



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Σχεδιασμός και Υλοποίηση Κινητής Εφαρμογής  
Πληθοπορισμού για Αξιολόγηση και Ανασκόπηση  
Τόπων Πολιτιστικού Περιεχομένου σε Διεπαφή Χάρτη  
(React Native App Development)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΕΥΘΥΜΙΟΥ ΠΕΓΚΑ

Επιβλέπων: Γεώργιος Στάμου  
Καθηγητής Ε.Μ.Π.

ΕΡΓΑΣΤΗΡΙΟ ΤΕΧΝΟΛΟΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ

Αθήνα, Μάιος 2019





Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών  
Εργαστήριο Τεχνολογίας Λογισμικού

Σχεδιασμός και Υλοποίηση Κινητής Εφαρμογής  
Πληθοπορισμού για Αξιολόγηση και Ανασκόπηση  
Τόπων Πολιτιστικού Περιεχομένου σε Διεπαφή Χάρτη  
(React Native App Development)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΕΥΘΥΜΙΟΥ ΠΕΓΚΑ

Επιβλέπων: Γεώργιος Στάμου  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 29η Μαΐου 2019.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....  
Γεώργιος Στάμου  
Καθηγητής Ε.Μ.Π.

.....  
Ανδρέας Γ. Σταφυλοπάτης  
Καθηγητής Ε.Μ.Π.

.....  
Νικόλαος Παπασπύρου  
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάιος 2019

(Υπογραφή)

.....  
**ΕΥΘΥΜΙΟΣ ΠΕΓΚΑΣ**

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

© 2019 – All rights reserved



Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών  
Εργαστήριο Τεχνολογίας Λογισμικού

Copyright ©–All rights reserved Ευθύμιος Πέγκας, 2019.

Με επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται αποκλειστικά προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει σε καμία περίπτωση να ερμηνευτεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.



# Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Π.

Επίσης ευχαριστώ ιδιαίτερα τον μεταπτυχιακό φοιτητή κ. ... για την καθοδήγηση και ...

Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου, ...





# Περίληψη

Αντικείμενο της διπλωματικής εργασίας είναι η ανάπτυξη και υλοποίηση ενός αλγορίθμου για την εύρεση πιθανότερων εγγύτερων γειτόνων από συγκεκριμένες σημειακές εστίες σε μία υποθετική υπηρεσία για κατόχους κινητών τηλεφώνων. Όποτε κάποιος χρήστης υποβάλλει ένα ερώτημα, θέτει τρία κριτήρια: (i) μία σημειακή εστία ενδιαφέροντος  $q$ , (ii) τον επιθυμητό αριθμό  $k$  των αναζητούμενων γειτόνων, καθώς και (iii) ένα κατώφλι πιθανότητας  $\theta$ .

Για λόγους προστασίας του απορρήτου, κανένας χρήστης δεν αποκαλύπτει στους υπόλοιπους το ακριβές γεωγραφικό στίγμα του, αλλά δηλώνει μία ευρύτερη περιοχή αβεβαιότητας. Στην προκειμένη περίπτωση, οι περιοχές αυτές μοντελοποιούνται σύμφωνα με την κανονική κατανομή. Φυσικά, η αβεβαιότητα μπορεί να έχει διαφορετικές παραμέτρους, εκφράζοντας διαφορετικές βαθμίδες ιδιωτικότητας. Με τον όρο ‘πιθανότεροι εγγύτεροι γείτονες’, εννοούμε ότι σε μία συγκεκριμένη περιοχή αναζήτησης γύρω από την εστία  $q$ , έχουν βρεθεί τουλάχιστον  $k$  κινούμενοι χρήστες με πιθανοτική κάλυψη μεγαλύτερη από το δεδομένο κατώφλι  $\theta$ .

Η εργασία επικεντρώνεται κυρίως στην ανάπτυξη τεχνικών δεικτοδότησης, φιλτραρίσματος και κλαδέματος βάσει των οποίων θα μπορούμε να μειώσουμε το κόστος και τον χρόνο επεξεργασίας των δεδομένων. Ο αλγόριθμος που προτείνεται επιλέχτηκε να είναι προσεγγιστικός ως προς τον υπολογισμό της πιθανοτικής κάλυψης των περιοχών αβεβαιότητας και παρέχει μία λύση στο πρόβλημα της αποτίμησης πιθανοτικών ερωτημάτων εγγύτερων γειτόνων για αβέβαιες θέσεις κινούμενων αντικειμένων. Με εφαρμογή των παραπάνω τεχνικών, πραγματοποιήθηκαν πειράματα σε συνθετικά δεδομένα πάνω στον χάρτη της Αττικής, από τα οποία προέκυψαν θετικά αποτελέσματα. Επίσης, επιβεβαιώθηκαν οι αναμενόμενες επιδόσεις τους σχετικά με τους χρόνους εκτέλεσης και την ακρίβεια των απαντήσεων. Αυτό που μπορεί να εξαχθεί ως γενικό συμπέρασμα της εργασίας είναι ότι ο εν λόγω αλγόριθμος είναι κατάλληλος για προβλήματα πραγματικού χρόνου, θυσιάζοντας την ακρίβεια για χάρη της έγκαιρης απόκρισης.

## Λέξεις Κλειδιά

Αβεβαιότητα, Πιθανοτικά ερωτήματα εγγύτερων γειτόνων, διδιάστατη κανονική κατανομή, κινούμενα αντικείμενα, ρεύματα δεδομένων.



# Abstract

The purpose of this diploma thesis is to develop and implement an algorithm for most likely nearest neighbors monitoring from specific focal points in a hypothetical service for smartphone users. Whenever a user submits a most likely nearest neighbors query, sets three criteria: (i) a focal point of interest  $q$ , (ii) the desired number  $k$  of nearest neighbors, and (iii) a probability threshold  $\theta$ .

Because of privacy protection reasons, no user compromises their geographical position to the rest, but declares a wider *uncertainty region*. In this case, these regions are modeled according to the bivariate Gaussian distribution. Of course, uncertainty can acquire different parameters, expressing different scales of privacy. By using the term “*most likely nearest neighbors*”, we mean that in a certain search region around point  $q$ ,  $k$  moving users with probabilistic coverage above a certain threshold  $\theta$  have been found.

This thesis mainly focuses on developing indexing, filtering and pruning techniques which will enable us to reduce the cost and processing time of data. The suggested algorithm is deliberately chosen to be approximate in the calculation of probabilistic coverage of uncertain regions and provides a solution to the problem of answering probabilistic nearest neighbor queries for uncertain positions of moving objects. By utilizing the above techniques, a experimental study was conducted against synthetic datasets generated using the map of Athens. In addition, the expected performance on the execution times and accuracy of answers was confirmed. The overall conclusion of this thesis is that the algorithm is suitable for real time problems, where some accuracy may be sacrificed for the benefit of timely response.

## Keywords

Uncertainty, Probabilistic nearest neighbor queries, bivariate Gaussian distribution, moving objects, data streams.



# Περιεχόμενα

Ευχαριστίες	1
Περίληψη	3
Abstract	5
Περιεχόμενα	9
Κατάλογος Σχημάτων	11
Κατάλογος Πινάκων	13
Κατάλογος Παραθέσεων	15
1 Σχεδίαση Συστήματος	17
1.1 Σχεδίαση Μοντέλων των Διεπιφανειών της Εφαρμογής . . . . .	17
1.1.1 Αρχιτεκτονική Σχεδίασης Οθονών . . . . .	17
1.1.2 Λογική Σχεδίασης και Ροή Πληροφορίας Μεταξύ οθονών . . . . .	18
1.1.3 Διεπαφές Μηνυμάτων Προς το Χρήστη . . . . .	21
1.1.4 Screen Containers . . . . .	22
1.1.4.1 WelcomeScreen . . . . .	23
1.1.4.2 RegistrarScreen . . . . .	24
1.1.4.3 LoginScreen . . . . .	24
1.1.4.4 HomeScreen . . . . .	25
1.1.4.5 CreateHotspotScreen . . . . .	26
1.1.4.6 HotspotListScreen . . . . .	27
1.1.4.7 CommentScreen . . . . .	29
1.1.4.8 ProfileScreen . . . . .	30
1.2 Σχεδίαση Συστήματος Εξυπηρέτησης Αιτημάτων . . . . .	32
1.2.1 Λογική Σχεδίασης Εξυπηρετητή και Υποσυστήματα . . . . .	32
1.3 Σχεδίαση Μοντέλου Βάσης Δεδομένων . . . . .	33
1.3.1 Μοντελοποίηση Δεδομένων . . . . .	34

<b>2</b>	<b>Υλοποίηση Συστήματος</b>	<b>37</b>
2.1	Λεπτομέρειες Υλοποίησης Διεπιφανειών Χρήστη . . . . .	37
2.1.1	Βασικά Προγραμματιστικά Χαρακτηριστικά Οθονών . . . . .	38
2.1.1.1	Τεχνική Πλοήγησης στην Εφαρμογή . . . . .	38
2.1.1.2	Επικεφαλίδα Οθόνης . . . . .	38
2.1.1.3	Κεντρικό Μενού Οθονών . . . . .	38
2.1.1.4	Βασικά Πλήκτρα Ενεργειών . . . . .	38
2.1.2	Λειτουργικότητα Οθονών Εφαρμογής . . . . .	38
2.1.2.1	WelcomeScreen . . . . .	38
2.1.2.2	RegisterScreen . . . . .	38
2.1.2.3	LoginScreen . . . . .	38
2.1.2.4	HomeScreen . . . . .	38
2.1.2.5	CreateHotspotScreen . . . . .	38
2.1.2.6	HotspotListScreen . . . . .	38
2.1.2.7	CommentScreen . . . . .	38
2.1.2.8	ProfileScreen . . . . .	38
2.1.3	Τεχνικές Ενιαίας Διαχείρισης Δεδομένων . . . . .	38
2.1.3.1	React Redux . . . . .	38
2.1.3.2	Redux Thunk Middleware . . . . .	38
2.1.4	Βιβλιοθήκες και Βοηθητικά Προγραμματιστικά Πακέτα . . . . .	38
2.1.5	Ιεράρχηση Αρχείων και Δομή Φακέλων . . . . .	38
2.2	Σύστημα αξιολόγησης . . . . .	38
2.3	Οργάνωση πειραμάτων . . . . .	39
2.3.1	Παραγωγή συνθετικών δεδομένων . . . . .	39
2.4	Αποτελέσματα της μελέτης . . . . .	39
2.5	Σύνοψη συμπερασμάτων αξιολόγησης . . . . .	39
2.5.0.1	Server Routes . . . . .	40
2.5.0.2	Server Controllers . . . . .	40
<b>3</b>	<b>Έλεγχος</b>	<b>43</b>
3.1	Λεπτομέρειες υλοποίησης . . . . .	43
3.1.1	‘Τίτλος θέματος 1’ . . . . .	43
3.1.2	‘Τίτλος θέματος 1’ . . . . .	43
3.2	Πλατφόρμες και προγραμματιστικά εργαλεία . . . . .	43
<b>4</b>	<b>Επίλογος</b>	<b>45</b>
4.1	Σύνοψη και συμπεράσματα . . . . .	45
4.2	Μελλοντικές επεκτάσεις . . . . .	45
	<b>Βιβλιογραφία</b>	<b>47</b>
	<b>Κατάλογος Ακρονύμων</b>	<b>53</b>







# Κατάλογος Σχημάτων

1.1	Ροή πληροφορίας μεταξύ οθονών της εφαρμογής. . . . .	19
1.2	Επικοινωνία μεταξύ πατέρα-παιδιών. . . . .	23
1.3	Αρχικό σχέδιο της οθόνης έναρξης και των οθονών εγγραφής/σύνδεσης. . . . .	25
1.4	Αρχικό σχέδιο της κεντρικής οθόνης. . . . .	26
1.5	Αρχικό σχέδιο της οθόνης δημιουργίας νέου hotspot. . . . .	27
1.6	Αρχικό σχέδιο της οθόνης προσωπικών hotspots του χρήστη. . . . .	28
1.7	Αρχικό σχέδιο της οθόνης σχολιασμών επάνω σε κάποιο hotspot. . . . .	30
1.8	Αρχικό σχέδιο της οθόνης προσωπικού λογαριασμού του χρήστη. . . . .	31
1.9	Λογική σχεδίασης του συστήματος δρομολόγησης αιτημάτων. . . . .	32
1.10	Λογική σχεδίασης του συστήματος ελεγκτών στον εξυπηρετητή αιτημάτων της εφαρμογής. . . . .	33
1.11	Σχέδιο μοντελοποίησης των δεδομένων της εφαρμογής. . . . .	34
1.12	Φιλοσοφία πίσω από τη σχεδίαση της MongoDB. . . . .	35
2.1	Κλιμάκωση χρόνου εκτέλεσης για διάφορες υποδιαίρεσεις του καννάβου . . . . .	39
2.2	Ιεράρχηση του συστήματος δρομολογητών στην εφαρμογή. . . . .	40



# Κατάλογος Πινάκων

2.1	Παράμετροι πειραμάτων . . . . .	39
-----	---------------------------------	----



# Κατάλογος Παραθέσεων



# Κεφάλαιο 1

## Σχεδίαση Συστήματος

Στο προηγούμενο κεφάλαιο έγινε μια εισαγωγική αναφορά στα δομικά μέρη του συστήματος της εφαρμογής, δηλαδή του frontend (client) και του backend (server και βάση δεδομένων). Αναπτύχθηκαν οι λειτουργικότητες της εφαρμογής και οι αντίστοιχες οθόνες που θα τις καλύπτουν. Επίσης, έγινε μια αναφορά στα συστατικά του server και πως αυτά αλληλεπιδρούν με τη βάση δεδομένων για την λήψη και αποστολή δεδομένων από και προς τον client.

