυπα τα οποία επιστρέφουν στους υπεύθυνους για την ανάπτυξη (προγραμματιστές - developers) να υλοποιήσουν κατανεμημένες εφαρμογές (χρησιμοποιώντας διαφορετικά εργαλεία από διαφορετικούς προμηθευτές) ώστε να κατασκευάσουν εφαρμογές που χρησιμοποιούν ένα συνδυασμό από ενότητες λογισμικού (software modules) οι οποίες καλούνται από συστήματα που ανήκουν σε διαφορετικά τμήματα ενός οργανισμού ή σε διαφορετικούς οργανισμούς [8].

JSON format

Η εξέλιξη της τεχνολογίας στις mobile συσκευές έκανε αναγκαία την εξεύρεση νέου format ώστε τα δεδομένα των web services να γίνουν πιο ελαφριά σε όγκο δεδομένων αποφεύγοντας την φλυαρία του XML φορμάτ. Λόγο αυτής της ανάγκης το JSON (JavaScript Object Notation) format επικράτησε έναντι του XML στην δημιουργία web services που απευθύνονται σε mobile εφαρμογές .

Το JSON όπως και το XML δεν είναι γλώσσες προγραμματισμού αλλά μια σύνταξη για αποθήκευση και ανταλλαγή δεδομένων. Όπως αναφέρθηκα το κύριο πλεονέκτημα του JSON format έναντι του XML είναι ο ογκος που χρειάζεται για να περιγράψει τα δεδομένα.

Για παράδειγμα η περιγραφή ενός array με τα ίδια δεδομένα υπαλλήλων γίνεται στα 2 format ως εξής

JSON

```
{ "employees": [
  { "firstName": "John",
    "lastName": "Doe" },
  { "firstName": "Anna",
    "lastName": "Smith" },
  { "firstName": "Peter",
    "lastName": "Jones" }
]}
```

XML

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

Όπως βλέπουμε το JSON format μπορεί να περιγράψει την ίδια πληροφορία με λιγότερους χαρακτήρες . Αυτό σε μεγάλο όγκο δεδομένων αποτελεί μεγάλο πλεονέκτημα έναντι του XML format και τον κύριο λόγο που το JSON επικράτησε στην νέα εποχή των mobile web services.

Με δεδομένα τα παραπάνω η ανάπτυξη της εφαρμογής έγινε χρησιμοποιώντας JSON format στα web services.

Ας αναλύσουμε το κάθε μέρος ξεχωριστά.

Μέρος 1ο : Web Services , Database Server και administration website

Εισαγωγή

Η γλώσσα υλοποίησης του πρώτου μέρους της εφαρμογής είναι η PHP με βάση δεδομένων την MySQL στο backend και Bootstrap framework(html/ css/ js) στο frontend . Η εφαρμογή τρέχει σε apache web server με λειτουργικό σύστημα linux.

Οι απαιτήσεις του πρώτου μέρους της εφαρμογής είναι η δημιουργία των web services της εφαρμογής όπως επίσης και ενός εύχρηστου περιβάλλοντος μέσα στο οποίο θα γίνεται η εισαγωγή / επεξεργασία των δεδομένων.

Η ανάπτυξη των web services και web εφαρμογών τα τελευταία χρόνια έχει φέρει στο προσκήνιο πολλές γλώσσες, frameworks και εργαλεία. Οι απαιτήσεις του πρώτου μέρους λοιπόν θα μπορούσαν να καλυφθούν με πολλές και διαφορετικές επιλογές.

Δεδομένου της απλότητας των web service που έχει ανάγκη η εφαρμογή και του μικρού σχετικά όγκου που θα χρειαστεί σε γραμμές κώδικα κρίθηκε ότι μια παραδοσιακή επιλογή όπως αυτή του LAMP stack (Linux/Apache/MySQL/PHP) είναι η καλύτερη δυνατή.

Αυτό γιατί η επιλογή αυτή κάλυπτε όλες τις βασικές απαιτήσεις που κρίθηκε ότι έπρεπε να υπάρχουν όπως

- Οικειότητα με την γλώσσα και τα εργαλεία της
- Ωριμότητα της γλώσσας
- Open Source εργαλεία
- Εύκολο deployment

Ο συνδυασμός Linux / Apache / MySQL / PHP είναι η πιο δημοφιλής πλατφόρμα ανάπτυξης web εφαρμογών με μεγάλη ωριμότητα , απλότητα και ταχύτητα στην ανάπτυξη, εύκολο deployment της εφαρμογής και ισχυρό community που υποστηρίζει το συγκεκριμένο stack τεχνολογιών εδώ και πολλά χρόνια .

Πάρα την ύπαρξη πιο μοντέρνων γλωσσών/frameworks και εργαλείων για web εφαρμογές και web services όπως python/django, javascript/node js, c#.NET , java/JAVA EE αλλά και scalable NoSQL βάσεις δεδομένων όπως mongo db που γίνονται όλο και πιο δημοφιλείς το LAMP stack αποτελεί το πιο πολυχρησιμοποιημένο stack ανάπτυξης στο web και πολύ οικία επιλογή στο μεγαλύτερο ποσοστό των developers.

CMS πλατφόρμες όπως το wordpress και web εφαρμογών όπως το facebook διατηρούν την ρηρ στην πρώτη γραμμή της ανάπτυξης web εφαρμογών ακόμη και σήμερα και αναδεικνύουν τις δυνατότητες της ακόμη και σε enterprise επιπέδου εφαρμογές.

Όσον αφορά το frontend κομμάτι της εφαρμογής η επιλογή ήταν πιο εύκολη υπόθεση. Οι απαιτήσεις της εφαρμογής είναι η ύπαρξη ενός εύχρηστου user interface όπου θα γίνεται η διαχείριση των δεδομένων της εφαρμογής. Η κατεξοχήν επιλογή γλωσσών για όμορφο και εύχρηστο web UI (user interface) είναι ο συνδυασμός HTML/CSS/JS .

Οι απαιτήσεις ήταν να έχουμε ένα εύχρηστο framework ανάπτυξης user interface με ui controls όπως κουμπιά , πίνακες ,layout. Η επιλογή που έγινε λοιπόν ήταν το bootstrap framework .

«Το Bootstrap είναι μια συλλογή εργαλείων ανοιχτού κώδικα ([Ελεύθερο λογισμικό](#)) για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Περιέχει [HTML](#) και [CSS](#) για τις μορφές τυπογραφίας, κουμπιά πλοήγησης και άλλων στοιχείων του περιβάλλοντος, καθώς και προαιρετικές επεκτάσεις [JavaScript](#). Είναι το πιο δημοφιλές πρόγραμμα στο GitHub» <https://el.wikipedia.org/wiki/Bootstrap>

Παρά την ύπαρξη πολλών HTML/CSS/JS frameworks το bootstrap έχει επικρατήσει και αποτελεί την πιο δημοφιλή open source και δωρεάν επιλογή. Προσφέρει πολλές δυνατότητες , πλούσιο documentation και παραδείγματα.

Ακολουθούν ενδεικτικά δύο εικόνες από το περιβάλλον της σελίδα διαχείρισης.

Σελίδα διαχείρισης- Διαχείριση κατηγοριών

Όνομα	Επεξεργασία
Αλλεργίες	Αλλαγή Διαγραφή
Ειχθηρήσεις	Αλλαγή Διαγραφή
Επείγοντα	Αλλαγή Διαγραφή
Λοιμώξεις	Αλλαγή Διαγραφή
Παιδικές ασθένειες	Αλλαγή Διαγραφή
Σπάνιες ασθένειες	Αλλαγή Διαγραφή
Συμπτώματα στα παιδιά	Αλλαγή Διαγραφή
Συχνές ασθένειες	Αλλαγή Διαγραφή
Χρόνιες ασθένειες	Αλλαγή Διαγραφή

Σελίδα διαχείρισης- Προσθήκη σημείου ενδιαφέροντος

The screenshot shows a web browser window with the URL `apokapa.com/public/sickid/create_poi.php`. The page title is "ΕΙΣΑΓΩΓΗ ΝΕΟΥ ΣΗΜΕΙΟΥ". A map of Piraeus is displayed with a red pin at the location of Grigoriou Iampraki 80. A form is overlaid on the map with the following fields:

- ΟΝΟΜΑ :
- ΤΗΛΕΦΩΝΟ :
- ΠΕΡΙΓΡΑΦΗ :
- ΑΠΟΣΤΗΚΕΥΤΗ :

Below the map, there are instructions in Greek: "Επιλέξτε κατηγορία και εισάγετε διεύθυνση με τη μορφή 'ΟΔΟΣ ΑΡΙΘΜΟΣ ΠΟΛΗ'". Below this, the form shows:

- ΚΑΤΗΓΟΡΙΑ : ΠΑΙΔΙΑΤΡΟΙ
- ΔΙΕΥΘΥΝΣΗ : Grigoriou Iampraki 80 PIRAEUS

At the bottom, there are two buttons: "ΕΠΙΒΕΒΑΙΩΣΕ ΔΙΕΥΘΥΝΣΗ ΣΤΟΝ ΧΑΡΤΗ" and "Πίσω".

MySQL Database

Η **MySQL** είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Για την αποθήκευση των δεδομένων η εφαρμογή χρησιμοποιεί μια MySQL βάση δεδομένων με τους παρακάτω πίνακες.

Table	Action	Rows	Type	Collation	Size	Overhead
admins	Browse Structure Search Insert Empty Drop	4	MyISAM	latin1_swedish_ci	2.3 KiB	-
astheneies	Browse Structure Search Insert Empty Drop	9	MyISAM	utf8_unicode_ci	2.4 KiB	848
astheneies_expand	Browse Structure Search Insert Empty Drop	34	MyISAM	utf8_unicode_ci	284.1 KiB	-
markers	Browse Structure Search Insert Empty Drop	13	MyISAM	utf8_general_ci	3.3 KiB	-
pois_cat	Browse Structure Search Insert Empty Drop	2	MyISAM	utf8_unicode_ci	2.1 KiB	-
version	Browse Structure Search Insert Empty Drop	2	MyISAM	utf8_unicode_ci	2.1 KiB	-
6 tables	Sum	64	InnoDB	utf8_unicode_ci	296.1 KiB	84 B

Πίνακας admins – Διαχειριστές εφαρμογής

Id (Primary key)	int(11)	Αύξων αριθμός διαχειριστή
username	varchar(50)	Όνομα διαχειριστή
Hashed_password	varchar(60)	Password διαχειριστή

Στον πίνακα admins φυλάγονται οι διαχειριστές που έχουν πρόσβαση στην σελίδα διαχείρισης της εφαρμογής και είναι υπεύθυνοι για την ανανέωση του περιεχομένου.

Πίνακας astheneies – Κατηγορίες ασθενειών

Aid (Primary key)	int(11)	Αύξων αριθμός κατηγορίας
title	varchar(50)	Τίτλος κατηγορίας

Στον πίνακα **astheneies** φυλάγονται οι κατηγορίες των ασθενειών της εφαρμογής.