Αυτό το κεφάλαιο θα επικεντρωθεί στην σχεδίαση των υποδομών της εφαρμογής. Συγκεκριμένα, θα γίνει αναφορά στις διεπιφάνειες που θα συνιστούν κάθε οθόνη της εφαρμογής και θα παρουσιαστούν τα διαγράμματα ροής για τα σενάρια χρήσης της εφαρμογής (βλ. ενότητα 4.1). Στην ενότητα 4.2, θα γίνει αναφορά στην πλευρά του server. Θα παρουσιαστεί η διαδικασία σχεδίασης των εξυπηρετητών αιτημάτων και θα μελετηθούν οι τακτικές και οι τεχνολογίες που χρησιμοποιήθηκαν. Τέλος, στην ενότητα 4.3, θα γίνει μια εκτενής ανάλυση του τρόπου με τον οποίο σχεδιάστηκε η βάση δεδομένων για να ανταποκρίνεται βέλτιστα στις απαιτήσεις της εφαρμογής.

### 1.1 Σχεδίαση Μοντέλων των Διεπιφανειών της Εφαρμογής

Όπως έχει ήδη αναφερθεί, η εφαρμογή αποτελείται από οθόνες, καθεμία εκ των οποίων συνιστά και μια διεπαφή χρήστη. Σε αυτή την ενότητα θα αναλυθούν οι τεχνικές σχεδίασης των διεπαφών χρήστη και θα παρουσιαστούν παραστατικά διαγράμματα των συστατικών μερών τους.

#### 1.1.1 Αρχιτεκτονική Σχεδίασης Οθονών

Κατά την σχεδίαση μιας mobile εφαρμογής, ο προγραμματιστής καλείται να αποφανθεί σχετικά με τον τρόπο διαμερισμού του εσωτερικού της εφαρμογής (compartmentalization). Τα δομικά μέρη της εφαρμογής θεωρούνται ως εξαρτήματα, τα οποία στο σύνολό τους συνθέτουν μεγαλύτερα τμήματα της εφαρμογής. Τέτοια εξαρτήματα είναι τα πλήκτρα ενεργειών,

τα μενού, οι διεπαφές εισόδου και όλα τα διαδραστικά στοιχεία της εφαρμογής. Η κατάλληλη διασύνδεση μεταξύ πολλών διαδραστικών στοιχείων διαμορφώνει μια διεπιφάνεια χρήστη. Η διεπιφάνειες χρήστη ή αλλιώς κλάσεις, αποτελούν το κύριο συνθετικό στοιχείο (component) μιας οθόνης της εφαρμογής. Κάθε οθόνη περιλαμβάνει διαφορετικές κλάσεις, οι οποίες εξυπηρετούν διαφορετικούς σκοπούς. Για παράδειγμα, η οθόνη εγγραφής/σύνδεσης του χρήστη στην εφαρμογή αποτελείται από μια διεπαφή ατυτοποίησης, η οποία περιέχει μια φόρμα με πολλαπλά στοιχεία εισόδου κειμένου, πλήκτρα και μηνύματα. η κεντρική οθόνη της εφαρμογής συνίσταται από μια διεπαφή χάρτη, η οποία αποτελεί τη διεπιφάνεια προβολής όλων των hotspots και περιέχει markers για καθένα από τα hotspots.

Είναι προφανές λοιπόν πως, αν και η οθόνη ταυτοποίησης χρήστη διαφέρει ως προς το σκοπό της από την οθόνη χάρτη, η δομή τους ακολουθεί μια ομοιόμορφη αρχιτεκτονική σχεδίασης. Στη συνέχεια, θα γίνει μια αναλυτική αναφορά στις μεθόδους που χρησιμοποιήθηκαν κατά τη σχεδίαση των οθονών της εφαρμογής και θα παρουσιαστούν τα μοντέλα κάθε διεπιφάνειας.

### 1.1.2 Λογική Σχεδίασης και Ροή Πληροφορίας Μεταξύ οθονών

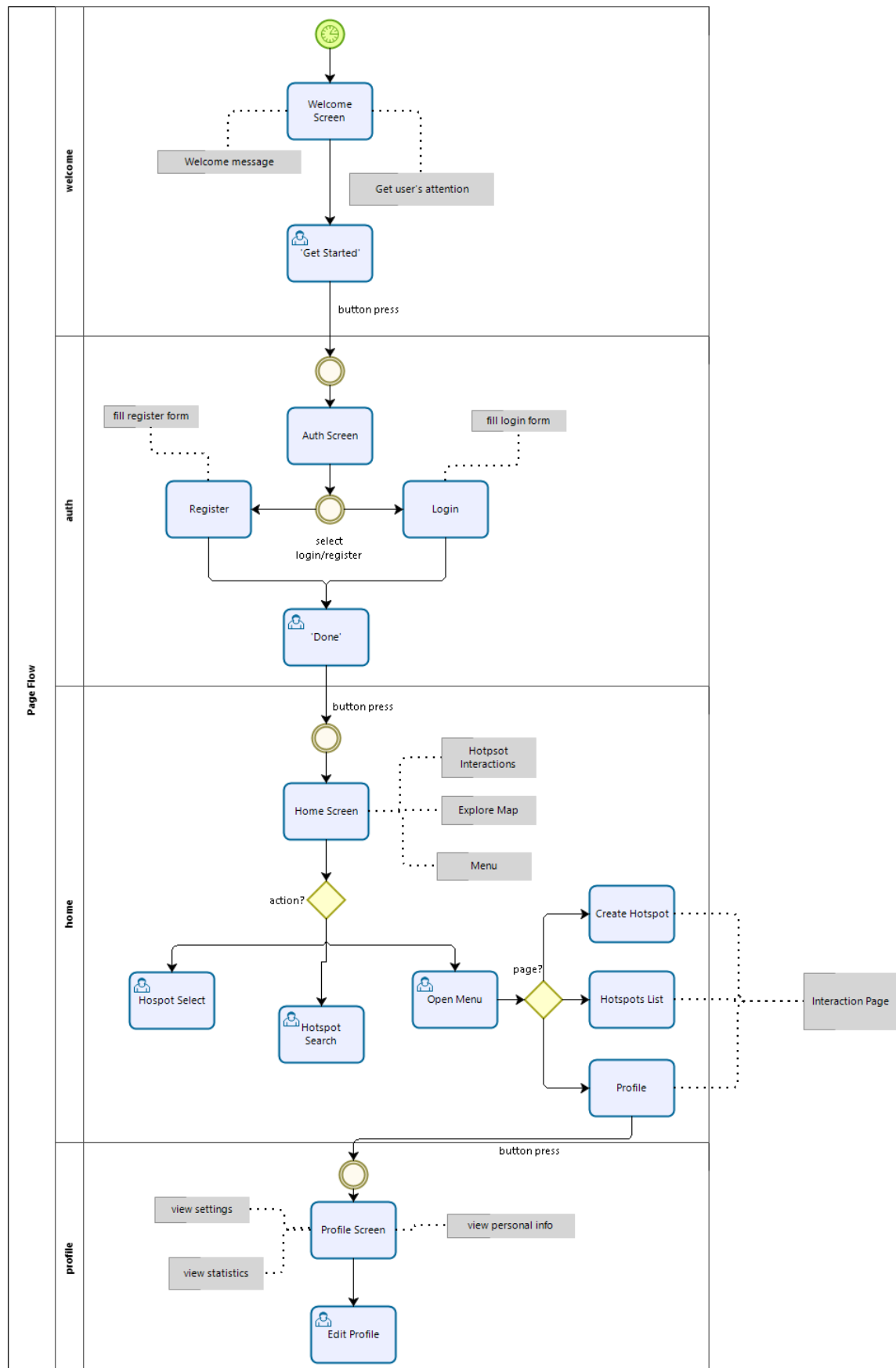
Η λογική σχεδίασης που ακολουθεί η εφαρμογή διακρίνεται σε 5 στάδια, τα οποία σειριοποιούνται ως εξής:

1. οθόνη έναρξης
2. ταυτοποίηση χρήστη
3. κεντρική οθόνη
4. οθόνες μηνυμάτων και αλληλεπίδρασης
5. προσωπικός λογαριασμός και ρυθμίσεις

Το μοντέλο αυτό ροής της πληροφορίας (βλ. Σχ. 1.1) ακολουθείται από την πλειοψηφία των σημερινών εφαρμογών. Κατά τη διαδικασία εξερεύνησης της εφαρμογής, ο χρήστης εισάγεται στον τρόπο λειτουργίας της εφαρμογής σταδιακά. Κάθε στάδιο προσφέρει στο χρήστη εμπειρία συγκεκριμένου τύπου, χωρίς να τον επιβαρύνει με αχρείαστες υπηρεσίες. Ο χρήστης θα πρέπει να μπορεί να αποκτήσει εμπειρία άμεσα. Η εκμάθηση της εφαρμογής θα πρέπει να είναι εύκολη και αβίαστη και όχι να δημιουργεί την αίσθηση ανικανότητας ή δυσκολίας χειρισμού.

Επίσης, κάθε σημείο της εφαρμογής θα πρέπει να είναι εύκολα προσβάσιμο από κάθε οθόνη της εφαρμογής. Για το λόγο αυτό κάθε οθόνη πρέπει να περιλαμβάνει μια επικεφαλίδα με πλήκτρα πλοήγησης και χαρακτηριστικά γνωρίσματα που ενημερώνουν το χρήστη για το την πορεία του εντός της εφαρμογής.





Σχήμα 1.1: Ροή πληροφορίας μεταξύ οθονών της εφαρμογής.

Στο πρώτο στάδιο της εφαρμογής, ο χρήστης εισάγεται με άμεσο και σύντομο τρόπο στο νόημα της εφαρμογής. Ένα φιλικό μήνυμα καλωσορίσματος εξυπηρετεί στη διαμόρφωση μιας θετικής πρώτης εντύπωσης στο χρήστη. Το μόνο απαραίτητο διαδραστικό μέσο είναι το πλήκτρο που θα μεταφέρει το χρήστη στο επόμενο στάδιο.

Το στάδιο ταυτοποίησης του χρήστη απαρτίζεται από δύο σελίδες. Η πρώτη σελίδα αποτελείται από τη φόρμα εγγραφής, ενώ η δεύτερη διαθέτει μια φόρμα σύνδεσης. Και οι δύο σελίδες συνοδεύονται από ένα πλήχρο υποβολής. Κάθε φόρμα θα πρέπει να είναι σαφής, τόσο ως προς τα στοιχεία που απαιτεί από το χρήστη, όσο και ως προς τον τρόπο συμπλήρωσής της. Για το λόγο αυτό γίνεται χρήση διευκρινιστικών τίτλων, παραδειγμάτων εισόδου και άλλων μηνυμάτων τέτοιου σκοπού. Επίσης, θα πρέπει να ληφθούν υπόψη πιθανά σφάλματα που μπορεί προκύψουν, όπως η αγνόηση ενός ή περισσότερων σημείων εισόδου, η λανθασμένη συμπλήρωση ενός ή περισσότερων κελιών, ο λανθασμένος συνδυασμός διαπιστευτηρίων του χρήστη κλπ. Η εφαρμογή θα διαθέτει μια πλήρως καταρτισμένη και εξυπηρετική διεπαφή εμφάνισης μηνυμάτων λάθους στο χρήστη, με σκοπό την εύκολη διόρθωση σφαλμάτων και τη γρήγορη σύνδεση/εγγραφή στην εφαρμογή.

Η κεντρική οθόνη είναι το πιο σύνθετο στάδιο της εφαρμογής. Αποτελείται από πολλά διαδραστικά στοιχεία και διεπαφές χρήστη, οι κυριότερες εκ των οποίων είναι ο χάρτης των hotspots, η μπάρα αναζήτησης, το μενού πλοήγησης και το μενού επιλογής φίλτρων. Αν και η πολυπλοκότητα είναι υψηλή, οι αντίστοιχες διεπιφάνειες είναι σχεδιασμένες με τέτοιο τρόπο, ώστε να απλοποιούν τις ενέργειες που απαιτούνται από το χρήστη. Εικονίδια, τίτλοι και βοηθητικές περιγραφές χρησιμοποιούνται καθ' όλη την έκταση της κεντρικής σελίδας. Τα μηνύματα είναι σύντομα, χρησιμοποιώντας όσο το δυνατόν λιγότερους χαρακτήρες για να διευκολύνουν το χρήστη και να δημιουργούν μια εύχρηστη και ευανάγνωστη διεπιφάνεια. Άλλωστε, ο χώρος στην οθόνη μιας κινητής συσκευής είναι εξ αρχής περιορισμένος και δεν ευνοεί το συνωστισμό πληροφοριών σε ένα σημείο ή την υπερβολική λεπτομέρεια στο σχεδιασμό.

Το ενδιάμεσο στάδιο ορίζεται από τις οθόνες αλληλεπίδρασης του χρήστη με τις εξειδικευμένες υπηρεσίες που προσφέρει η εφαρμογή. Η οθόνη γρήγορης προβολής ενός hotspot, η οθόνη προβολής λεπτομερειών ενός hotspot, η οθόνη δημιουργίας ενός hotspot και η οθόνη λίστας προσωπικών hotspots του χρήστη αποτελούν το πεδίο δράσης του χρήστη. Οι οθόνες αυτές διαμορφώνουν τη φυσιογνωμία της εφαρμογής και τη διακρίνουν από τις υπόλοιπες εφαρμογές, με τις μοναδικές λειτουργικότητες που προσφέρουν.

Τέλος, το στάδιο του προσωπικού λογαριασμού αποτελείται από τις οθόνες που αφορούν τα προσωπικά στοιχεία του χρήστη. Οι οθόνες αυτές είναι σχεδιασμένες έτσι ώστε να προσφέρουν με άμεσο και σύντομο αλλά περιεκτικό τρόπο μια εικόνα του λογαριασμού του εκάστοτε χρήστη. Επιπλέον, ο χρήστης έχει τη δυνατότητα να προβάλλει επιπρόσθετα στοιχεία που αφορούν τη δράση του εντός της εφαρμογής. Στην οθόνη του προσωπικού λογαριασμού, ο χρήστης μπορεί να τροποποιήσει τις προσωπικές του ρυθμίσεις, καθώς και να επεξεργαστεί τα προσωπικά του στοιχεία.

### 1.1.3 Διεπαφές Μηνυμάτων Προς το Χρήστη

Η εφαρμογή είναι σχεδιασμένη ώστε να επιτρέπει τη μέγιστη δυνατή ευελιξία στο χρήστη εντός της εφαρμογής. Ωστόσο, προκειμένου να αποφευχθούν σφάλματα και σενάρια μη επιτρεπτής λειτουργίας, είναι απαραίτητο να ληφθούν μέτρα που προστατεύουν το χρήστη εντός της εφαρμογής, αλλά και την ίδια την εφαρμογή. Είναι χαρακτηριστικό της εφαρμογής να δέχεται εισόδο από το χρήστη σε πολλαπλά σημεία, με αποτέλεσμα να είναι απαραίτητος κάθε φορά ο έλεγχος της παρεχόμενης εισόδου του χρήστη, για να εξασφαλίζεται κάθε φορά η εύρυθμη λειτουργία της εφαρμογής και ταυτοχρόνως να προλαμβάνονται ανεπιθύμητες ενέργειες από την πλευρά του χρήστη με τον καλύτερο δυνατό τρόπο.

Ο ρόλος του ελεγκτή εισόδου του χρήστη αποδίδεται σε μια διεπαφή μηνυμάτων που σχεδιάστηκε για αυτό το σκοπό. Η διεπαφή μηνυμάτων είναι υπεύθυνη για τον έλεγχο της εισόδου που δίνεται από την πλευρά του χρήστη, την κατάλληλη τροποποίηση και την αποδοχή ή απόρριψη της. Στη συνέχεια, εμφανίζει μήνυμα λάθους ή επιτυχίας ανάλογα με την έκβαση τις εκάστοτε ενέργειας. Σε κάθε οθόνη που διαθέτει ένα ή και περισσότερα στοιχεία εισόδου, είναι πιθανό αυτά να διαφέρουν μεταξύ τους ως προς τις απαιτούμενες προϋποθέσεις που πρέπει να ικανοποιούν. Για το λόγο αυτό η διεπαφή μηνυμάτων λάθους συνίσταται από ένα σύστημα, στο οποίο κάθε διεπαφή εισόδου αντιστοιχίζεται σε έναν ελεγκτή σφάλματος. Στην περίπτωση όπου η είσοδος του χρήστη δεν ανήκει μεταξύ των επιθυμητών, εμφανίζεται στην οθόνη κατάλληλο μήνυμα σφάλματος. Εάν η είσοδος που δωθεί από το χρήστη ανήκει στις επιτρεπτές τιμές, τότε η διεπαφή είτε εμφανίζει μήνυμα επιτυχίας, είτε ανακατευθύνει το χρήστη στην ανανεωμένη οθόνη.

Τα είδη των μηνυμάτων προς το χρήστη μπορούν να ταξινομηθούν σε τέσσερις κατηγορίες:

- μηνύματα σφάλματος
- μηνύματα επιβεβαίωσης
- μηνύματα επιτυχίας
- μηνύματα δικαιωμάτων

Τα μηνύματα σφάλματος εμφανίζονται στην περίπτωση που ο χρήστης δώσει είσοδο που δεν ανήκει στις επιτρεπτές τιμές. Κάθε είσοδος του χρήστη πρέπει να πληροί κάποιες προϋποθέσεις. Για παράδειγμα, σε μια φόρμα εγγραφής, το πεδίο ηλεκτρονικής διεύθυνσης θα πρέπει να περιέχει μια σειρά χαρακτήρων που πληρούν τα κριτήρια (πχ. example@domain.com). Στην περίπτωση που η είσοδος του χρήστη δεν ικανοποιεί την παραπάνω συνθήκη, ένα κατάλληλο μήνυμα λάθους εμφανίζεται στην οθόνη. Όταν η είσοδος του χρήστη είναι μεταξύ των επιθυμητών, το μήνυμα λάθους εξαφανίζεται.

Σε κάποια σημεία της εφαρμογής είναι πιθανό ο χρήστης να αλλάξει γνώμη για μια ενέργεια. Τα κρίσιμα αυτά σημεία συνοδεύονται από ένα στάδιο επιβεβαίωσης πριν την υποβολή των δεδομένων στη βάση και την πραγματοποίηση της αντίστοιχων λειτουργιών. Το στάδιο αυτό αποτελείται από ένα μήνυμα στο οποίο διευκρινίζονται στο χρήστη οι ενέργειες που θα

επακολουθήσουν. Εάν ο χρήστης είναι σύμφωνος με αυτές, τότε προχωράει στη διαδικασία αποδοχής, ενώ αν δεν επιθυμεί να πραγματοποιηθούν οι παραπάνω ενέργειες ακυρώνει τη διαδικασία. Με τον τρόπο αυτό, η εφαρμογή προσθέτει ένα ακόμη στάδιο προστασίας του χρήστη, από πιθανά δικά του σφάλματα. Για παράδειγμα, στην οθόνη δημιουργίας νέου hotspot, μόλις ο χρήστης συμπληρώσει σωστά την αντίστοιχη φόρμα και πατήσει το πλήκτρο υποβολής, ένα μήνυμα επιβεβαίωσης εμφανίζεται στην οθόνη.

Κατά τη διαδικασία δημιουργίας νέου hotspot, ο χρήστης επιλέγει ένα χρονικό διάστημα ισχύος του hotspot. Μετά την επιλογή, ένα μήνυμα επιτυχίας εμφανίζεται στην οθόνη, διευκρινίζοντας στο χρήστη το αποτέλεσμα της ενέργειας που μόλις πραγματοποιήθηκε. Στις περισσότερες περιπτώσεις επιτυχίας, ο χρήστης ενημερώνεται έμμεσα, από την ανανέωση της οθόνης με τη συμπερίληψη των καινούριων δεδομένων που παρείχε ο ίδιος.

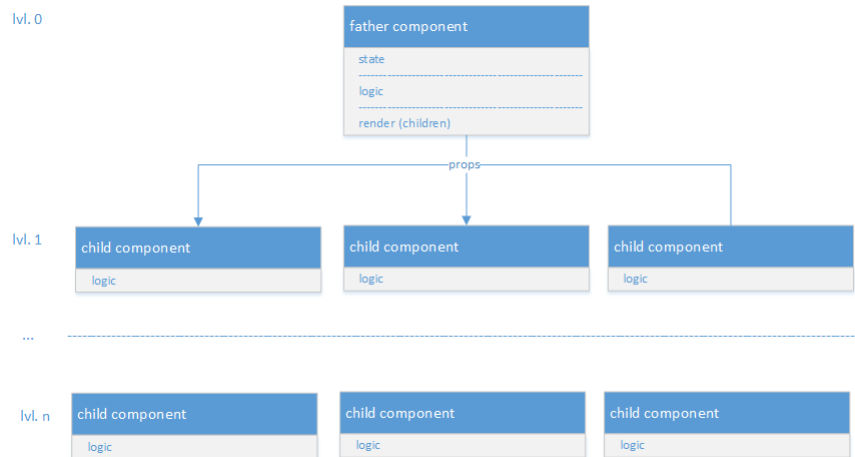
Τέλος, όπως έχει ήδη αναφερθεί, η εφαρμογή κάνει χρήση εξωτερικών ρυθμίσεων της συσκευής σε αρκετά σημεία. Για το λόγο αυτό είναι απαραίτητη η εμφάνιση κατάλληλων μηνυμάτων στα σημεία αυτά, όπου ζητείται από το χρήστη η παροχή του δικαιώματος στην εφαρμογή να πραγματοποιήσει αλλαγές στις ρυθμίσεις της συσκευής για την ολοκληρωμένη και εύρυθμη λειτουργία της. Ο χρήστης ενημερώνεται κάθε φορά με κατάλληλο μήνυμα για το είδος των ζητούμενων δικαιωμάτων και μπορεί να αποδεχθεί ή να απορρίψει την παροχή αυτών προς την εφαρμογή.

#### 1.1.4 Screen Containers

Κάθε οθόνη αποτελείται από ένα σύνολο κλάσεων και στοιχείων που συνθέτουν την διεπιφάνεια στην οποία δρα ο χρήστης. Το σύνολο αυτό συνιστά ένα container, ενώ τα στοιχεία που αποτελούν εξαρτήματα της διεπιφάνειας αποκαλούνται components. Κάθε screen έχει δικά της components, τα οποία επικοινωνούν μεταξύ τους μέσω του state. Η επικοινωνία μεταξύ components ακολουθεί το μοτίβο πατέρα-παιδιού, σύμφωνα με το οποίο η πληροφορία μεταδίδεται από το υψηλότερο (πατέρας) στα χαμηλότερα επίπεδα (παιδιά). Η σχέση αυτή είναι μονοσήμαντη. Αυτό σημαίνει πως η πληροφορία ρέει προς μία μόνο κατεύθυνση, από τον πατέρα προς τα παιδιά.

Τα components χωρίζονται σε δύο είδη. Αυτά που αναπαριστούν δεδομένα και αυτά που διαχειρίζονται την κατάσταση των δεδομένων στις διάφορες χρονικές στιγμές κατά τον κύκλο ζωής της εφαρμογής. Η πρώτη κατηγορία, αποτελείται από functional ή αλλιώς representational components. Τέτοιας μορφής είναι συνήθως τα παιδιά. Δεν διαθέτουν state και μπορούν μόνο να αναπαραστήσουν λογική. Από την άλλη, τα stateful components μπορούν να πραγματοποιήσουν αλλαγές στο state και να διαχειριστούν λογική. Κάθε οθόνη διαθέτει τουλάχιστον ένα stateful component το οποίο αποτελεί και τον πατέρα. Ο αριθμός των παιδιών ποικίλλει και εξαρτάται από τις λειτουργίες που καλείται να εξυπηρετήσει κάθε οθόνη. Είναι προτιμότερη η τακτική διάσπασης των λειτουργιών μιας οθόνης σε επιμέρους components, καθώς έτσι μειώνεται ο όγκος του κώδικα και αποσυμπλέκονται οι λειτουργίες. Κάθε παιδί αποτελείται από ένα ή και περισσότερα επίπεδα με functional components. Κάθε stateful component διαθέτει μία μέθοδο αναπαράστασης δεδομένων (render method). Ο πατέρας χρησιμοποιεί τη render

ως μέσο αναπαράστασης των παιδιών του. Ο συγχρονισμός της πληροφορίας επιτυγχάνεται χάρις την ιδιότητα του πατέρα να μεταφέρει το state στα παιδιά του μέσω των props.



Σχήμα 1.2: Επικοινωνία μεταξύ πατέρα-παιδιών.

#### 1.1.4.1 WelcomeScreen

Σκοπός της οθόνης έναρξης της εφαρμογής είναι να καλωσορίσει το χρήστη με ένα σύντομο και φιλικό μήνυμα. Η WelcomeScreen παίζει καθοριστικό ρόλο στην πρώτη εντύπωση του χρήστη. Για το λόγο αυτό, θα πρέπει να δωθεί ιδιαίτερη σημασία στα στοιχεία που θα χρησιμοποιηθούν σε αυτό το σημείο της εφαρμογής. Χωρίς πολλές πληροφορίες, ο χρήστης εισάγεται στο νόημα της εφαρμογής και παροτρύνεται να ξεκινήσει να τη χρησιμοποιεί.