Πίνακας astheneies_expand - Ασθένειες

Aid_exp (Primary key)	int(11)	Αύξων αριθμός ασθένειες
Aid (Foreign key)	int(11)	Αύξων αριθμός κατηγορίας
Title_exp	varchar(50)	Τίτλος ασθένειας
description	mediumtext	Περιγραφή ασθένειας
descrver	int(11)	Έκδοση περιγραφής

Στον πίνακα **astheneies_expand** φυλάγονται τα δεδομένα των ασθενειών της εφαρμογής ανα αύξων αριθμό κατηγορίας . Το πεδίο **descrver** ορίζει την έκδοση του περιεχομένου ώστε να γίνεται έλεγχος από την τοπική βάση της εφαρμογής android αν αυτό έχει αλλάξει ώστε να το ανανεώσει. Έτσι αποφεύγεται η σπατάλη δεδομένων για να κληθεί στην εφαρμογή η ίδια πληροφορία.

Πίνακας pois_cat – Κατηγορίες σημείων ενδιαφέροντος.

Cat_id (Primary key)	int(11)	Αύξων αριθμός κατηγορίας
name	varchar(50)	Όνομα κατηγορίας

Στο πίνακα pois_cat φυλάγονται οι κατηγορίες των σημείων ενδιαφέροντος της εφαρμογής όπως Παιδίατροι , Φαρμακεία και οποιαδήποτε μελλοντική προσθήκη.

Πίνακας markers – Σημεία ενδιαφέροντος

Id (Primary key)	int(11)	Αύξων αριθμός σημείου
Name	varchar(50)	Όνομα
Address	varchar(50)	Διεύθυνση
Lat	float(10,6)	Latitude
Lng	float(10,6)	Longitude
Type (foreing key)	varchar(50)	Κατηγορία σημείου
Tel	varchar(50)	Τηλέφωνο
description	varchar(200)	Περιγραφή

Στον πίνακα markers φυλάγονται οι πληροφορίες των σημείων ενδιαφέροντος της εφαρμογής όπως παιδίατροι και φαρμακεία ανά κατηγορία (type) σημείου.

Πίνακας version – Εκδόσεις πινάκων

id	int(11)	Αύξων αριθμός
Table_name	varchar(50)	Όνομα πίνακα
version	int(11)	Έκδοση πίνακα

Ο πίνακας version είναι βοηθητικός πίνακας που χρησιμοποιείτε από την εφαρμογή android ώστε να πληροφορηθεί αν η offline τοπική βάση της εφαρμογής συμβαδίζει με την live βάση. Έτσι αν δεν έχει συμβεί κάποια αλλαγή στους πίνακες όπως να προστεθεί κατηγορία η κάποια ασθένεια η android εφαρμογή δεν ξανακαλεί τα δεδομένα της MySql βάσης αλλά χρησιμοποιεί τα δεδομένα της τοπικής βάσης.

PHP Services

Τα δεδομένα της mobile και web εφαρμογής καλούνται μέσω php services τα οποία επιστρέφουν τα δεδομένα από την βάση προς τον client που τα καλεί σε μορφή JSON.

Κλήση κατηγοριών ασθενειών : get_all_astheneies.php

```
<?php

/*
 * Following code will list all astheneies
 */

// array for JSON response
$response = array();

// include db connect class
require_once __DIR__ . '/db_connect.php';

// connecting to db
$db = new DB_CONNECT();

// get all astheneies from astheneies table
$result = mysql_query("SELECT *FROM astheneies") or
die(mysql_error());

// check for empty result
if (mysql_num_rows($result) > 0) {
    // looping through all results
    // astheneies node
    $response["astheneies"] = array();

    while ($row = mysql_fetch_array($result)) {
        // temp user array
        $astheneies = array();
        $astheneies["aid"] = $row["aid"];
        $astheneies["title"] = $row["title"];

        // push single astheneies into final response array
        array_push($response["astheneies"], $astheneies);
    }
    // success
    $response["success"] = 1;

    // echoing JSON response
    echo json_encode($response);
} else {
    // no astheneies found
    $response["success"] = 0;
    $response["message"] = "No astheneies found";
}
```

```
// echo no users JSON
echo json_encode($response);
}
?>
```

Κλήση ασθενειών ανά κατηγορία : `get_all_astheneies_expand2.php`

(ανά aid κατηγορίας) `?aid=+aid`

```
<?php

/*
 * Following code will get astheneies_expand by category
 */

// array for JSON response
$response = array();

// include db connect class
require_once __DIR__ . '/db_connect.php';

// connecting to db
$db = new DB_CONNECT();

// check for post data
if (isset($_GET["aid"])) {
    $aid = $_GET['aid'];

    // get a astheneia from astheneies table

    $result = mysql_query("SELECT * FROM astheneies_expand WHERE
aid=$aid");

    if (!empty($result)) {
        // check for empty result
        if (mysql_num_rows($result) > 0) {
            // looping through all results
            // astheneies node
            $response["astheneies_expand"] = array();

            while ($row = mysql_fetch_array($result)) {
                // temp user array
                $astheneies_expand = array();
                $astheneies_expand["aid_exp"] = $row["aid_exp"];
                $astheneies_expand["title_exp"] = $row["title_exp"];
            }
        }
    }
}
```



```
        // push single astheneies into final response array
        array_push($response["astheneies_expand"],
$astheneies_expand);
    }
    // success
    $response["success"] = 1;

    // echoing JSON response
    echo json_encode($response);

}

else {
    // no astheneia found
    $response["success"] = 0;
    $response["message"] = "No astheneia found";

    // echo no users JSON
    echo json_encode($response);
}
} else {
    // no astheneia found
    $response["success"] = 0;
    $response["message"] = "No astheneia found";

    // echo no users JSON
    echo json_encode($response);
}
} else {
    // required field is missing
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";

    // echoing JSON response
    echo json_encode($response);
}
?>
```

Κλήση δεδομένων ασθένειας ανά αύξων αριθμό ασθένειας : get_astheneies_detail.php

(ανά A/A ασθένειας) ?aid_exp= aid_exp

```
<?php

/*
 * Following code will get single astheneia_expand details
 */

// array for JSON response
$response = array();

// include db connect class
require_once __DIR__ . '/db_connect.php';

// connecting to db
$db = new DB_CONNECT();

// check for post data
if (isset($_GET["aid_exp"])) {
    $aid_exp = $_GET['aid_exp'];

    // get astheneia_expand
    $result = mysql_query("SELECT title_exp,description FROM
    astheneies_expand WHERE aid_exp = $aid_exp");

    if (!empty($result)) {
        // check for empty result
        if (mysql_num_rows($result) > 0) {

            $result = mysql_fetch_array($result);

            $astheneia = array();
            $astheneia["title_exp"] = $result["title_exp"];
            $astheneia["description"] = $result["description"];

            // success
            $response["success"] = 1;

            // user node
            $response["astheneia"] = array();

            array_push($response["astheneia"], $astheneia);

            // echoing JSON response
            echo json_encode($response);
        } else {
            // no astheneia found
            $response["success"] = 0;
            $response["message"] = "No astheneia found";
        }
    }
}
```

```

        // echo no users JSON
        echo json_encode($response);
    }
} else {
    // no astheneia found
    $response["success"] = 0;
    $response["message"] = "No astheneia found";

    // echo no users JSON
    echo json_encode($response);
}
} else {
    // required field is missing
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";

    // echoing JSON response
    echo json_encode($response);
}
?>

```

Μέρος 2ο : Android Application

Το δεύτερο μέρος της υλοποίησης αποτελεί τη δημιουργία μιας mobile εφαρμογής η οποία θα καταναλώνει τα web services προσφέροντας στον τελικό χρήστη χρήσιμες πληροφορίες όπως αυτές αναλύονται στην εισαγωγή.

Το κριτήριο επιλογής ήταν η προσπάθεια να καλύψουμε όσο μεγαλύτερο μέρος των χρηστών κινητής τηλεφωνίας γίνεται. Η επιλογή ήταν να φτιαχτεί μια native mobile εφαρμογή σε ένα από τα 3 δημοφιλή mobile λειτουργικά συστήματα iOS, Android, Windows . Με βάση τα παραπάνω η επιλογή του Android μοιάζει μονόδρομος αφού έχει κυριαρχήσει σε ποσοστό χρήσης έναντι των δυο άλλων επιλογών.

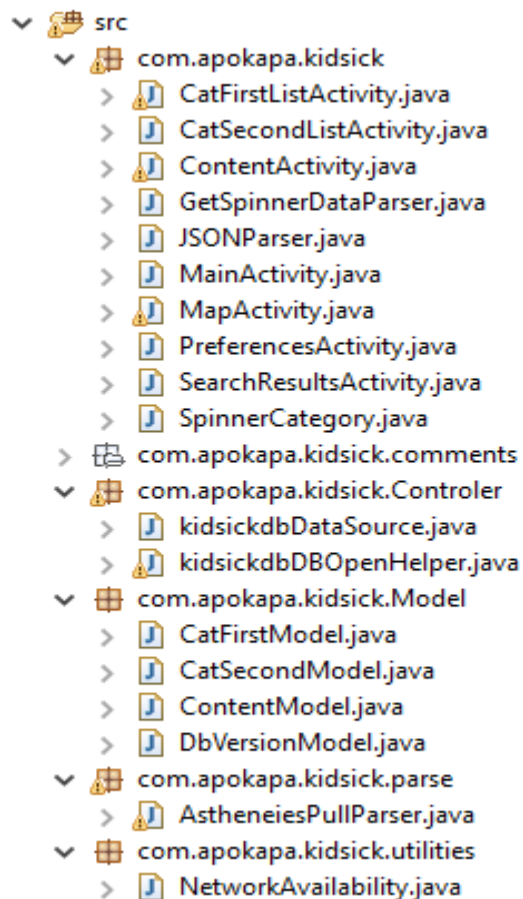
«Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Handset Alliance|Open Handset Alliance . Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Το Android είναι κατά κύριο λόγο σχεδιασμένο για συσκευές με οθόνη αφής, όπως τα έξυπνα τηλέφωνα και τα τάμπλετ, με διαφορετικό περιβάλλον χρήσης για τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear). Παρόλο που έχει αναπτυχθεί για συσκευές με οθόνη αφής, έχει χρησιμοποιηθεί σε κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, συνηθισμένους Η/Υ (π.χ. το HP Slate 21) και σε άλλες ηλεκτρονικές συσκευές.

Το Android είναι το πιο ευρέως διαδεδομένο λογισμικό στον κόσμο. Οι συσκευές με Android έχουν περισσότερες πωλήσεις από όλες τις συσκευές [Windows](#), [iOS](#) και [Mac OS X](#) μαζί. ^{[9][10]}»

Το κενό στις άλλες πλατφόρμες θα προσπαθήσουμε να καλύψουμε με AngularJS web application το οποίο αποτελεί το 3ο μέρος της υλοποίησης.

Η ανάπτυξη της εφαρμογής έγινε σε JAVA η οποία αποτελεί και την πιο δημοφιλή επιλογή για ανάπτυξη εφαρμογών android. Το δημοφιλές συνεπάγεται και μεγάλο community έτοιμο να παρέχει λύσεις, παραδείγματα και έτοιμα libraries ώστε να αναπτύξουμε την εφαρμογή μας με τις καλύτερες δυνατές προϋποθέσεις.

Δομή κώδικα εφαρμογής android



Η δομή της εφαρμογής android αποτελείτε από τα activities τους controller και τα μοντέλα της εφαρμογής

Τα activities αποτελούν τις δραστηριότητες της εφαρμογής. Κάθε activity σε μια android εφαρμογή ενσαρκώνει και μια λειτουργία της εφαρμογής. Το αρχείο MainActivity αποτελεί την αρχική εικόνα της εφαρμογής όταν ξεκινάει και περιέχει όλες τις επιλογές που οδηγούν στα υπόλοιπα activities της εφαρμογής.

```
public void gotoMap(View v)
{
    if(isNetworkAvailable())
    {
        try
        {
            Intent intent = new Intent(this, MapActivity.class);
            startActivity(intent);
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
    else
    {
        Toast.makeText(this, TAG_NOINTERNET,
Toast.LENGTH_LONG).show();
    }
}
```

Εδώ βλέπουμε χαρακτηριστικά το σημείο του κώδικα στο MainActivity που οδηγεί αν επιλεγεί το αντίστοιχο κουμπί που καλεί την gotoMap στο MapActivity της εφαρμογής.

Το επόμενο σημαντικό κομμάτι είναι τα μοντέλα της εφαρμογής. Τα μοντέλα αποτελούν κλάσεις που αντιπροσωπεύουν τις οντότητες της εφαρμογής όπως κατηγορίες ασθενειών , ασθένειες κτλ.

```
package com.apokapa.kidsick.Model;

import java.text.NumberFormat;

public class CatFirstModel {
    private long id;
    private int aid;
    private String title;

    public long getId() {
        return id;
    }

    public void setId(long id) {
        this.id = id;
    }

    public int getAid() {
        return aid;
    }

    public void setAid(int aid) {
        this.aid = aid;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    @Override
    public String toString() {

        NumberFormat nf = NumberFormat.getIntegerInstance();
        return title + "\n(" + nf.format(aid) + ")";
    }
}
```

Εδώ βλέπουμε την κλάση που περιγράφει το μοντέλο για τις κατηγορίες ασθενειών.

Offline λειτουργία και συγχρονισμός

Στην εφαρμογή android υπάρχει η δυνατότητα τα δεδομένα των ασθενειών στις οποίες πλοηγείτε ο χρήστης να αποθηκεύονται σε μια τοπική βάση δεδομένων sqlite ώστε να μπορεί να έχει πρόσβαση σε αυτά όταν είναι εκτός σύνδεσης και ταυτόχρονα να μην χρειαστεί να τα ανακτήσει ξανά τα δεδομένα αν αυτά δεν έχουν υποστεί αλλαγές από τον διαχειριστή της βάσης.

Το SQLite είναι μια δημοφιλής επιλογή ως ενσωματωμένη βάση δεδομένων για τοπική αποθήκευση ή αποθήκευση πελάτη σε λογισμικό εφαρμογής όπως ως φυλλομετρητής. Είναι η πιο πλατιά αναπτυσσόμενη μηχανή βάσης δεδομένων και χρησιμοποιείται σήμερα από πολλούς φυλλομετρητές ευρείας χρήσης, από λειτουργικά συστήματα και από ενσωματωμένα συστήματα, μεταξύ άλλων. Το SQLite έχει συνδέσεις με πολλές γλώσσες προγραμματισμού. [17]

Ο τρόπος με τον οποίο υλοποιείτε η λειτουργία αυτή είναι με την δημιουργία στην βάση του web server της εφαρμογής ενός πίνακα και πεδίων που ελέγχουν την έκδοση της βάσης. Έτσι αν κάτι προστεθεί για παράδειγμα στην λίστα με τις κατηγορίες ασθενειών τότε η εφαρμογή στο κινητό θα ξανακατεβάσει την λίστα αυτή διαφορετικά θα χρησιμοποιήσει την λίστα που έχει στην τοπική βάση από την προηγούμενη φορά που έγινε η ανάκτηση.

Με αυτό τον τρόπο η εφαρμογή δίνει την δυνατότητα στον χρήστη να μπορέσει να έχει πρόσβαση σε δεδομένα που έχει ανακτήσει χωρίς να είναι συνδεδεμένος στο διαδίκτυο μέσω της τοπικής βάσης. Ταυτόχρονα σε περίπτωση που ο χρήστη είναι συνδεδεμένος τότε η εφαρμογή δεν ανάκατά όλη την πληροφορία αλλά κάνει απλά έναν μικρό σε κόστος όγκου έλεγχο ώστε να ελέγξει αν έχει αλλάξει κάτι στην βάση. Έτσι αποφεύγουμε να κατεβάσουμε πολλές φορές μια λίστα ασθενειών πολλών kilobytes αν δεν έχει αλλάξει κάτι. Ο έλεγχος της έκδοσης κοστίζει μόνο την ανακτήσει ενός json string .

Τα βήματα που ακολουθεί η λειτουργία είναι τα εξής

ΑΝ ΥΠΑΡΧΕΙ ΣΥΝΔΕΣΗ

Υπάρχουν τα δεδομένα που ζήτησε ο χρήστης στη τοπική βάση?

ΑΝ ΝΑΙ : Έλεγε την έκδοση της τοπικής βάσης με την έκδοσή του web server. Αν δεν έχει αλλάξει κάτι δείξε τα δεδομένα της τοπικής βάσης διαφορετικά κατέβασε τα δεδομένα ξανά.

ΑΝ ΟΧΙ : Κατέβασε τα δεδομένα στη τοπική βάση και ανανέωσε την έκδοση της τοπικής βάσης.

ΑΝ ΔΕΝ ΥΠΑΡΧΕΙ ΣΥΝΔΕΣΗ

Υπάρχουν τα δεδομένα που ζήτησε ο χρήστης στη τοπική βάση?

ΑΝ ΝΑΙ : Δείξε τα δεδομένα της τοπικής βάσης.

ΑΝ ΟΧΙ : Δείξε μήνυμα ότι δεν υπάρχει διαθεσιμότητα

Ακολουθεί κομμάτι του κώδικα όπου ξεκινάνε οι έλεγχοι που περιγράφονται στα βήματα

```
//ON CREATE BUNDLE
@Override
public void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.second_activity);
    //go back menu option
    getActionBar().setDisplayHomeAsUpEnabled(true);
    //sqlite
    datasource = new kidsickdbDataSource(this);
    datasource.open();

    //datasource.deleteAll_DbVersion();
    //datasource.deleteAllAstheneies();

    List<DbVersionModel> dbversion_list=
datasource.findAll_DbVersion();

    if(dbversion_list.size()==0) //Ελέγχει αν υπάρχει
τοπικη βάση
    {
        if (isNetworkAvailable())//Αν δεν υπάρχει βάση Ελέγχει
αν υπάρχει δυκτιο ώστε να κατεβάσει
        {

            //get db version from server with async task
            db_version_mode="insert";
            new GetDbVersion().execute();

        }
        else
        {
            toastmessage(TAG_NOINTERNET);
        }
    }
    Else
    {

        if (isNetworkAvailable())//Αν υπάρχει τοπική βάση και
διαδίκτιο ελέγχει την έκδοση
        {

            db_version_mode="check";
            new GetDbVersion().execute();//Η function αυτή
ελέγχει αν η τοπικη βάση είναι στη τελευταία έκδοση και δείχνει τα
τοπικά δεδομένα η αλλιώς τα ανακτά ξανά.
```



```

Else //Αν δεν υπάρχει διαδίκτυο ελέγχει αν τα δεδομένα υπάρχουν στη
τοπική βάση διαφορετικά βγάζει μήνυμα μη διαθεσιμότητας
    {

        List<CatFirstModel> astheneies = datasource.findAll();
if (astheneies.size() >0)
        {
            //Initializing Hashmap for ListView
            astheneiesList = new ArrayList<HashMap<String,
String>>();

            List<CatFirstModel> astheneies_list =
datasource.findAll();
for (int i=0; i<astheneies_list.size(); i++)
            {

                Log.i(LOGTAG,"astheneies_list.size() "+i);
                CatFirstModel astheneia = new CatFirstModel();
                astheneia=astheneies_list.get(i);
                int aid_int = astheneia.getAid();
                String aid= Integer.toString(aid_int);
                String title=astheneia.getTitle();
                Log.i(LOGTAG,"astheneies_list.size()
"+title+aid);

                // creating new HashMap
                HashMap<String, String> map = new
HashMap<String, String>();
                // adding each child node to HashMap key =>
value

                map.put(TAG_AID, aid);
                map.put(TAG_TITLE, title);
                // adding HashList to ArrayList
                astheneiesList.add(map);

            }

            ListAdapter adapter = new SimpleAdapter(
                CatFirstListActivity.this, astheneiesList,
                R.layout.list_item, new String[] { TAG_AID,
                    TAG_TITLE},
                    new int[] { R.id.aid, R.id.title });
            // updating listview
            setListAdapter(adapter);
        }
else
        {
            toastmessage(TAG_NOINTERNET);
        }
    }
}

```

Μέρος 3^ο : Web Application

Το 3^ο μέρος της υλοποίησης αφορά την δημιουργία ενός website το οποίο δυνητικά αφενός θα είναι η ιστοσελίδα της mobile εφαρμογής(link για download) και αφετέρου ενσωματώνει μια AngularJS εφαρμογή η οποία καταναλώνει τα ίδια web services με την mobile εφαρμογή και θα παρέχει αντίστοιχη πληροφόρηση ανεξαρτήτου πλατφόρμας. Δηλαδή θα μπορεί να τρέξει από υπολογιστή αλλά και από οποιοδήποτε λειτουργικού συστήματος κινητό τηλέφωνο η tablet μέσα από web browser .

Η επιλογή της πλατφόρμας έγινε με κριτήριο να εξερευνηθούν οι δυνατότητες του καινούριου web framework της Microsoft .NET 5 (CORE) / MVC6 το οποίο κατά τη διάρκεια της πτυχιακής ήταν σε beta και RC έκδοση.

Το .NET CORE αποτελεί ότι νεότερο υπάρχει σε web application framework και αποτελεί την προσπάθεια της Microsoft να ευθυγραμμιστεί με όλα τα καινούρια εργαλεία και μεθοδολογίες ανάπτυξης web εφαρμογών. Αρχικά το .NET CORE είναι multiplatform δηλαδή ξεφεύγει από τα στενά όρια των windows και μπορεί να γίνει deploy ακόμα και σε Linux server όπως η rhr για παράδειγμα. Επίσης ενσωματώνει το WEB API του .NET πράγμα που σημαίνει ότι μπορούμε να χτίσουμε μαζί με το website και web services στο ίδιο project.

Το .NET CORE θα αποτελούσε ιδανική επιλογή και για την δημιουργία του web api της εφαρμογής ωστόσο δεν κρίθηκε αρκετά ώριμο ακόμα λόγω των αλλαγών που γίνονται συνεχώς αφού βρίσκεται ακόμα υπό ανάπτυξη τουλάχιστον την περίοδο που ξεκίνησε η υλοποίηση αυτής της εφαρμογής .