Ένα εξίσου σημαντικό στοιχείο που πρέπει να ληφθεί υπόψη είναι η χρονική απόσταση μεταξύ WelcomeScreen και HomeScreen. Στην περίπτωση όπου η έναρξη αποτελεί μια πολυσέλιδη εισαγωγική διαδικασία, με αρκετά κείμενα, είναι πολύ πιθανό να αποθαρρύνει το χρήστη. Ο χρόνος που απαιτείται για να μεταβεί ο χρήστης από την οθόνη έναρξης στην κεντρική οθόνη της εφαρμογής πρέπει να είναι ο ελάχιστος δυνατός. Οι μόνοι παράγοντες από τους οποίους θα εξαρτηθεί αυτό, είναι η διαδικασία εγγραφής του χρήστη στην εφαρμογή και οι απαραίτητες ειδοποιήσεις που σχετίζονται με εξωτερικές ρυθμίσεις της συσκευής για την εύρυθμη και ολοκληρωμένη λειτουργία της εφαρμογής.

Είναι πολύ συχνό το φαινόμενο διαγραφής μιας εφαρμογής πρωτού ο χρήστης φτάσει στην κεντρική οθόνη. Αυτό οφείλεται κατά κύριο λόγο στα πολλαπλά μηνύματα που εμφανίζονται κατά τη διαδικασία εισόδου στην εφαρμογή. Συνεχόμενες ειδοποιήσεις σχετικές με ρυθμίσεις που απαιτούνται εντός της εφαρμογής κουράζουν το χρήστη και απομακρύνουν την εφαρμογή από το στόχο της, να κερδίσει δηλαδή το ενδιαφέρον του χρήστη. Οι ειδοποιήσεις θα πρέπει να βρίσκονται σε λογικά σημεία στην εφαρμογή και να εμφανίζονται μόνο όταν ο χρήστης πρόκειται να χρησιμοποιήσει μια υπηρεσία που απαιτεί εξωτερικούς πόρους και πρόκειται να αλλάξει τις ρυθμίσεις της συσκευής.

#### 1.1.4.2 RegistrarScreen

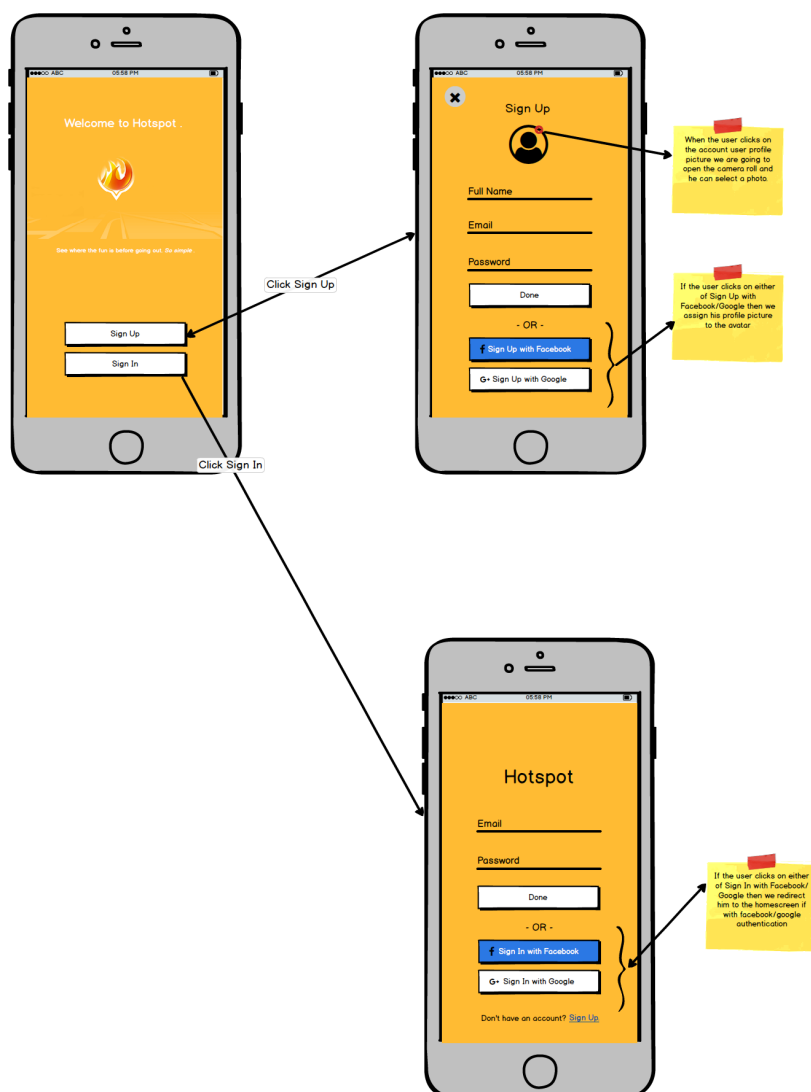
Η οθόνη εγγραφής του χρήστη στην εφαρμογή είναι ένα ιδιαίτερα κρίσιμο σημείο της εφαρμογής. Ο χρήστης σχηματίζει την πρώτη του εντύπωση και για το λόγο αυτό θα πρέπει η οθόνη εγγραφής να μην εμποδίζει τη διαμόρφωση μιας θετικής εικόνας. Αυτό σημαίνει πως θα πρέπει να είναι φιλική όσον αφορά στη χρήση της, να μην αποτελείται από κουραστικά στοιχεία και περιττές λεπτομέρειες και να είναι σύντομη. Στο σχεδιασμό της φόρμας εγγραφής θα πρέπει να ληφθεί υπόψη το γεγονός ότι ο χρήστης δεν αντιδρά θετικά στις καθυστερήσεις και τις πρόσθετες υποχρεώσεις που δημιουργεί η διαδικασία εγγραφής στην εφαρμογή. Έτσι, ο χρόνος που απαιτείται για τη συμπλήρωση της φόρμας και την ολοκλήρωση της διαδικασίας εγγραφής θα πρέπει να είναι ο ελάχιστος δυνατός.

Συνεπώς, η οθόνη θα περιέχει μόνο τα απαραίτητα στοιχεία για την σωστή εγγραφή του χρήστη στην εφαρμογή. Η φόρμα εγγραφής θα είναι σύντομη αλλά επεξηγηματική, τα στοιχεία εισόδου θα είναι λιτά αλλά ευδιάκριτα και το πλήκτρο υποβολής θα είναι κατάλληλου μεγέθους και εύκολο στη χρήση. Κάθε στοιχείο εισόδου θα αντιστοιχίζεται με την κατάλληλη μορφή του πληκτρολογίου, ανάλογα με τη φύση του. Δηλαδή, εάν η είσοδος απαιτεί ηλεκτρονική διεύθυνση, τότε το πληκτρολόγιο θα παρέχει γρήγορη πρόσβαση στα απαραίτητα σύμβολα και αλφαριθμητικούς χαρακτήρες που απαιτούνται στην περίπτωση αυτή. Ο κωδικός πρόσβασης θα προστατεύεται αυτόματα και για λόγους αποφυγής προβλημάτων θα ζητείται η επιβεβαίωσή του.

Στην περίπτωση όπου υπάρχουν σφάλματα, μια διεπαφή σφαλμάτων θα εμφανίζει κατάλληλα μηνύματα λάθους στο χρήστη. Τα μηνύματα θα είναι σύντομα και σαφή και θα στοχεύουν κάθε λάθος ξεχωριστά για την καλύτερη αντιμετώπισή τους από την πλευρά του χρήστη. Με αυτό τον τρόπο, επιδιώκεται η γρήγορη και αποτελεσματική συμπλήρωση της φόρμας και η μετάβαση στο κύριο τμήμα της εφαρμογής.

#### 1.1.4.3 LoginScreen

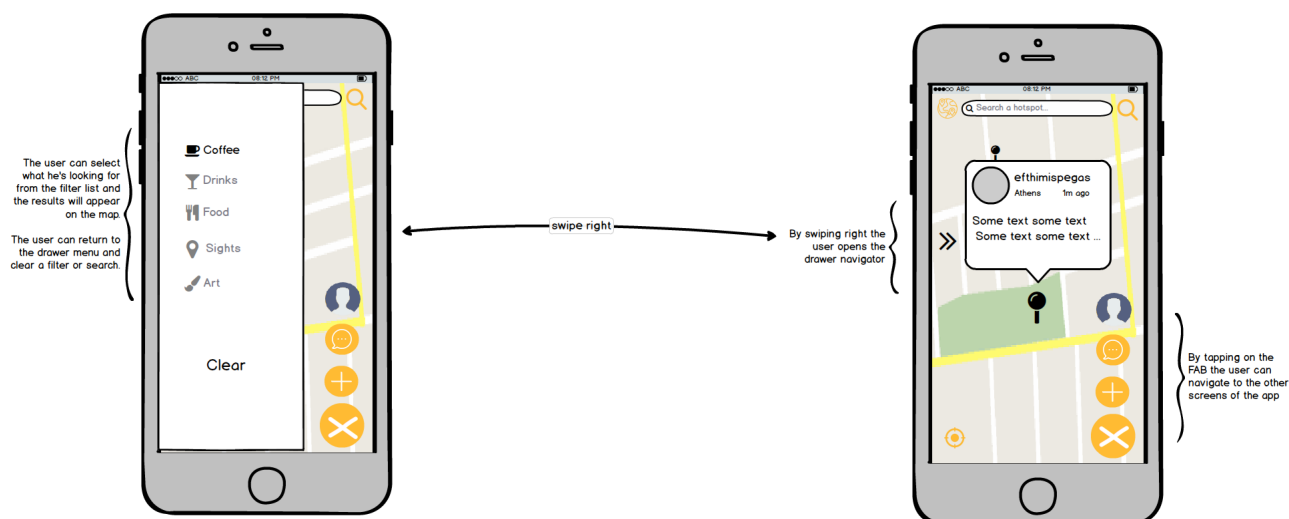
Η οθόνη σύνδεσης είναι πολύ πιο απλή από την οθόνη εγγραφής και έχει σκοπό την γρήγορη είσοδο του χρήστη στην εφαρμογή. Για το λόγο αυτό, περιέχει μόνο τις εισόδους για τα διαπιστευτήρια του χρήστη. Ο χρήστης στη φόρμα σύνδεσης καλείται να παραθέσει το συνδυασμό ηλεκτρονικής διεύθυνσης και κωδικού πρόσβασης. Στην περίπτωση που ο παραπάνω συνδυασμός δεν είναι σωστός, η φόρμα ανταποκρίνεται με κατάλληλο μήνυμα λάθους, μέσω μιας διεπαφής σφαλμάτων παρόμοια με αυτή της φόρμας εγγραφής.



Σχήμα 1.3: Αρχικό σχέδιο της οθόνης έναρξης και των οθονών εγγραφής/σύνδεσης.

#### 1.1.4.4 HomeScreen

Η κεντρική οθόνη αποτελεί τον πυρήνα της εφαρμογής. Για το λόγο αυτό, επιβάλλεται η σχεδιάσή της να είναι σφαιρική. Αυτό σημαίνει πως ο χρήστης θα πρέπει να μπορεί να αποκτήσει πρόσβαση σε όλες τις υπόλοιπες οθόνες από το σημείο αυτό. Οι περισσότερες λειτουργικότητες ξεκινούν από την οθόνη αυτή, πράγμα που καθιστά την απόκτηση εμπειρίας από την πλευρά του χρήστη εύκολη και άμεση.



Σχήμα 1.4: Αρχικό σχέδιο της κεντρικής οθόνης.

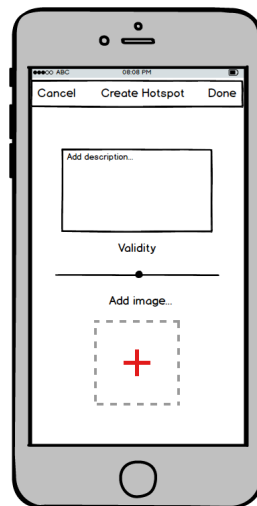
Η HomeScreen περιέχει περισσότερες λειτουργικότητες από κάθε άλλη οθόνη. Προκειμένου ο χρήστης να μπορεί να πλοηγηθεί στις υπόλοιπες οθόνες, θα διαθέτει ένα διακριτικό αλλά κομψό πλήκτρο στο κάτω δεξιά μέρος της οθόνης. Με το πάτημά του θα αναδύονται επιμέρους πλήκτρα με εικονίδια που αντιστοιχούν στις υπόλοιπες οθόνες της εφαρμογής. Για την αναζήτηση συγκεκριμένων σημείων ενδιαφέροντος, ο χρήστης θα εισάγει τις λέξεις-κλειδιά στην μπάρα αναζήτησης στο πάνω μέρος της οθόνης και έπειτα θα μπορεί είτε να επιλέξει ένα από τα προτεινόμενα αποτελέσματα, είτε να κάνει γενικευμένη αναζήτηση μέσω του πλήκτρου αναζήτησης. Επιπρόσθετα φίλτρα αναζήτησης θα εμφανίζονται όταν ο χρήστης σύρει το αριστερό άκρο της οθόνης προς τα δεξιά, ή πατάει στο εικονίδιο που βρίσκεται στο επάνω αριστερά μέρος της οθόνης. Προηγούμενες αναζητήσεις ή εφαρμοσμένα φίλτρα θα ανααιρούνται επιλέγοντας το πλήκτρο καθαρισμού από το ίδιο μενού. Τέλος, ο χρήστης θα μπορεί να επαναφέρει τη θέση του χάρτη στην τοποθεσία του μέσω πλήκτρου για το σκοπό αυτό που θα βρίσκεται στο κάτω αριστερά μέρος της οθόνης.