Κρίθηκε σκόπιμο σε αυτό το σημείο να ενσωματωθεί η Angular εφαρμογή μας μέσα σε ένα .NET CORE web project ώστε αργότερα να υπάρχει η δυνατότητα να χτίσουμε ένα καινούριο web api και να υπάρξει μια ομαλή μετάβαση του 1^{ου} μέρους της υλοποίησης από LAMP stack σε .NET CORE/SQL SERVER.

Η επιλογή έγινε ώστε μελλοντικά το .NET5 website να λειτουργήσει σαν ένα portal που θα μπορούσε να φιλοξενήσει και άλλες mobile εφαρμογές σε Angular JS. Στην αρχική του μορφή το website θα προσφέρει κάποιες πληροφορίες για την εφαρμογή , ενσωματώνει σύστημα εγγραφής χρηστών και παρέχει πρόσβαση στο Angular JS application που θα προσφέρει ανάλογη πληροφόρηση με την mobile εφαρμογή στο Android.

Το web application της εφαρμογής έχει φτιαχτεί στο σύγχρονο JavaScript framework Angular JS .

Η ανάγκη δημιουργίας web εφαρμογών με διαδραστικό user interface οδήγησε αρχικά στην χρήση τεχνολογιών όπως java ,flash και Silverlight τα οποία έτρεχαν με την βοήθεια plugins πάνω στον εκάστοτε browser . Η ανάγκη που οδήγησε σε αυτήν την εξέλιξη ήταν η δυνατότητα στις web σελίδες να συμπεριφέρονται περισσότερο σαν εφαρμογές και λιγότερο σαν στατικές σελίδες.

Η JavaScript αν και ξεκίνησε σαν μια απλή scripting language για απλές λειτουργίες στο web μετεξελίχθηκε με libraries όπως η δημοφιλή jQuery και τεχνικές όπως η AJAX στην κύρια

γλώσσα του web. Σύγχρονα JavaScript Frameworks όπως η Angular JS , React JS, Backbone JS, Knockout JS δίνουν δυνατότητες για δημιουργία διαδραστικών web εφαρμογών.

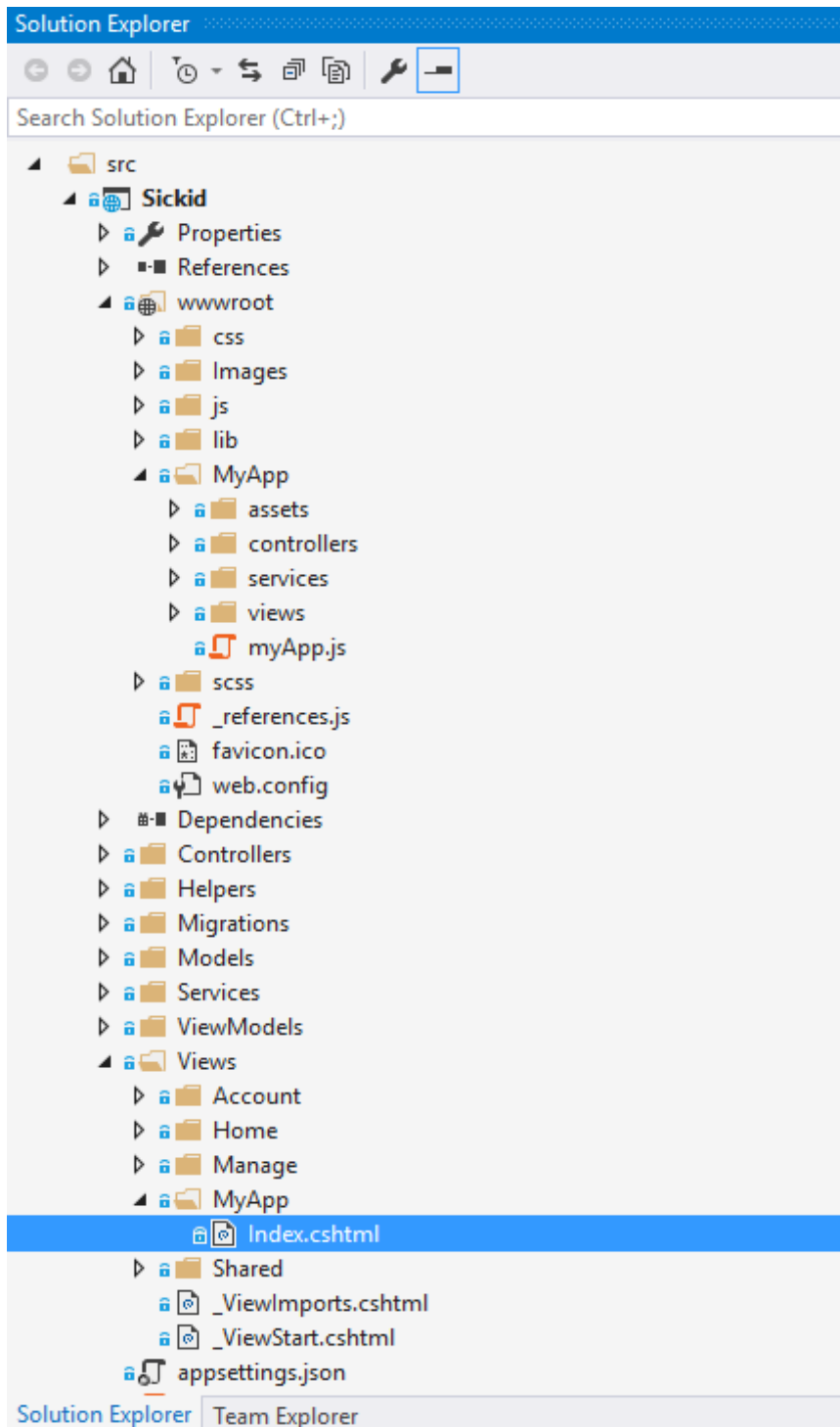
Η εξέλιξη της JavaScript έδωσε εξαιρετικά εργαλεία και μεθοδολογίες για την δημιουργία web εφαρμογών και αποτελεί πλέον μονόδρομο στην υλοποίηση εφαρμογών σε περιβάλλον browser. Οι δυνατότητες της JavaScript αποδείχθηκαν τόσο ισχυρές ώστε να παραγκωνίσουν και να οδηγήσουν σε «αργό θάνατο» όλα τα παλιά εργαλεία ανάπτυξης web εφαρμογών όπως flash και Silverlight.

Η επιλογή της Angular JS έγινε γιατί προσφέρει ένα ολοκληρωμένο MVC framework που προσφέρει δυνατότητες routing κάτι που λείπει από τα υπόλοιπα JavaScript frameworks τα οποία καλύπτουν αυτή την ανάγκη με εξωτερικά εργαλεία.

“Η angular είναι μια γλώσσα που δρα στον browser του υπολογιστή δηλαδή μια client-side γλώσσα. Για την ακρίβεια και για να είμαστε πιο σωστοί δεν είναι γλώσσα αλλά ένα mvc(η ακόμα καλύτερα ένα mww(model view whatever)) Framework το οποίο σου δίνει απεριόριστες δυνατότητες. Η angular είναι ένα framework που βασίζεται στην javascript και δημιουργεί ένα “αντικείμενο” σε όλη την σελίδα που θα υποδείξουμε. Αυτό το αντικείμενο που δημιουργείτε μπορούμε να το αλλάξουμε μέσα από τους controllers να το μορφοποιήσουμε με τα services προσθέτοντας data. Η κυρίως μεταβλητή της angular είναι η \$scope και εκεί που γίνεται όλη η “μαγεία” είναι ο controller. Το πιο σημαντικό προτέρημα της angular είναι η προσαρμοστικότητα της και η ευκολία εκμάθησής της. [11]

Η ανάπτυξη εφαρμογών σε JavaScript αποτελεί την ιδανική επιλογή για multiplatform εφαρμογές σε web, mobile, tablet ανεξάρτητου λειτουργικού συστήματος. Με την ανάπτυξη της εφαρμογής σε Angular JS προσπαθούμε να καλύψουμε το κενό που αφήνει η android εφαρμογή στα άλλα mobile λειτουργικά iOS και Windows. Φυσικά δεν έχουμε όλες τις δυνατότητες που μας δίνει η native android εφαρμογή αλλά υπάρχουν όλες οι προϋποθέσεις για ανάπτυξη όλων των δυνατοτήτων μελλοντικά.

Δομή web εφαρμογής



Η εφαρμογή είναι μια υβριδική υλοποίηση ενός dotnetcore web application μέσα από το οποία έχουμε πρόσβαση σε ένα portal το οποίο φιλοξενεί την angular js εφαρμογή. Όπως βλέπουμε η δομή της εφαρμογής ακολουθεί το πρότυπο MVC. Το αρχείο index.cshtml στον φάκελο MyApp είναι το endpoint της angular εφαρμογής η οποία βρίσκεται στον φάκελο MyApp στο wwwroot.

Όπως βλέπουμε στον κώδικα που ακολουθεί το αρχείο index.cshtml φορτώνει όλα τα απαραίτητα javascript scripts που απαιτούνται από την angular εφαρμογή και στο σημείο

```
<body ng-app="myApp" layout="row">
```

την καλεί να φορτώσει στην σελίδα.

```

@*
    For more information on enabling MVC for empty projects, visit
http://go.microsoft.com/fwlink/?LinkID=397860
*@
@{
    Layout = null;
}

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Angular Material - Starter App</title>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="description" content="">
    <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-
scalable=no" />

    <link rel='stylesheet'
href='<a href="http://fonts.googleapis.com/css?family=Roboto:400,500,700,400italic">http://fonts.googleapis.com/css?family=Roboto:400,500,700,400italic'>
    <link rel="stylesheet" href="~/lib/angular-material/angular-material.css"
/>
    <link rel="stylesheet" href="~/MyApp/assets/app.css" />
    <link rel="stylesheet" href="~/lib/font-awesome/css/font-awesome.min.css"
/>

    <style type="text/css">
/**
 * Hide when Angular is not yet loaded and initialized
 */
[ng\:cloak], [ng-cloak], [data-ng-cloak], [x-ng-cloak], .ng-cloak, .x-
ng-cloak {
    display: none !important;
}
</style>
</head>

<body ng-app="myApp" layout="row">

    <div ng-view layout="row" style="width:100%">
</div>

    <script src="~/lib/angular/angular.js"></script>
    <script src="~/lib/angular-route/angular-route.min.js"></script>
    <script src="~/lib/angular-animate/angular-animate.js"></script>
    <script src="~/lib/angular-aria/angular-aria.js"></script>
    <script type="text/javascript" src="~/lib/angular-material/angular-
material.js"></script>
    <script src="~/MyApp/myApp.js"></script>
    <script src="~/MyApp/services/sickid.js"></script>
    <script src="~/MyApp/controllers/astheneiesController.js"></script>
    <script
src="~/MyApp/controllers/astheneiesByCategoryController.js"></script>
</body>

</html>

```

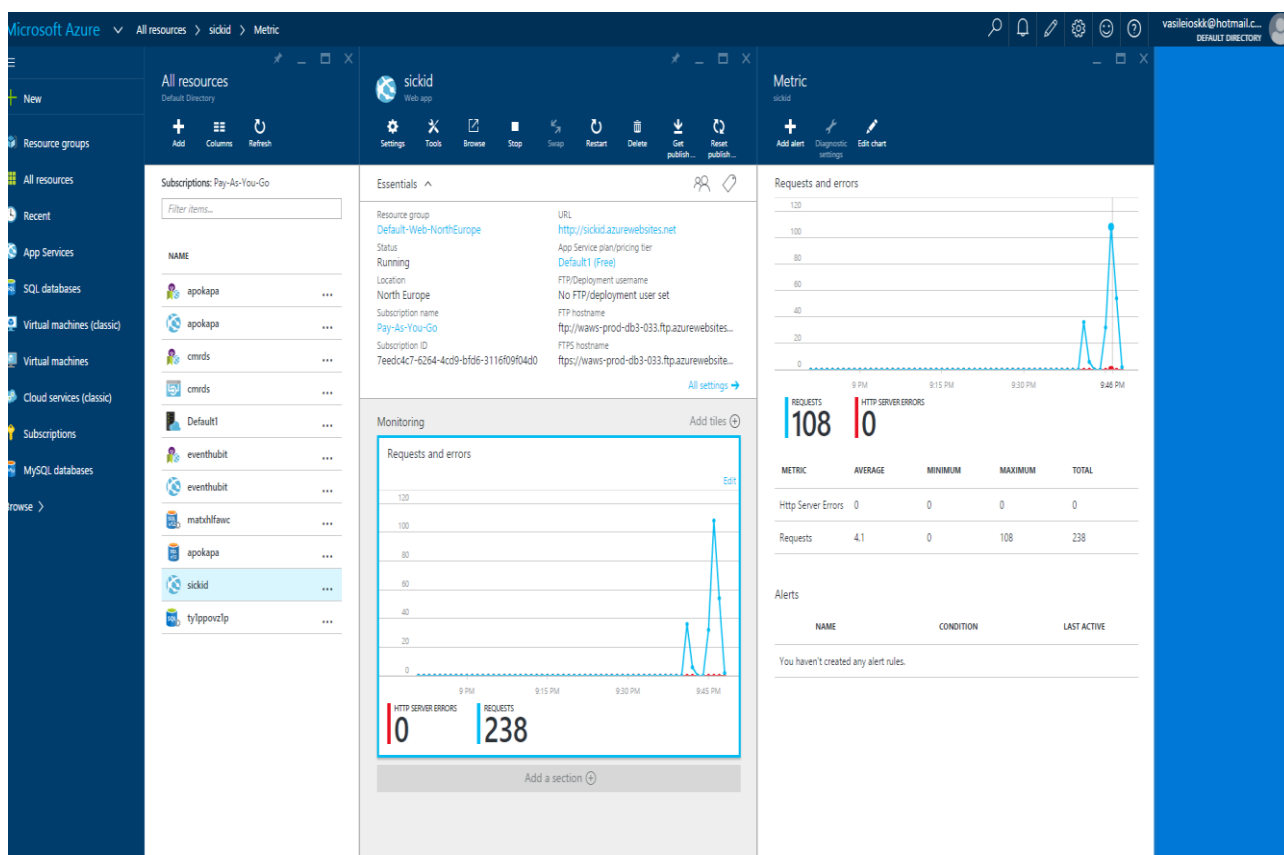
Εγκατάσταση web εφαρμογής

Σημαντικό ρόλο στην επιλογή της νέας πλατφόρμας της Microsoft dotnet core είναι η δυνατότά που προσφέρει για εγκατάσταση σε Linux server με την χρήση Docker container αλλά και στην cloud πλατφόρμα της Microsoft Azure. Το Docker (Ντόκερ) είναι μια πλατφόρμα λογισμικού ανοιχτού κώδικα που υλοποιεί [Εικονικοποίηση](#) (Virtualization) σε επίπεδο [Λειτουργικού Συστήματος](#). Ουσιαστικά το Docker προσφέρει αυτοματοποιημένες διαδικασίες για την ανάπτυξη εφαρμογών σε απομονωμένες Περιοχές Χρήστη (User Spaces) που ονομάζονται Software Containers. Το λογισμικό χρησιμοποιεί τεχνολογίες του πυρήνα του [Linux](#) όπως τα cgroups και οι χώροι ονομάτων πυρήνα (kernel namespaces), για να επιτρέπει σε ανεξάρτητα software containers να εκτελούνται στο ίδιο λειτουργικό σύστημα. Έτσι αποφεύγεται η χρήση επιπλέον υπολογιστικών πόρων που θα απαιτούσε μια [εικονική μηχανή](#) (virtual machine).^[16]

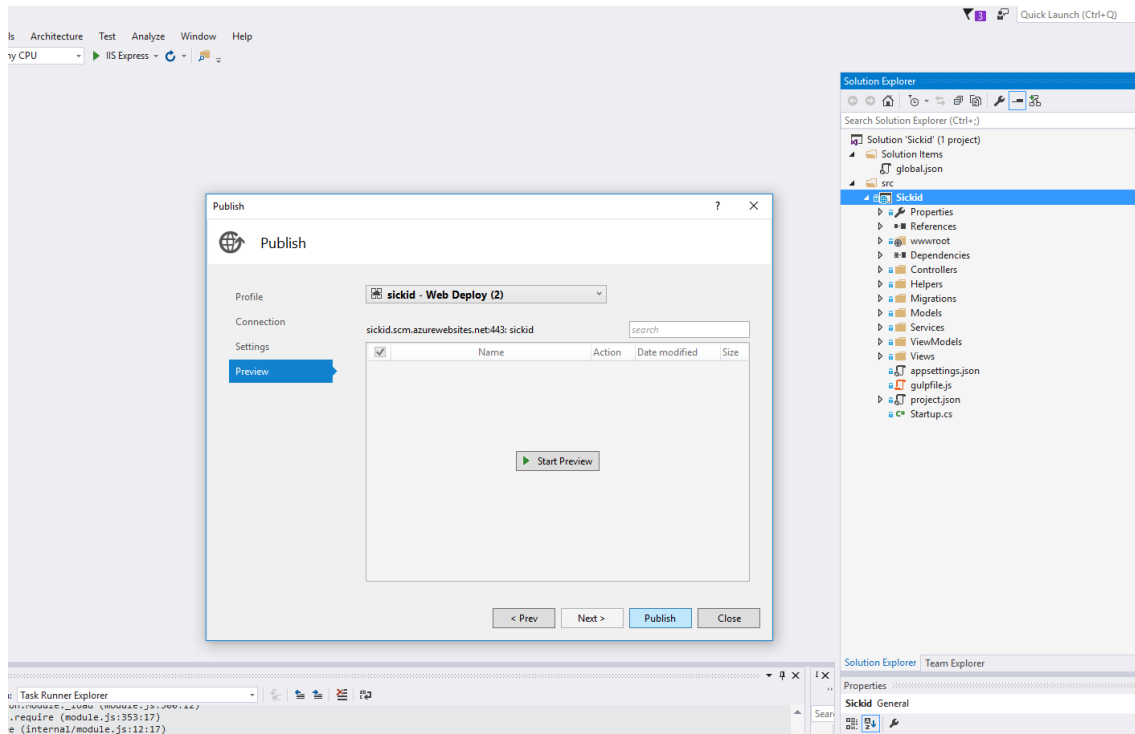
Azure Deployment

Αν και το docker είναι μια πολλά υποσχόμενη τεχνολογία για εγκατάσταση web εφαρμογών το Microsoft Azure προσφέρει μια πιο ευέλικτη πλατφόρμα cloud η οποία μας επιτρέπει να δημιουργούμε και φιλοξενούμε εφαρμογές χωρίς να επιβαρυνόμαστε με την συντήρηση της υποδομής του web application και της βάσης δεδομένων. Το Azure εξασφαλίζει υψηλή διαθεσιμότητα 99,9% και επιτρέπει να αναπτύσσουμε και να τρέχουμε εφαρμογές και υπηρεσίες με ασφάλεια χωρίς να μας απασχολεί η υποδομή.

Το Azure παρέχει μια ολοκληρωμένη πλατφόρμα διαχείρισης και ελέγχου της εφαρμογής ώστε να απαλλάξει τον προγραμματιστή από όλες τις εργασίες που απαιτούνται για το στήσιμο την συντήρηση της υποδομής.



Με την επιλογή publish στο visual studio επιλέγουμε azure deployment και εισάγουμε τα στοιχεία του azure λογαριασμού μας. Έτσι με μερικά μόλις βήματα έχουμε την εφαρμογή μας εγκατεστημένη και έτοιμη για λειτουργία.

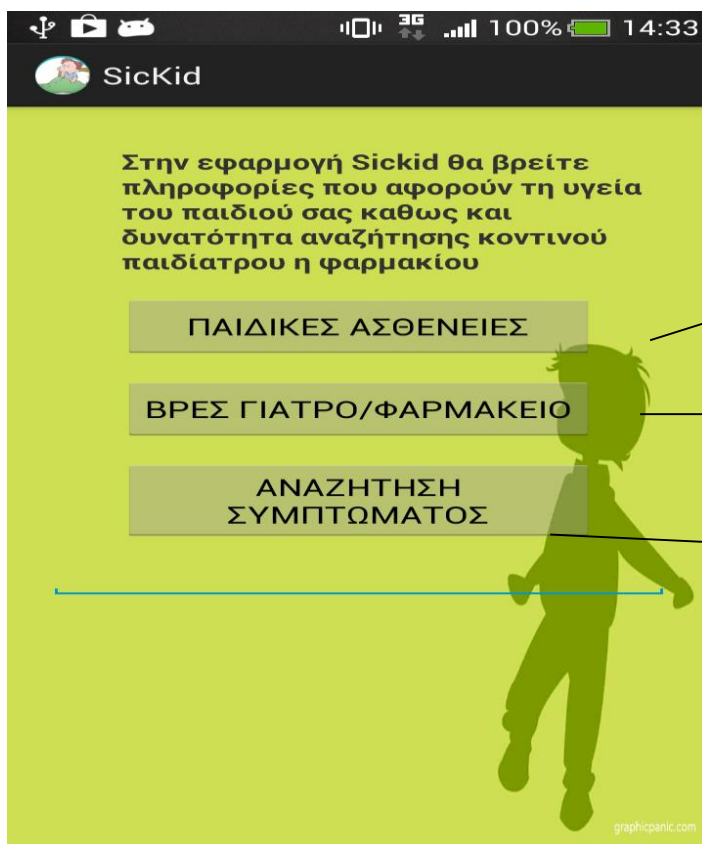


Κεφάλαιο 4: Παρουσίαση και χρήση εφαρμογής

Εφαρμογή Android

Στο εγχειρίδιο αυτό γίνεται μια σύντομη αναφορά στον τρόπο λειτουργίας της εφαρμογής, καθώς και στις εργασίες που μπορείτε να εκτελέσετε στο πλαίσιο αυτής.