Από τα παραπάνω γίνεται αντιληπτή η αυξημένη πολυπλοκότητα που συνοδεύει τη σχεδίαση της HomeScreen. Ωστόσο, είναι κρίσιμο ο χρήστης να μην οδηγείται σε αυτό το συμπέρασμα κατά τη χρήση. Κάθε βήμα στη σχεδίαση καθεμιάς από τις προαναφερθείσες λειτουργικότητες θα πρέπει λοιπόν να χαρακτηρίζεται από απλότητα και να είναι βέλτιστη. Αυτό επιτυγχάνεται με τη χρήση απλών και ευδιάκριτων εικονιδίων, την προκαταβολική εκτέλεση πολλαπλών λειτουργιών σε προηγούμενο χρονικό σημείο και την υλοποίηση κάθε διεπαφής με εξελιγμένες βιβλιοθήκες και εργαλεία που χρησιμοποιούνται από τις πιο καταξιωμένες εφαρμογές σήμερα. Με αυτό τον τρόπο, δίνεται στο χρήστη η εντύπωση της αποδοτικής και ομαλής λειτουργίας της εφαρμογής, πράγμα που αποτελεί και τον τελικό στόχο.

#### 1.1.4.5 CreateHotspotScreen

Όταν ο χρήστης θέλει να δημιουργήσει ένα νέο hotspot, μπορεί να το κάνει πηγαίνοντας στην CreateHotspotScreen. Η οθόνη δημιουργίας νέου hotspot εξυπηρετεί έναν αποκλειστι-





Σχήμα 1.5: Αρχικό σχέδιο της οθόνης δημιουργίας νέου hotspot.

κό σκοπό. Για το λόγο αυτό, στη σχεδίασή της συμπεριλήφθηκαν στοιχεία που αφορούν στις ανάγκες της συγκεκριμένης οθόνης.

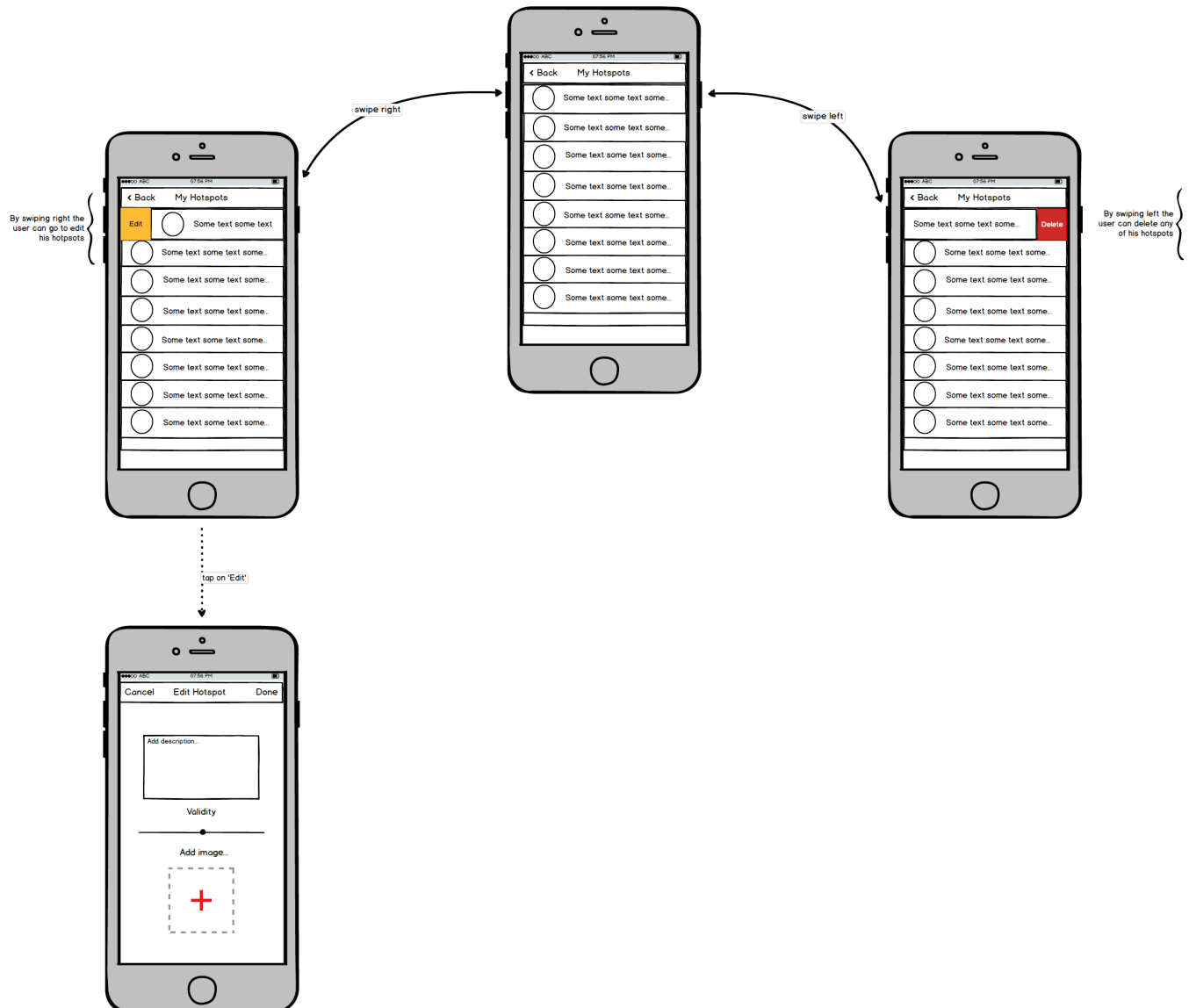
Οι προϋποθέσεις που πρέπει να πληροί η οθόνη για να είναι πλήρως λειτουργική και να συναντάει όλες τις απαιτήσεις, είναι η φόρμα δημιουργίας νέου hotspot, η διεπαφή για την ενσωμάτωση αρχείου εικόνας και τα βασικά πλήκτρα ενεργειών και πλοήγησης. Η φόρμα είναι βρίσκεται σε απλουστευμένη μορφή και αποτελείται από ένα στοιχείο εισόδου που αντιστοιχεί στην περιγραφή του hotspot προς δημιουργία, και ένα slider που αντιστοιχεί στο χρονικό διαστημα για το οποίο θα είναι έγκυρο το hotspot.

Όπως και κάθε οθόνη που δέχεται είσοδο από το χρήστη, έτσι και εδώ, απαιτείται κατάλληλος έλεγχος. Η είσοδος του χρήστη θα συνοδεύεται από διεπαφή μηνυμάτων κατάλληλης μορφής. Στην περίπτωση λάθους, η διεπαφή μηνυμάτων θα εμφανίζει στο χρήστη επεξηγηματικό μήνυμα λάθους, που θα προσδιορίζει το σημείο και το είδος του σφάλματος που έχει δημιουργηθεί. Εάν δεν υπάρχουν σφάλματα, τότε η υποβολή του νέου hotspot γίνεται με επιτυχία, και ο χρήστης επαναφέρεται στο χάρτη. Το καινούριο hotspot φαίνεται στο χάρτη, στην τοποθεσία του χρήστη.

#### 1.1.4.6 HotspotListScreen

Προκειμένου να γίνει περισσότερο λειτουργική, η εφαρμογή ταξινομεί τα προσωπικά hotspots του χρήστη. Πηγαίνοντας στη HotspotListScreen, ο χρήστης μπορεί να εξερευνηήσει τη λίστα με όλες τις δημοσιεύσεις του. Σε αυτή την οθόνη ο χρήστης μπορεί να διαχειριστεί τα προσωπικά του hotspots σε μόλις λίγα βήματα. Σύροντας την οθόνη προς τα επάνω, ο χρήστης μπορεί να περιηγηθεί στη σελίδα και να δει περισσότερα hotspots, καθώς και σχετικές πληροφορίες. Η λίστα είναι ταξινομημένη με βάση την ημερομηνία δημιουργίας. Έτσι, το πιο πρόσφατο hotspot βρίσκεται στην κορυφή, ενώ το παλαιότερο βρίσκεται στο τέλος της λίστας. Επίσης, για την γρηγορότερη φόρτωση και εμφάνιση των δεδομένων της οθόνης, η εφαρμογή φορτώνει τα πρώτα επτά hotspots του χρήστη. Έπειτα, ο χρήστης μπορεί να σύρει

την οθόνη προς τα επάνω και να πατήσει το πλήκτρο εμφάνισης περισσότερων αποτελεσμάτων με την επιγραφή ‘show more’.



Σχήμα 1.6: Αρχικό σχέδιο της οθόνης προσωπικών hotspots του χρήστη.

Στην οθόνη προσωπικών hotspots, Ο χρήστης έχει τη δυνατότητα να πραγματοποιήσει αλλαγές στα δικά του hotspots. Η λίστα είναι εφοδιασμένη με κρυφές λειτουργικότητες τις οποίες μπορεί να αξιοποιήσει ο χρήστης σύροντας ένα στοιχείο της λίστας προς μια οριζόντια κατεύθυνση. Για την επεξεργασία ενός hotspot, ο χρήστης σύρει ένα hotspot προς τα δεξιά. Τότε, πατώντας το πλήκτρο επεξεργασίας με την επιγραφή ‘edit’, η εφαρμογή μεταφέρει το χρήστη στη σελίδα επεξεργασίας του επιλεγμένου hotspot. Εκεί, ο χρήστης μπορεί να πραγματοποιήσει τις επιθυμητές αλλαγές στην περιγραφή ή την εικόνα του hotspot. Για την διαγραφή ενός hotspot, ο χρήστης σύρει ένα hotspot προς τα αριστερά. Τότε, πατώντας το πλήκτρο διαγραφής με το εικονίδιο ‘delete’, η εφαρμογή διαγράφει το επιλεγμένο hotspot από

τη λίστα.

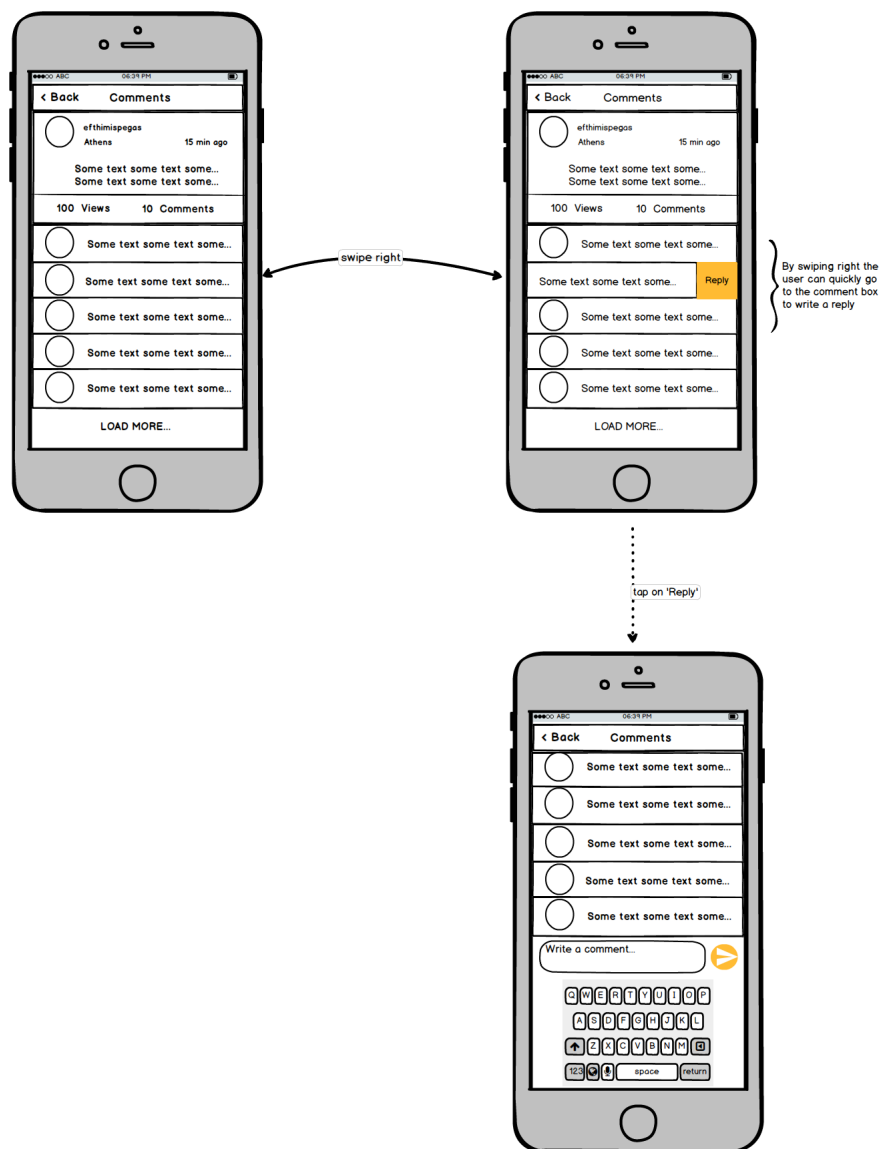
Η HotspotListScreen εξυπηρετεί την ανάγκη του χρήστη να ελέγχει την κατάσταση των προσωπικών του hotspots. Είναι σχεδιασμένη έτσι ώστε να παρέχει στο χρήστη τις επιθυμητές πληροφορίες για τα hotspots του, χωρίς να χρειάζεται να μεταβεί σε άλλη σελίδα. Για παράδειγμα, μπορεί γρήγορα να δει τον αριθμό των views και των comments ενός hotspot. Εάν ο χρήστης επιθυμεί να δει αναλυτικές πληροφορίες για ένα συγκεκριμένο hotspot, τότε μπορεί να πατήσει στο εικονίδιο σχολίων με την επιγραφή ‘comments’. Στη συνέχεια, η εφαρμογή μεταβαίνει στην οθόνη σχολίων CommentScreen του επιλεγμένου hotspot. Περισσότερα για την οθόνη αυτή αναφέρονται στην επόμενη παράγραφο.