Κεντρικό μενού επιλογών



ΕΠΙΛΟΓΗ ΓΙΑ ΝΑ ΔΕΙΤΕ ΤΙΣ ΚΑΤΗΓΟΡΙΕΣ ΠΑΙΔΙΚΩΝ ΑΣΘΕΝΕΙΩΝ ΠΟΥ ΥΠΑΡΧΟΥΝ ΣΤΗΝ ΕΦΑΡΜΟΓΗ

ΕΠΙΛΟΓΗ ΓΙΑ ΝΑ ΒΡΕΙΤΕ ΚΟΝΤΙΝΟ ΦΑΡΜΑΚΕΙΟ / ΠΑΙΔΙΑΤΡΟ ΕΦΑΡΜΟΖΟΝΤΑΣ ΦΙΛΤΡΟ ΑΠΟΣΤΑΣΗΣ

ΕΠΙΛΟΓΗ ΓΙΑ ΝΑ ΑΝΑΖΗΤΗΣΕΤΕ ΚΑΠΟΙΑ ΑΘΕΝΕΙΑ Η ΣΥΜΠΤΩΜΑ

Επιλογή εμφάνισης κατηγοριών



ΕΠΙΛΕΞΤΕ ΚΑΠΟΙΑ
ΚΑΤΗΓΟΡΙΑ ΓΙΑ ΝΑ
ΔΕΙΤΕ ΤΙΣ ΑΣΘΕΝΕΙΕΣ
ΠΟΥ ΥΠΑΡΧΟΥΝ ΣΕ
ΑΥΤΗ

Επιλογή ασθένειας



ΕΠΙΛΕΞΤΕ ΚΑΠΟΙΑ
ΑΣΘΕΝΕΙΑ ΓΙΑ ΝΑ
ΔΙΑΒΑΣΕΤΕ
ΠΛΗΡΟΦΟΡΙΕΣ ΓΙΑ
ΑΥΤΗΝ

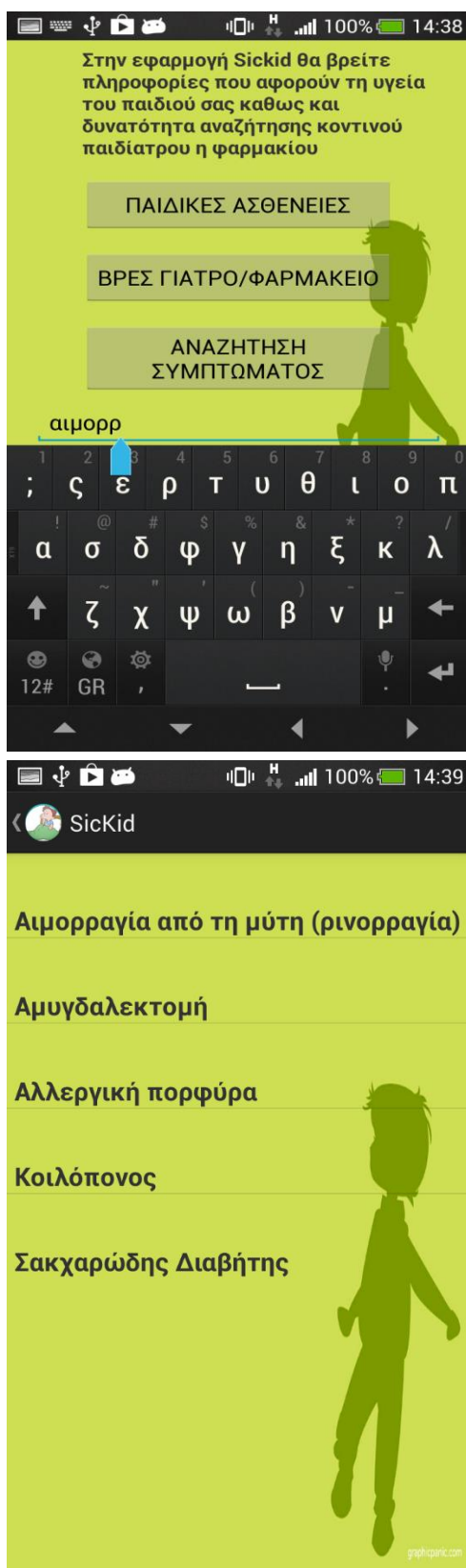
Λεπτομέρειες ασθένειας

The screenshot shows a mobile application interface. At the top, there is a status bar with icons for signal strength, 3G, 100% battery, and the time 14:33. Below the status bar is a dark navigation bar with a back arrow and the text 'SicKid'. The main content area has a light green background and features the title 'Βήχας-άσθμα' in a large, bold font. Below the title is a paragraph of text explaining that cough and asthma are common causes of pediatric visits and that asthma is the most common chronic condition in children. It notes that although cough is a symptom of asthma, it is not always the case, and that every child who coughs has asthma. It then discusses the relationship between cough and asthma symptoms and aims to help parents understand these conditions better to provide appropriate care for their children. Below this is a sub-section titled 'Τι είναι βήχας;' followed by another paragraph explaining that cough is a common symptom in children, sometimes causing concern for parents, and that it can be disruptive for both the child and their environment. It states that cough is a natural physiological defense mechanism of the body. A faint silhouette of a child is visible in the background of the text area. At the bottom right of the page, the text 'graphicpanic.com' is visible.

ΕΠΙΣΤΡΟΦΗ
ΠΙΣΩ

ΕΠΙΛΕΞΤΕ ΚΑΠΟΙΑ
ΑΣΘΕΝΕΙΑ ΓΙΑ ΝΑ
ΔΙΑΒΑΣΤΕ
ΠΛΗΡΟΦΟΡΙΕΣ ΓΙΑ
ΑΥΤΗΝ

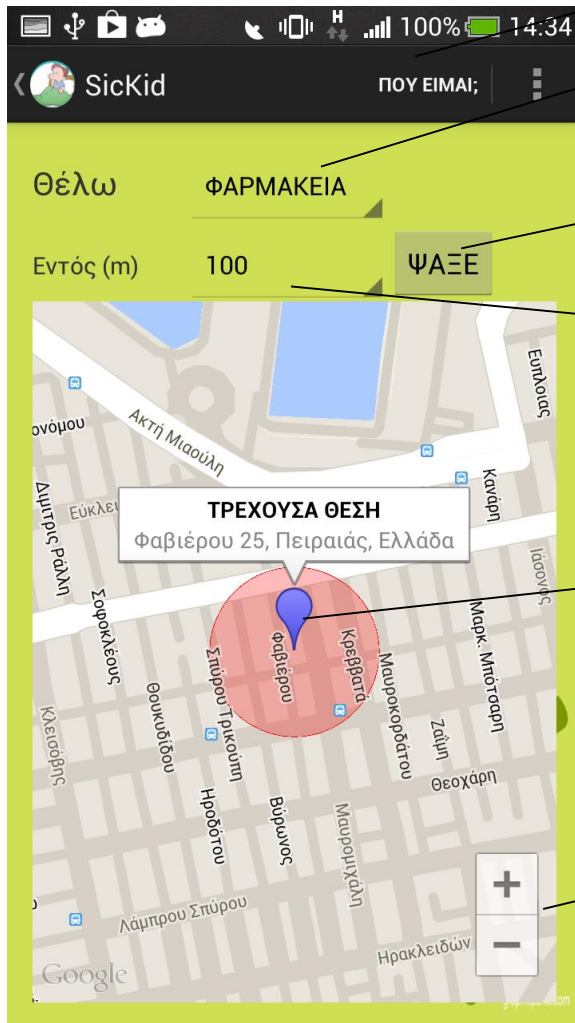
Αναζήτηση συμπτώματος



ΓΡΑΨΤΕ ΤΟ
ΣΥΜΠΤΩΜΑ ΠΟΥ
ΑΝΑΖΗΤΗΤΕ ΚΑΙ
ΠΑΤΗΣΤΕ ΑΝΑΖΗΤΗΣΗ
ΣΥΜΠΤΩΜΑΤΟΣ

ΑΠΟΤΕΛΕΣΜΑΤΑ
ΑΝΑΖΗΤΗΣΗΣ ,
ΕΠΙΛΕΞΤΕ ΚΑΠΟΙΟ ΓΙΑ
ΑΝΑΛΥΤΙΚΕΣ
ΠΛΗΡΟΦΟΡΙΕΣ

Αναζήτηση Φαρμακείου – Γιατρού



ΕΠΙΛΟΓΗ ΓΙΑ
ΕΝΤΟΠΙΣΜΟ ΤΗΣ
ΘΕΣΗΣ ΠΟΥ
ΒΡΙΣΚΕΣΤΕ

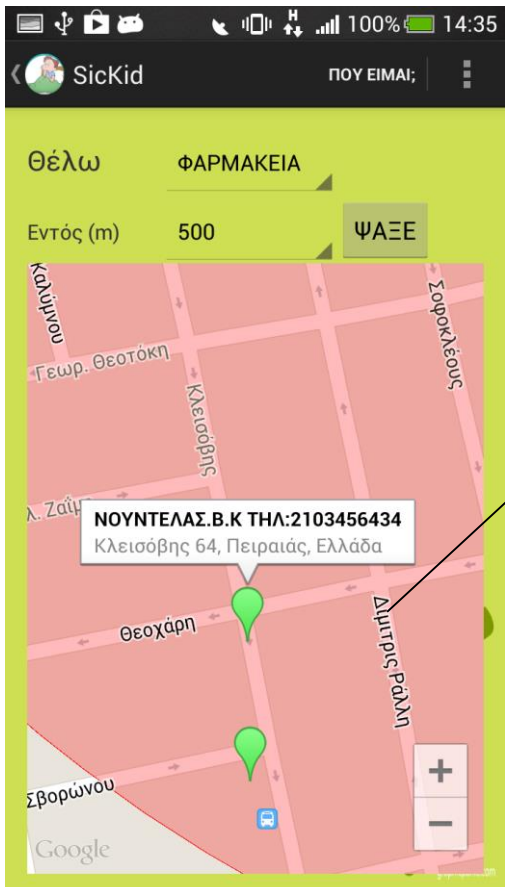
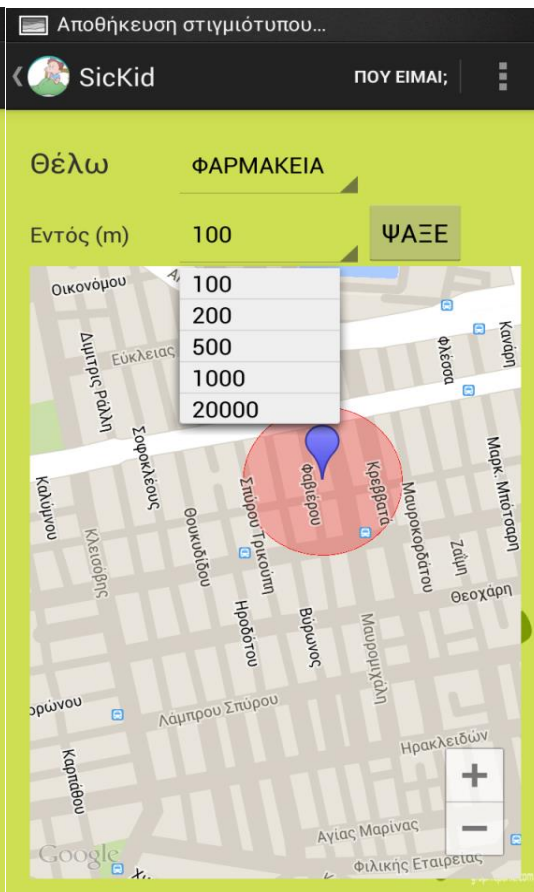
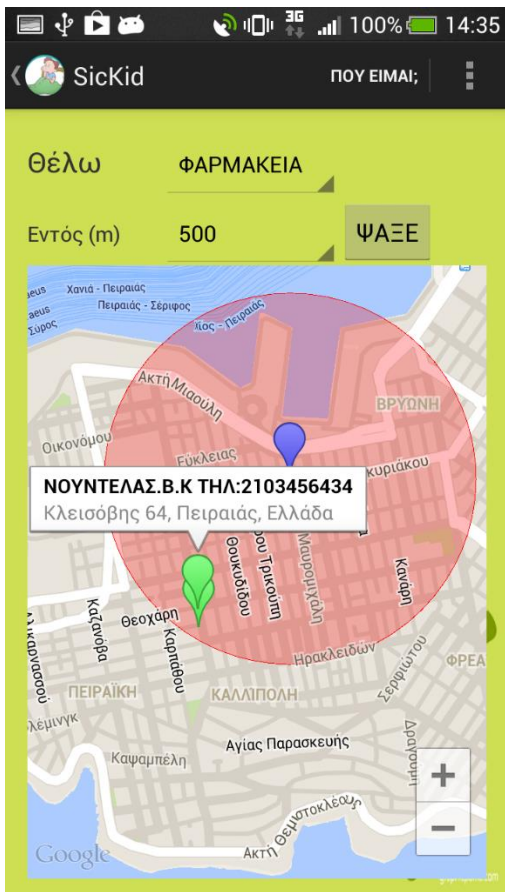
ΕΠΙΛΟΓΗ
ΚΑΤΗΓΟΡΙΑΣ
ΑΝΑΖΗΤΗΣΗ

ΕΠΙΛΟΓΗ ΓΙΑ ΝΑ
ΞΕΚΙΝΗΣΕΙ Η
ΑΝΑΖΗΤΗΣΗ

ΕΠΙΛΟΓΗ
ΜΕΓΙΣΤΗΣ ΑΠΟΣΤΑΣΗΣ
ΣΤΗΝ ΟΠΟΙΑ ΘΕΛΕΤΕ ΝΑ
ΓΙΝΕΙ Η ΑΝΑΖΗΤΗΣΗ

ΕΠΙΛΕΓΟΝΤΑΣ ΤΟ ΜΠΛΕ
ΜΑΡΚΕΡ ΣΤΟΝ ΧΑΡΤΗ
ΕΜΦΑΝΙΖΕΤΑΙ Η
ΔΙΕΥΘΗΝΣΗ ΣΤΗΝ ΟΠΟΙΑ
ΕΝΤΟΠΙΖΕΤΑΙ ΤΟ ΚΙΝΗΤΟ
ΣΑΣ

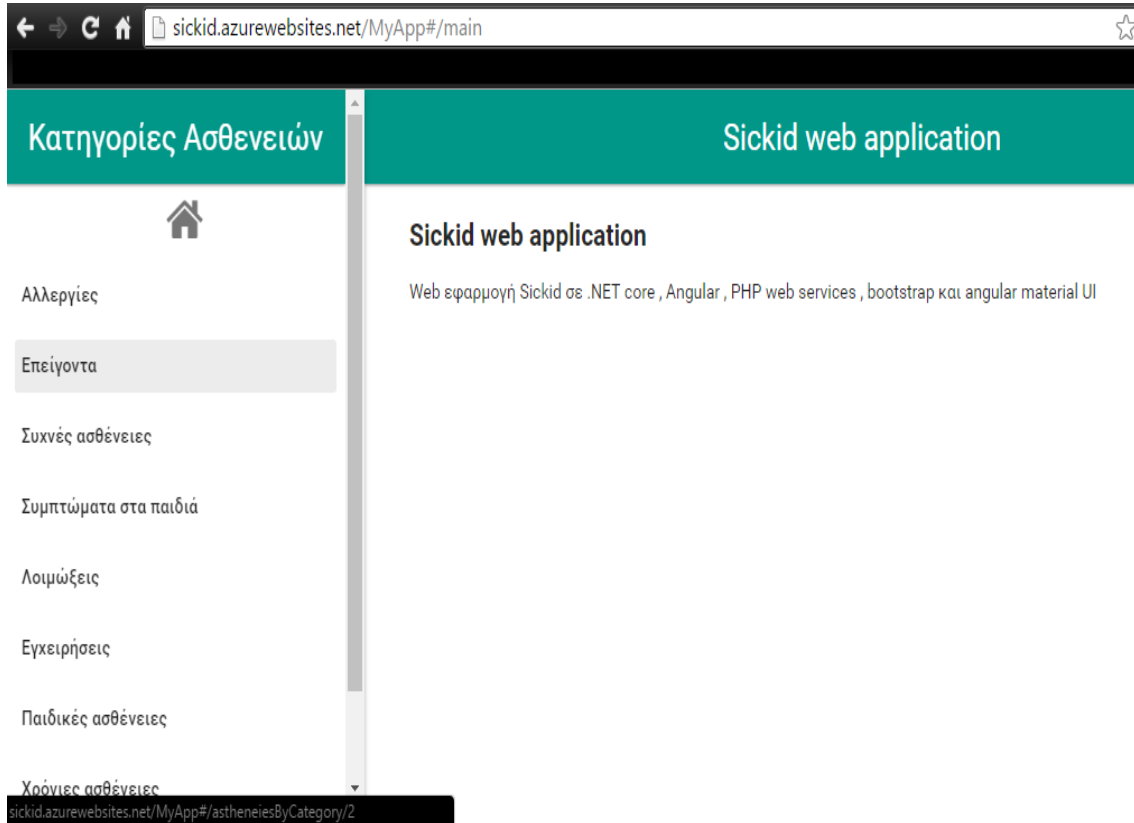
ΕΠΙΛΟΓΗ ZOOM IN ΚΑΙ
ZOOM OUT ΣΤΟΝ ΧΑΡΤΗ



ΕΠΙΛΕΓΟΝΤΑΣ ΤΟ ΠΡΑΣΙΝΟ
ΜΑΡΚΕΡ ΕΜΦΑΝΙΖΕΤΑΙ Η
ΔΙΕΥΘΗΝΣΗ ΤΟ ΤΗΛΕΦΩΝΟ ΚΑΙ
ΤΟ ΟΝΟΜΑ ΚΑΠΟΙΟΥ
ΦΑΡΜΑΚΕΙΟΥ ΠΟΥ ΕΧΕΙ
ΕΝΤΟΠΙΣΤΕΙ ΜΕΤΑ ΤΗΝ
ΑΝΑΖΗΤΗΣΗ

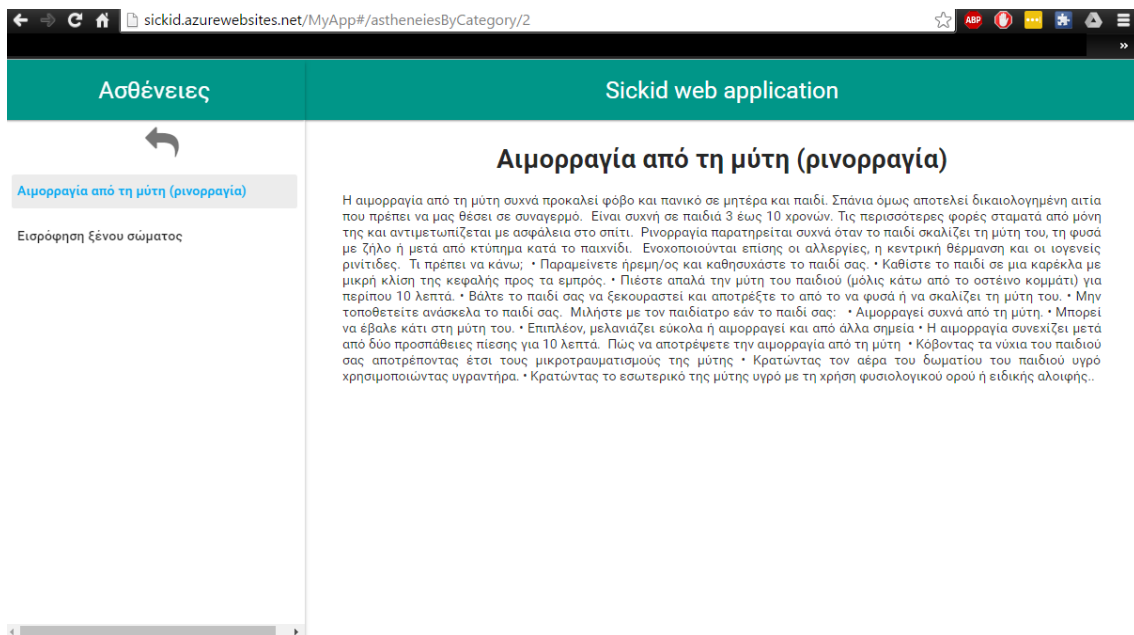
Web εφαρμογή angular js.

Επιλογή κατηγορίας



The screenshot shows a web browser at the URL `sickid.azurewebsites.net/MyApp#/main`. The application has a teal header with the text "Κατηγορίες Ασθενειών" on the left and "Sickid web application" on the right. Below the header is a sidebar menu with a home icon and the following categories: "Αλλεργίες", "Επείγοντα" (highlighted), "Συχνές ασθένειες", "Συμπτώματα στα παιδιά", "Λοιμώξεις", "Εγχειρήσεις", "Παιδικές ασθένειες", and "Χρόνιες ασθένειες". The main content area displays "Sickid web application" and a description: "Web εφαρμογή Sickid σε .NET core , Angular , PHP web services , bootstrap και angular material UI".

Επιλογή ασθένειας



The screenshot shows the web browser at the URL `sickid.azurewebsites.net/MyApp#/astheneiesByCategory/2`. The header now shows "Ασθένειες" on the left and "Sickid web application" on the right. The sidebar menu has a back arrow and the following items: "Αιμορραγία από τη μύτη (ρινορραγία)" (highlighted) and "Εισαρόφηση ξένου σώματος". The main content area features the title "Αιμορραγία από τη μύτη (ρινορραγία)" and a detailed paragraph of text explaining the condition, its symptoms, and first aid measures.