#### 1.1.4.7 CommentScreen

Για μια πιο λεπτομερή περιήγηση στις πληροφορίες σχετικά με κάποιο hotspot, ο χρήστης μπορεί να μεταβεί στην CommentScreen. Εκεί βρίσκονται όλα τα δεδομένα που αφορούν ένα συγκεκριμένο hotspot, όπως ο αριθμός των comments και των views, σχόλια άλλων χρηστών στο συγκεκριμένο hotspot, απαντήσεις πάνω σε σχόλια και φωτογραφία του hotspot, αν υπάρχει.

Η οθόνη αυτή συμπεριλαμβάνει όλα εκείνα τα στοιχεία που έχουν να κάνουν με την αλληλεπίδραση των χρηστών με ένα συγκεκριμένο hotspot. Εάν ο χρήστης επιθυμεί να γράψει ένα σχολιασμό ή να απαντήσει σε κάποιο σχόλιο, μπορεί να το κάνει εδώ. Σύροντας την οθόνη προς τα επάνω, ο χρήστης μπορεί χειροκίνητα να μεταφερθεί στο τέλος της λίστας με τα σχόλια, όπου βρίσκεται η διεπαφή δημιουργίας νέου σχολίου. Ένας άλλος τρόπος να επιτευχθεί αυτό είναι σύροντας κάποιο σχόλιο προς τα αριστερά και πατώντας στο πλήκτρο ‘reply’. Η εφαρμογή τότε θα μεταφέρει αυτομάτως το χρήστη στη διεπαφή δημιουργίας νέου σχολίου. Η διεπαφή είναι σχεδιασμένη ώστε να είναι ευδιάκριτη και μοντέρνα, αλλά ταυτοχρόνως απλή. Το τμήμα δημιουργίας νέου σχολίου συνοδεύεται και από το πλήκτρο υποβολής του νέου σχολίου, το οποίο φέρει το χαρακτηριστικό εικονίδιο ‘send’.

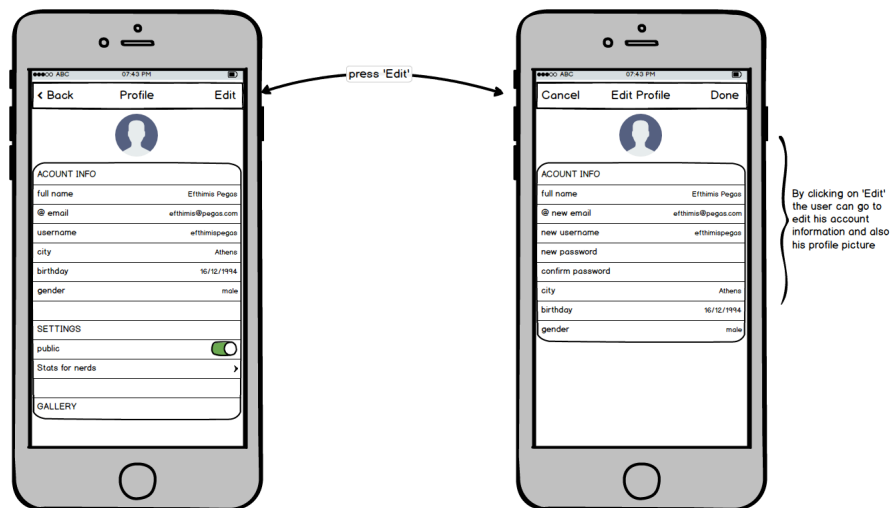
Η οθόνη είναι σχεδιασμένη ώστε να είναι αποδοτική και γρήγορη. Η λίστα με τα σχόλια θα πρέπει να φορτώνεται άμεσα. Η εμπειρία που προσφέρεται στο χρήστη θα πρέπει να είναι η καλύτερη δυνατή. Συνεπώς, η καθυστέρηση φόρτωσης θα πρέπει να είναι η ελάχιστη δυνατή. Η οθόνη είναι σχεδιασμένη με τέτοιο τρόπο, ώστε αν η λίστα με τα σχόλια είναι αρκετά μεγάλη, τότε να φορτώνεται μόνο ένας ορισμένος αριθμός σχολίων την πρώτη φορά. Εάν ο χρήστης επιθυμεί αν φορτώσει περισσότερα σχόλια, τότε μπορεί να περιηγηθεί στο τέλος της λίστας σύροντας την οθόνη προς τα επάνω και να πατήσει το πλήκτρο με την επιγραφή ‘load more’. Με την τμηματοποίηση της λίστας σχολίων, αποφεύγεται η περίπτωση μεγάλης καθυστέρησης φόρτωσης λόγω τεράστιου όγκου δεδομένων.



Σχήμα 1.7: Αρχικό σχέδιο της οθόνης σχολιασμών επάνω σε κάποιο hotspot.

#### 1.1.4.8 ProfileScreen

Ο προσωπικός λογαριασμός του χρήστη είναι ένα κρίσιμο σημείο της εφαρμογής και έχει δωθεί ιδιαίτερη προσοχή στη σχεδιάσή του. Ο χρήστης θα πρέπει να μπορεί, από τη μία να προβάλλει τα προσωπικά του στοιχεία και από την άλλη να επεξεργαστεί τις προσωπικές του ρυθμίσεις ή να αλλάξει το profile του. Εδώ ο χρήστης μπορεί να δει επίσης δεδομένα χρήσης της εφαρμογής, όπως ποσοστά αλληλεπίδρασης με άλλους χρήστες, συνολικό αριθμό σχολίων και προβολών, καθώς επίσης και το είδος του κοινού με το οποίο ο χρήστης αλληλεπιδρά περισσότερο. Στόχος της οθόνης profile είναι να ολοκληρώσει τη θετική εικόνα του χρήστη για την εφαρμογή και να αυξήσει το ενδιαφέρον του για αυτή.



Σχήμα 1.8: Αρχικό σχέδιο της οθόνης προσωπικού λογαριασμού του χρήστη.

Οι πληροφορίες που προβάλλονται στην ProfileScreen διακρίνονται σε τρεις κατηγορίες. Ταξινομημένα πρώτα είναι τα προσωπικά στοιχεία του χρήστη, με τα οποία συμπλήρωσε τη φόρμα εγγραφής την πρώτη φορά που εισήλθε στην εφαρμογή. Έπειτα ακολουθούν οι προσωπικές ρυθμίσεις του χρήστη, όπως το είδος του λογαριασμού που έχει δημιουργήσει (public ή private). Εάν ο χρήστης το επιθυμεί, μπορεί να προβάλλει τα στατιστικά δεδομένα που βρίσκονται σε ξεχωριστή σελίδα, πατώντας το βέλος που βρίσκεται στα δεξιά της επιγραφής 'Stats for nerds'. Τέλος, στο κάτω μέρος της οθόνης βρίσκεται η συλλογή φωτογραφιών του χρήστη από τα προσωπικά hotspots που έχει δημιουργήσει ο ίδιος στο παρελθόν.

Για την επεξεργασία των προσωπικών του στοιχείων, ο χρήστης μπορεί να μεταβεί στην σελίδα επεξεργασίας profile. Πατώντας το πλήκτρο με την επιγραφή 'Edit' που βρίσκεται στο δεξί μέρος της επικεφαλίδας, ο χρήστης μεταφέρεται στην EditProfileScreen. Εκεί, ο χρήστης μπορεί να επεξεργαστεί τα προσωπικά του στοιχεία, όπως το όνοματεπώνυμό του, την ηλεκτρονική του διεύθυνση, το όνομα χρήστη του, την εικόνα προφίλ του κλπ. Ακόμη, είναι εφικτή η αλλαγή των ρυθμίσεων του χρήστη, όπως το είδος του λογαριασμού. Για την επιτυχή αλλαγή των στοιχείων του ο χρήστης πρέπει να συμπληρώσει τον κωδικό πρόσβασης. Η φόρμα επεξεργασίας των προσωπικών δεδομένων του χρήστη συνοδεύεται από μια διεπαφή μηνυμάτων λάθους, όπως και κάθε οθόνη που απαιτεί είσοδο από το χρήστη. Στην περίπτωση όπου ένα ή περισσότερα πεδία είναι εσφαλμένα, κατάλληλο μήνυμα λάθους εμφανίζεται κάτω από το αντίστοιχο πεδίο. Στην περίπτωση όπου δεν υπάρχουν σφάλματα, η υποβολή της φόρμας ολοκληρώνεται με επιτυχία και ο χρήστης μεταφέρεται πίσω στον προσωπικό του λογαριασμό. Τα νέα στοιχεία του χρήστη φαίνονται στην οθόνη.

Χαρακτηριστικό της εφαρμογής αποτελεί το γεγονός ότι δεν είναι εύκολη η αποσύνδεση του χρήστη από την εφαρμογή. Αυτό οφείλεται στο γεγονός ότι το πλήκτρο αποσύνδεσης δεν υπάρχει σε καμία από τις προηγούμενες οθόνες. Έτσι, ο χρήστης δεν μπορεί να αποσυνδεθεί παρά μόνο όταν έχει ολοκληρώσει την περιήγησή του εντός της εφαρμογής και έχει αποκτήσει μια πλήρη εικόνα των υπηρεσιών που προσφέρει. Για την αποσύνδεση από την εφαρμογή, ο χρήστης μεταβαίνει στον προσωπικό του λογαριασμό και πατάει το πλήκτρο αποσύνδεσης

που φέρει την επιγραφή ‘Log out’. Έπειτα, μεταφέρεται αυτόματα στην οθόνη έναρξης της εφαρμογής.

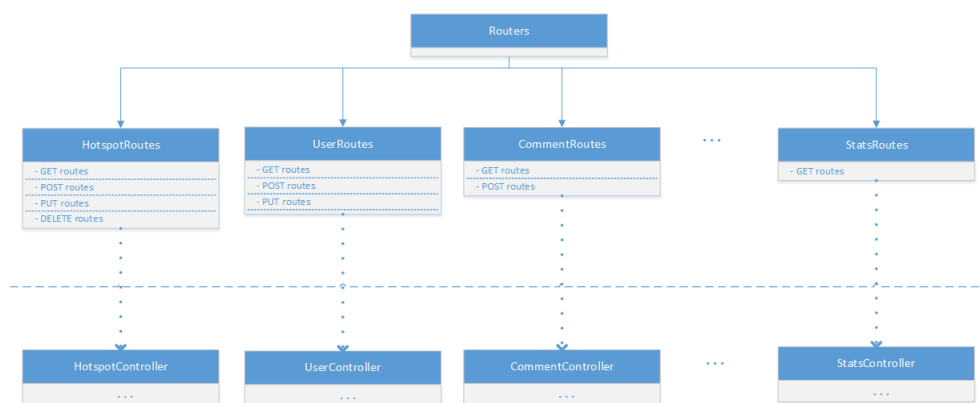
Είναι καθήκον της λογικής σχεδίασης της εφαρμογής να ενθαρρύνει συνεχώς την αλληλεπίδραση του χρήστη με τις υπηρεσίες της. Όπως φαίνεται από τα παραπάνω, αυτό επιτυγχάνεται τόσο μέσα από την παροχή πληθώρας λειτουργικιοτήτων, όσο και μέσω της παρουσίασης στατιστικών δεδομένων που αφορούν τη σχέση του χρήστη με το υπόλοιπο σύνολο και δημιουργούν μια εικόνα της θέσης του χρήστη εντός του οικοσυστήματος της εφαρμογής. Αυτό συνεισφέρει στη συνεχή δράση του χρήστη και την ανάπτυξη στενών σχέσεων με την εφαρμογή.

## 1.2 Σχεδίαση Συστήματος Εξυπηρέτησης Αιτημάτων

Στην προηγούμενη ενότητα καλύφθηκε η λογική που ακολουθήθηκε στη σχεδίαση του client της εφαρμογής. Αναλύθηκαν οι δομές που απαρτίζουν τις διάφορες διεπιφάνειες χρήστη, και παρουσιάστηκαν τα αρχικά σχέδια αυτών. Σε αυτή την ενότητα θα αναπτυχθούν οι τεχνικές σχεδίασης του server και των υποσυστημάτων του.

Στην σχεδίαση του εξυπηρετητή αιτημάτων της εφαρμογής, πρωταρχικός σκοπός είναι η αποσύμπλεξη των λειτουργιών που δημιουργεί ένα αίτημα. Κάθε ενέργεια που λαμβάνει χώρα σε μια από τις διεπιφάνειες χρήστη, πυροδοτεί μια σειρά κλήσεων που απαιτούν τη μεταφορά δεδομένων από και προς τον server. Ο server έχει σχεδιαστεί με τέτοιο τρόπο, ώστε κάθε αίτημα να αποσυναρμολογείται στις επιμέρους διακριτές λειτουργίες που πρέπει να εκτελεστούν. Οι λειτουργίες αυτές δρομολογούνται με τη σειρά τους στα κατάλληλα υποσυστήματα για την εξυπηρέτησή τους.

### 1.2.1 Λογική Σχεδίασης Εξυπηρετητή και Υποσυστήματα



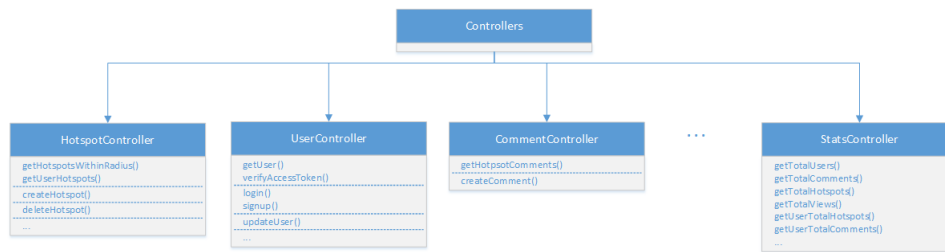
Σχήμα 1.9: Λογική σχεδίασης του συστήματος δρομολόγησης αιτημάτων.

Ο server της εφαρμογής θα συνίσταται από ένα σύστημα ελεγκτών (controllers) και δρομολογητών (routers), με σκοπό την αποτελεσματική και γρήγορη ικανοποίηση των αιτημάτων που δημιουργεί ο client. Κάθε δρομολογητής θα είναι υπεύθυνος για την σωστή ταξινόμηση

των αιτημάτων με βάση τη φύση τους και την δρομολόγηση στον κατάλληλο ελεγκτή. Έπειτα, ο ελεγκτής θα εκτελεί μια σειρά λειτουργιών που αντιστοιχούν στο συγκεκριμένο αίτημα που δημιουργήθηκε. Μόλις ολοκληρωθεί η παραπάνω διαδικασία, το αποτέλεσμα επιστρέφεται σε κατάλληλη μορφή πίσω στον client.

Το σύστημα δρομολόγησης των αιτημάτων διαιρείται σε επιμέρους routers (βλ. Σχ. 1.9). Για την καλύτερο προγραμματισμό του συστήματος και την απλοποίηση της σχεδίασης, επιλέχθηκε ως κριτήριο διάκρισης η φύση των λειτουργιών που εξυπηρετεί κάθε route (περισσότερα στην ενότητα ΠΟΙΑ ΕΝΟΤΗΤΑ).

Το σύστημα εξυπηρέτησης των αιτημάτων υλοποιείται από controllers που καθορίζουν τις εντολές που θα εκτελεστούν για την ολοκλήρωση κάθε αιτήματος (βλ. Σχ. 1.10). Οι εντολές συνοψίζονται σε μεθόδους, γνωστές ως ελεγκτές. Κάθε ελεγκτής αντιστοιχίζεται στην εξυπηρέτηση ενός αποκλειστικού αιτήματος. Το σύστημα των ελεγκτών είναι σχεδιασμένο ώστε να γίνεται κατανοητό από τον προγραμματιστή και ευέλικτο όσον αφορά στην επεξεργασία του. Όπως και στο σύστημα των δρομολογητών, έτσι και εδώ, κριτήριο στην διακριτοποίηση των ελεγκτών αποτελεί η φύση του αιτήματος (περισσότερα στην ενότητα ΠΟΙΑ ΕΝΟΤΗΤΑ).



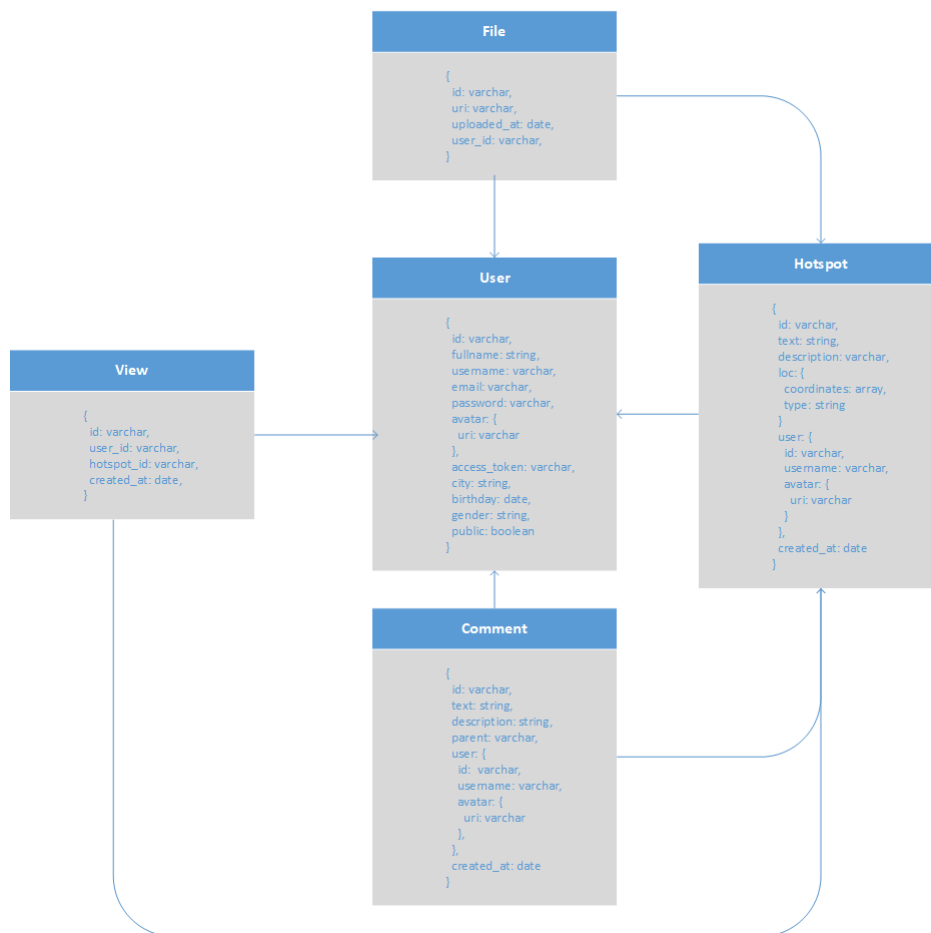
Σχήμα 1.10: Λογική σχεδίασης του συστήματος ελεγκτών στον εξυπηρετητή αιτημάτων της εφαρμογής.

Από τα παραπάνω λοιπόν, γίνεται αντιληπτή η λογική πορεία εξυπηρέτησης αιτημάτων στην οποία βασίζεται ο server. Ωστόσο, κατά την πορεία αυτή γίνεται πρόσβαση σε δεδομένα τα οποία είναι αποθηκευμένα σε έναν ειδικά διαμορφωμένο χώρο. Ο αποθηκευτικός αυτός χώρος αποτελεί τη βάση δεδομένων της εφαρμογής και αποτελεί ίσως το σημαντικότερο κομμάτι του backend. Στην ενότητα που ακολουθεί θα αναλυθούν οι βασικές σχεδιαστικές τεχνικές που διέπουν τη βάση δεδομένων της εφαρμογής, καθώς επίσης και τα μοντέλα στα οποία οργανώνονται τα δεδομένα της εφαρμογής.

### 1.3 Σχεδίαση Μοντέλου Βάσης Δεδομένων

Τα πολλά αιτήματα μεταξύ κλιεντ-σερερ δημιουργούν την ανάγκη για συνεχείς προσπελάσεις της βάσης δεδομένων. Συνεπώς, η δομή που θα ακολουθεί η βάση θα πρέπει να είναι σχεδιασμένη με τέτοιο τρόπο, ώστε να ευνοεί τη συνεχή πρόσβαση σε αυτή. Για να επιτευχθεί αυτό, θα πρέπει η μοντελοποίηση των δεδομένων να γίνεται με όσο το δυνατόν λιγότερες εξαρτήσεις και με τη χρήση απλών δομών για υψηλή απόδοση.

### 1.3.1 Μοντελοποίηση Δεδομένων



Σχήμα 1.11: Σχέδιο μοντελοποίησης των δεδομένων της εφαρμογής.

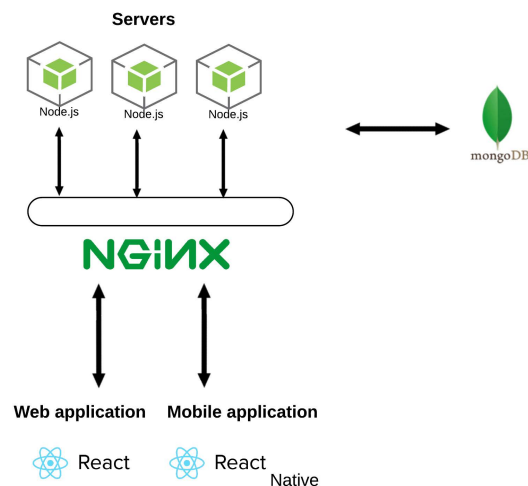
Όπως αναφέρθηκε στην ενότητα 3.3.2, τα δεδομένα της εφαρμογής μοντελοποιούνται σε collections. Η εσωτερική διάρθρωση κάθε τέτοιου μοντέλου περιλαμβάνει ορισμένες ιδιότητες και κάθε μοντέλο που δημιουργείται θα πρέπει να τις ακολουθεί αυστηρά. Κάθε μοντέλο διακρίνεται από τα υπόλοιπα της ίδιας συλλογής μέσω του αναγνωριστικού ID το οποίο προσδίδεται αυτόματα σε αυτό. Επίσης, υπάρχει η πιθανότητα κάποια ιδιότητα να είναι μοναδική για κάθε μοντέλο (πχ. η ηλεκτρονική διεύθυνση στο μοντέλο χρήστη). Αυτό επιτυγχάνεται με τον ορισμό ενός δείκτη μοναδικότητας (uniqueness index) στην συγκεκριμένη ιδιότητα του μοντέλου. Τα νέα μοντέλα αποθηκεύονται στην αντίστοιχη συλλογή για μετέπειτα χρήση από την εφαρμογή. Στο σχ. 1.11 φαίνεται η δομή των μοντέλων της εφαρμογής, καθώς και οι χαρακτηριστικές ιδιότητες του κάθε μοντέλου.

Η αναπαράσταση των δεδομένων με αυτό τον τρόπο, δίνει στον server τη δυνατότητα να προσπελάζει τη βάση δεδομένων με μεγάλη ταχύτητα. Έτσι, η αναζήτηση δεδομένων γίνεται σε μικρό χρονικό διάστημα και η απόδοση της εφαρμογής είναι η βέλτιστη δυνατή. Το ζητούμενο αυτό είναι ακόμα κυριότερα όταν πρόκειται για την σχεδίαση μιας εφαρμογής με πολλούς χρήστες, μιας και δημιουργούνται συνεχώς αιτήματα για πρόσβαση στη βάση δεδομένων. Η



ύπαρξη απλοποιημένων μοντέλων βοηθάει στην γρήγορη εξυπηρέτηση των παραπάνω αιτημάτων και δίνει τη δυνατότητα παραλληλοποίησης των διαδικασιών.

Καθοριστικό ρόλο στην ταχύτητα πρόσβασης στη βάση δεδομένων παίζει και η επιλογή της τεχνολογίας. Για το λόγο αυτό έχει επιλεγεί η MongoDB ως το είδος της βάσης που θα χρησιμοποιηθεί στην υλοποίηση. Η MongoDB επικεντρώνεται στην εξυπηρέτηση πολλαπλών αιτημάτων που απαιτούν ταυτόχρονη πρόσβαση σε δεδομένα της βάσης, αποτελώντας συνεπώς και την καλύτερη λύση (βλ. σχ. 1.12).



Σχήμα 1.12: Φιλοσοφία πίσω από τη σχεδίαση της MongoDB.



## Κεφάλαιο 2

# Υλοποίηση Συστήματος

Μέχρι στιγμής, έχει γίνει αναφορά στις τεχνολογίες και τις τεχνικές σχεδίασης της εφαρμογής. Το κεφάλαιο αυτό επεκτείνεται στην υλοποίηση της εφαρμογής, αναλύοντας τις μεθόδους που χρησιμοποιήθηκαν για τον προγραμματισμό τόσο του frontend, όσο και του backend. Η ενότητα 5.1 αφορά το frontend κομμάτι, δηλαδή με τον προγραμματισμό των διεπιφανειών χρήστη. Η ενότητα 5.2 καταπιάνεται με την υλοποίηση του backend, το οποίο απαρτίζεται από τον εξυπηρετητή αιτημάτων που δημιουργούνται από τις διεπιφάνειες χρήστη και τη βάση δεδομένων στην οποία αποθηκεύονται όλες οι πληροφορίες της εφαρμογής. Στην ενότητα 5.3 παρουσιάζονται τα τεχνολογικά εργαλεία τα οποία χρειάστηκαν στην υλοποίηση. Τέλος, στην ενότητα 5.4 γίνεται μια συνοπτική αναφορά στην πλοτφόρμα προγραμματισμού που επιλέχθηκε για την υλοποίηση της εφαρμογής.