Αιμορραγία από τη μύτη (ρινορραγία)

Η αιμορραγία από τη μύτη συχνά προκαλεί φόβο και πανικό σε μητέρα και παιδί. Σπάνια όμως αποτελεί δικαιολογημένη αιτία που πρέπει να μας θέσει σε συναγερμό. Είναι συχνή σε παιδιά 3 έως 10 χρονών. Τις περισσότερες φορές σταματά από μόνη της και αντιμετωπίζεται με ασφάλεια στο σπίτι. Ρινορραγία παρατηρείται συχνά όταν το παιδί σκαλίζει τη μύτη του, τη φυσά με ζήλο ή μετά από κτύπημα κατά το παιχνίδι. Ενοχοποιούνται επίσης οι αλλεργίες, η κεντρική θέρμανση και οι ιογενείς ρινιτίδες. Τι πρέπει να κάνω; • Παραμείνετε ήρεμη/ος και καθησυχάστε το παιδί σας. • Καθίστε το παιδί σε μια καρέκλα με μικρή κλίση της κεφαλής προς τα εμπρός. • Πιέστε απαλά την μύτη του παιδιού (μόλις κάτω από το οστέινο κομμάτι) για περίπου 10 λεπτά. • Βάλτε το παιδί σας να ξεκουραστεί και αποτρέψτε το από το να φυσά ή να σκαλίζει τη μύτη του. • Μην τοποθετείτε ανάσκελα το παιδί σας. Μιλήστε με τον παιδίατρο εάν το παιδί σας: • Αιμορραγεί συχνά από τη μύτη. • Μπορεί να εβале κάτι στη μύτη του. • Επιπλέον, μελανιάζει εύκολα ή αιμορραγεί και από άλλα σημεία. • Η αιμορραγία συνεχίζει μετά από δύο προσπάθειες πίεσης για 10 λεπτά. Πώς να αποτρέψετε την αιμορραγία από τη μύτη • Κοβοντας τα νύχια του παιδιού σας αποτρέποντας έτσι τους μικροτραυματισμούς της μύτης • Κρατώντας τον αέρα του δωματίου του παιδιού υγρό χρησιμοποιώντας υγραντήρα. • Κρατώντας το εσωτερικό της μύτης υγρό με τη χρήση φυσιολογικού ορού ή ειδικής αλοιφής..

Κεφάλαιο 5: Συμπεράσματα και μελλοντικές επεκτάσεις

Συμπεράσματα

Κατά τη διάρκεια ανάπτυξης της εφαρμογής και της έρευνας που έγινε στις επιμέρους τεχνολογίες που χρησιμοποιήθηκαν εξήγαγα πολλά χρήσιμα συμπεράσματα για την εξέλιξη των mobile και web εφαρμογών.

Αρχικά θα ήθελα να κάνω μια αναφορά στην ταχύτητα εξέλιξης των client(frontend) τεχνολογιών που αφορούν τις mobile και web εφαρμογές. Δε θα μπορούσα λοιπόν να μην σχολιάσω την κυριαρχία της JavaScript σε όλο το φάσμα των mobile και web εφαρμογών και την ταχύτητα που εξελίσσονται τα frameworks που προσφέρουν δυνατότητες δημιουργίας client οι οποίοι μπορούν να τρέξουν σε όλες τις συσκευές ανεξάρτητου λειτουργικού συστήματος. Αν και οι native mobile εφαρμογές εξακολουθούν κατά την διάρκεια συγγραφής της πτυχιακής να έχουν πιο πλήρεις δυνατότητες τα javascript frameworks μειώνουν αυτό το χάσμα και η πολλά υποσχόμενη έκδοση 2.0 της angular καθώς και το ionic framework προσφέρουν νέες δυνατότητες ανάπτυξης mobile εφαρμογών για όλο το φάσμα των συσκευών pc, mobile, tablet.

Η JavaScript έχει κατακτήσει πλέον το client side κομμάτι του web και συνεχίζει να εξελίσσεται προσφέροντας μοναδικές δυνατότητες ανάπτυξης multiplatform εφαρμογών προσπαθώντας να κατακτήσει και το mobile κομμάτι έναντι της ανάπτυξης σε native περιβάλλον. Οι εξελίξεις συνεπώς σε αυτόν τον τομέα μπορεί να είναι ραγδαίες.

Όσον αφορά το κομμάτι του backend και των web services και εδώ οι εξελίξεις υπήρξαν πολλές ειδικά με τον ερχομό των σύγχρονων τεχνολογιών της Microsoft .NET5 /MVC 6 τα οποία κατά τη διάρκεια συγγραφής της πτυχιακής βρίσκονται σε Release Candidate έκδοση . Όπως αναφέρθηκε οι δυνατότητες δημιουργίας web services που προσφέρει το νέο framework θα μπορούσαν να αποτελέσουν μια ιδανική εναλλακτική της php.

Άλλη μια εναλλακτική που εξετάστηκε και καταδεικνύει την δυναμική εξέλιξη της JavaScript σε όλο το φάσμα των web εφαρμογών ακόμα και στο backend είναι το framework Node JS. Το Node JS πρόκειται για javascript framework το οποίο δίνει δυνατότητα στην javascript να λειτουργήσει στο server side κομμάτι της εφαρμογής προσφέροντας εφάμιλλες δυνατότητες με την php , .NET 5 και όλες τις server side γλώσσες . Το Node JS είναι από την φύση του σχεδιασμένο για σύγχρονες real time web εφαρμογές με πολύ καλές δυνατότητες scaling και ασύγχρονης επικοινωνίας με τον server. Το NodeJS είναι ασύγχρονο από την σχεδίασή του. Η επέλαση λοιπόν της JavaScript σε όλο το φάσμα των εφαρμογών από το client side μέχρι το server side κομμάτι είναι γεγονός και λύνει τα χέρια σε πολλούς full stack developers που αναζητούν μια κοινή γλώσσα σε frontend και backend.

Η ωριμότητά και απλότητα της php ωστόσο επικράτησε έναντι των παραπάνω επιλογών που αξίζουν όμως συμπερασματικά να αναφερθούν σαν δυο από τις πιο δυνατές εναλλακτικές σε περίπτωση επέκτασης της εφαρμογής σε εμπορικό επίπεδο.

Μελλοντικές επεκτάσεις

Οι επεκτάσεις όσον αφορά το χρηστικό κομμάτι της εφαρμογής αφορούν κυρίως την δημιουργία μιας web πλατφόρμας και mobile εφαρμογών ενημέρωσης που αφορούν το παιδί.

Η επιλογή της δημιουργίας του .NET 5 project έγινε έχοντας κατά νου αυτή την επέκταση. Το κύριο χαρακτηριστικό της επέκτασης είναι η δυνατότητα χωρικής αναζήτησης και στην web εφαρμογή.

Επίσης η ύπαρξη εγγραφής χρηστών στη web εφαρμογή επιδιώκει την δυνατότητα να μπορούν γιατροί και φαρμακοποιοί με το ανάλογο ρόλο χρήστη να δηλώσουν την διεύθυνση τους ιατρείου ή του φαρμακείου τους στο λογαριασμό τους ώστε να εμπλουτιστούν τα δεδομένα της χωρικής αναζήτησης της εφαρμογής.

Φυσικά όλα αυτά απαιτούν προσπάθεια και ανθρώπινους πόρους που θα πρέπει να διαχειριστούν το κομμάτι του administration της εφαρμογής ώστε να παρέχετε στον τελικό χρήστη αξιόπιστη πληροφόρηση.

Επίσης σημαντική μελλοντική επέκταση είναι η δημιουργία ενός πιο ολοκληρωμένου RESTful Web api στο .NET 5 web api με καινούργια services και περισσότερα δεδομένα που να είναι διαθέσιμα σε οποιονδήποτε θέλει να τα εκμεταλλευτεί για την εφαρμογή του. Τα web services είναι σχεδιασμένα για να εξυπηρετούν αυτόν ακριβώς τον σκοπό την μεταφορά πληροφοριών σε οποιονδήποτε θέλει να τις εκμεταλλευτεί. Ειδικά σε έναν τόσο ευαίσθητο τομέα όπως η υγεία του παιδιού η ύπαρξη εύκολης πρόσβασης στην πληροφόρηση κρίνεται αναγκαία.

Βιβλιογραφία

1. Simple Example of MVC (Model View Controller) Design Pattern for Abstraction
2. Άλμα πάνω↑ "More deeply, the framework exists to separate the representation of information from user interaction." The DCI Architecture: A New Vision of Object-Oriented Programming - Trygve Reenskaug and James Coplien - March 20, 2009.
3. Άλμα πάνω↑ "... the user input, the modeling of the external world, and the visual feedback to the user are explicitly separated and handled by three types of object." Applications Programming in Smalltalk-80(TM):How to use Model-View-Controller (MVC).
4. Buschmann, Frank (1996) *Pattern-Oriented Software Architecture*.
5. <http://webapptester.com/mvc-framework-first-impression/> (2013)
6. IBM (2007), "New to SOA and Web services"
7. MSDN (2001), "XML Web Services Basics"
8. Roseindia.net (2007), "Web Services"
9. Open Handset Alliance (5 Νοεμβρίου 2007). Industry Leaders Announce Open Platform for Mobile Devices. Δελτίο τύπου. Ανακτήθηκε στις 5 Νοεμβρίου 2007.
10. Άλμα πάνω↑ Shankland, Stephen (12 Νοεμβρίου 2007). «Google's Android parts ways with Java industry group». CNET News.
11. <http://web-mate.gr/ti-einai-i-angular/> (2015)
12. Καλόγηρος Βασίλειος, Υπ.Διδάκτωρ Π.Θ (2013-
<http://www.news.gr/ellada/ekpaidefsh/article/79047/kainotoma-metapyhiaka-syndyazoyn-plhroforikh-me-e.html>)
13. Αντώνης Μπιλλήρης -Περιοδικό NETWEEK (2009-
http://www.datamed.gr/index.php?option=com_content&view=article&id=167:2009-06-30-10-07-28&catid=52:articles&Itemid=167)
14. LOCATION BASED SERVICES – ΕΙΣΑΓΩΓΙΚΟ ΣΗΜΕΙΩΜΑ(Σπύρος Αθανασίου-Ντίνος Αρκουμάνης)
15. ΕΦΑΡΜΟΓΕΣ ΤΗΣ ΤΗΛΕΙΑΤΡΙΚΗΣ ΣΤΗΝ ΕΛΛΑΔΑ (ΚΑΡΑΣΤΕΡΓΙΟΥ Χ.)
16. <https://www.docker.com/what-docker> (2015)
17. «Most Widely Deployed SQL Database Estimates». Sqlite.org. May 11, 2011.

Παράρτημα

Εγκαταστάσεις εφαρμογής.

Web administration/web services : <http://apokapa.com/public/login.php>

Android app: Sickid.apk

Dotnetcore / Angular App:

Azure : <http://sickid.azurewebsites.net/>

Linux/docker : <http://sickid.haubit.com/>

Angular without dotnet portal : <http://apokapa.com/sickid/>

Repositories κώδικα εφαρμογής στο GITHUB

Web administration/web services : <https://github.com/apokapa/Sickid-App-Web-Administration>

Android app: <https://github.com/apokapa/Sickid-Android-App>

Dotnetcore / Angular App: <https://github.com/apokapa/Sickid-.NET5-Angular-App>