### 2.1 Λεπτομέρειες Υλοποίησης Διεπιφανειών Χρήστη

Ιδιαίτερο ενδιαφέρον παρουσιάζουν οι τεχνικές που χρησιμοποιήθηκαν στην υλοποίηση του συστήματος διεπιφανειών χρήστη. Η εφαρμογή αφορά κινητές συσκευές με λειτουργικό iOS, επομένως δόθηκε μεγάλη βαρύτητα σε μεθόδους που υιοθετούν οι περισσότερες native εφαρμογές της ίδιας κατηγορίας. Κατά τον προγραμματισμό, σχεδόν όλες οι διεπαφές που χρησιμοποιούνται αφορούν αποκλειστικά iOS πλατφόρμες προκειμένου να προσφέρουν την καλύτερη δυνατή εμπειρία στο χρήστη.

Η εφαρμογή αναφέρεται σε κινητά τελευταίας γενιάς. Αυτό εισάγει κάποιους περιορισμούς στον προγραμματισμό, αλλά έχει ταυτοχρόνως και κάποια πλεονεκτήματα. Για παράδειγμα, οι διαστάσεις της οθόνης στις κινητές συσκευές είναι πολύ μικρότερες από αυτές ενός σταθερού υπολογιστή. Διαδραστικά στοιχεία όπως πλήκτρα, πεδία εισόδου, εικονίδια και άλλα γραφικά θα πρέπει να λαμβάνουν υπόψη τους τον παραπάνω περιορισμό. Συνεπώς, ο προγραμματισμός για mobile apps είναι σαφώς πιο δύσκολος από τον προγραμματισμό για desktop apps.

Το κυριότερο πλεονέκτημα του προγραμματισμού για να mobile apps είναι το γεγονός ότι δεν χρειάζεται να δωθεί ιδιαίτερη βάση σε λεπτομέρειες, αφού οι οθόνες των κινητών δεν επιτρέπουν την σχολαστική ενασχόληση με κάθε στοιχείο ή την υπερβολική λεπτομέρεια των διαδραστικών στοιχείων, αφού αυτό θα καθιστούσε την εμπειρία χρήσης της εφαρμογής

κουραστική για το χρήστη. Έτσι, ο προγραμματισμός επικεντρώνεται κυρίως γύρω από το κομμάτι με το οποίο θα αλληλεπιδρά ο χρήστης και τις τεχνικές λεπτομέρειες που συμπεριλαμβάνει. Για να εξασφαλιστεί μια ολοκληρωμένη, αλλά συγχρόνως απλή εμπειρία εντός της εφαρμογής, όλες οι διεπιφάνειες ακολουθούν αυτή την τακτική, όπου το διαδραστικό περιβάλλον αποτελείται αποκλειστικά από διεπαφές που είναι ζωτικές για την σωστή λειτουργία της κάθε οθόνης.

### **2.1.1 Βασικά Προγραμματιστικά Χαρακτηριστικά Οθονών**

Μιλώ για τα τμήματα ζωτικής σημασίας που έχουν οι περισσότερες οθόνες..

#### **2.1.1.1 Τεχνική Πλοήγησης στην Εφαρμογή**

#### **2.1.1.2 Επικεφαλίδα Οθόνης**

#### **2.1.1.3 Κεντρικό Μενού Οθονών**

#### **2.1.1.4 Βασικά Πλήκτρα Ενεργειών**

### **2.1.2 Λειτουργικότητα Οθονών Εφαρμογής**

#### **2.1.2.1 WelcomeScreen**

#### **2.1.2.2 RegisterScreen**

#### **2.1.2.3 LoginScreen**

#### **2.1.2.4 HomeScreen**

#### **2.1.2.5 CreateHotspotScreen**

#### **2.1.2.6 HotspotListScreen**

#### **2.1.2.7 CommentScreen**

#### **2.1.2.8 ProfileScreen**

### **2.1.3 Τεχνικές Ενιαίας Διαχείρισης Δεδομένων**

#### **2.1.3.1 React Redux**

#### **2.1.3.2 Redux Thunk Middleware**

### **2.1.4 Βιβλιοθήκες και Βοηθητικά Προγραμματιστικά Πακέτα**

### **2.1.5 Ιεράρχηση Αρχείων και Δομή Φακέλων**

## **2.2 Σύστημα αξιολόγησης**

Εδώ περιγράφουμε το σύστημα που χρησιμοποιήσαμε για να αξιολογήσουμε τις τεχνικές μας. Συνήθως, η περιγραφή γίνεται με κείμενο και με ένα βλοκ διαγράμμι περιγραφής των

φίγυρες/γριδ"γρανυλαριτψ-επς-ζονερτεδ-το.πδφ

Σχήμα 2.1: Κλιμάκωση χρόνου εκτέλεσης για διάφορες υποδιαιρέσεις του καννάβου

λειτουργιών του συστήματος. Αν το σύστημα είναι μεγάλο, τότε συζητήστε με τον επιβλέποντα μήπως χρειάζεται να υπάρχει ξεχωριστό κεφάλαιο με τίτλο ' Σχεδίαση συστήματος '.

## 2.3 Οργάνωση πειραμάτων

Εδώ περιγράφουμε λεπτομερώς πώς οργανώσαμε τα πειράματα. Π.χ. α) τί σύνολο δεδομένων χρησιμοποιήσαμε (συνθετικά, έτοιμες συλλογές) β) τί τιμές είχαν διάφοροι παράμετροι του συστήματός αξιολόγησης, κ.λ.π.

Οι τιμές των παραμέτρων μπορούν να φαίνονται και σε πίνακα, όπως λ.χ. στον Πίνακα 2.1:

Πλήθος κελιών καννάβου $c \times c$	50 × 50, 100 × 100, 200 × 200, <b>250 × 250</b> , 500 × 500, 1000 × 1000
Τυπική απόκλιση $\sigma$	25m, 50m, 75m, <b>100m</b> , 150m, 200m
Αριθμός εγγύτερων γειτόνων $k$	1, 2, <b>3</b> , 4, 5, 10, 20
Πιθανοτικό κατώφλι $\theta$	50%, 60%, 70%, <b>75%</b> , 80%, 90%, 99%

Πίνακας 2.1: Παράμετροι πειραμάτων

Αν χρησιμοποιήσατε συνθετικά δεδομένα, τότε εξηγήστε στην παρακάτω χωριστή υποενότητα τον τρόπο που τα δημιουργήσατε:

### 2.3.1 Παραγωγή συνθετικών δεδομένων

Τα πειραματικά δεδομένα παρήχθησαν ...

## 2.4 Αποτελέσματα της μελέτης

Εδώ παρουσιάζουμε τα αποτελέσματα των μετρήσεων με μορφή γραφικών παραστάσεων, όπως ενδεικτικά στο Σχ. 2.1. Δίνουμε λεπτομερή εξήγηση και σχολιασμό των αποτελεσμάτων, πάντα σε σχέση με το πρόβλημα που οι τεχνικές μας φιλοδοξούν να λύσουν. Φροντίστε να ομαδοποιήσετε τα αποτελέσματα (σε χωριστές υποενότητες) ανάλογα με τις παραμέτρους που μετράτε, π.χ. χωριστά το κόστος σε χρόνο από το κόστος σε χώρο ή όσον αφορά την ακρίβεια των απαντήσεων.

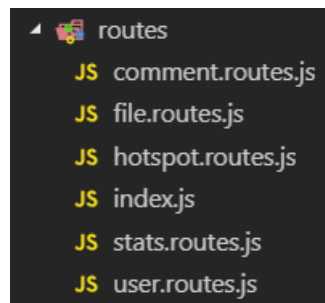
## 2.5 Σύνοψη συμπερασμάτων αξιολόγησης

Εδώ συνοψίζουμε τα συμπεράσματα της αξιολόγησης. Η σύνοψη να γίνεται σύντομα και καθαρά, π.χ. 1. αυτό, 2. το άλλο, κ.ο.κ.

### 2.5.0.1 Server Routes

Ο server χρησιμοποιεί ένα σύστημα δρομολόγησης (routes) των αιτημάτων του client που προορίζονται προς εξυπηρέτηση. Κάθε ομάδα υπηρεσιών που απευθύνονται σε μια συγκεκριμένη λειτουργία της εφαρμογής αντιστοιχίζεται σε ένα δρομολογητή (router). Ο router απαρτίζεται από επιμέρους routes, καθεμία εκ των οποίων εξυπηρετεί μια συγκεκριμένη λειτουργία. Έτσι, στο παραπάνω παράδειγμα, το αίτημα για τη δημιουργία νέου hotspot θα δρομολογηθεί για εξυπηρέτηση από τον router που είναι υπεύθυνος για τις ενέργειες που αφορούν ένα hotspot (*HotspotRoutes*).

Το σύστημα των δρομολογητών έχει σχεδιαστεί με στόχο την απλότητα και την αμεσότητα στην κατανόηση από τον προγραμματιστή (βλ. Σχ. 2.2). Αιτήματα που αφορούν τους χρήστες δρομολογούνται προς εξυπηρέτηση από τον router που είναι υπεύθυνος για τις ενέργειες που αφορούν τους χρήστες. Με τον ίδιο τρόπο, Αιτήματα που σχετίζονται με σχόλια χρηστών σε ένα hotspot, εξυπηρετούνται από εκείνο τον router που είναι υπεύθυνος για τα σχόλια.



Σχήμα 2.2: Ιεράρχηση του συστήματος δρομολογητών στην εφαρμογή.

Οι δρομολογητές του συστήματος είναι οι ακόλουθοι:

- **Hotspot Routes** – περιλαμβάνει routes που αφορούν την εξυπηρέτηση αιτημάτων σχετικά με hotspots, όπως δημιουργία, επεξεργασία, διαγραφή, προβολή, ανάκτηση κλπ.
- **User Routes** – περιλαμβάνει routes που αφορούν την εξυπηρέτηση αιτημάτων σχετικά με χρήστες, όπως εγγραφή, σύνδεση, επεξεργασία, προβολή, ανάκτηση κλπ.
- **Comment Routes** – περιλαμβάνει routes που αφορούν την εξυπηρέτηση αιτημάτων σχετικά με σχόλια, όπως δημιουργία, προβολή, απαρίθμηση κλπ.
- **File Routes** – περιλαμβάνει routes που αφορούν την εξυπηρέτηση αιτημάτων σχετικά με συνημμένα αρχεία σε hotspots.
- **Stats Routes** – περιλαμβάνει routes που αφορούν την εξυπηρέτηση αιτημάτων σχετικά με στατιστικά δεδομένα.

### 2.5.0.2 Server Controllers

Αφότου τα αιτήματα δρομολογηθούν προς εξυπηρέτηση στους αντίστοιχους routers, ακολουθεί η επεξεργασία τους και η εκτέλεσή τους μέσω κατάλληλων μεθόδων. Οι μέθοδοι που

αναλαμβάνουν την εκτέλεση των αιτημάτων ονομάζονται ελεγκτές (controllers) και αποτελούνται από ένα σύνολο ασύγχρονων εντολών και κλήσεων μεταξύ της βάσης δεδομένων και του server.

Όπως και με το σύστημα των δρομολογητών, έτσι και οι ελεγκτές έχουν σχεδιαστεί με στόχο να αποσυμπλέκουν τις ενέργειες που σχετίζονται με κάθε αίτημα (βλ. Σχ. ;;). Όλες οι ενέργειες που αφορούν τα hotspots εκτελούνται αποκλειστικά από έναν ελεγκτή. Το ίδιο συμβαίνει και με τις ενέργειες γύρω από τους χρήστες. Το σύστημα των controllers έχει παραπλήσια μορφή με αυτό των routers καθώς στόχος της σχεδίασης είναι να υπάρχει ένας βαθμός αναλογίας μεταξύ των διαφόρων τμημάτων, ο οποίος μειώνει αισθητά το χρόνο κατανόησης και βελτιστοποιεί τη διαδικασία υλοποίησης:

- **Hotspot Controller** – περιλαμβάνει τις μεθόδους που αφορούν την εξυπηρέτηση αιτημάτων σχετικά με hotspots, όπως δημιουργία, επεξεργασία, διαγραφή, προβολή, ανάκτηση κλπ.
- **User Controller** – περιλαμβάνει τις μεθόδους που αφορούν την εξυπηρέτηση αιτημάτων σχετικά με χρήστες, όπως εγγραφή, σύνδεση, επεξεργασία, προβολή, ανάκτηση κλπ.
- **Comment Controller** – περιλαμβάνει τις μεθόδους που αφορούν την εξυπηρέτηση αιτημάτων σχετικά με σχόλια, όπως δημιουργία, προβολή, απαρίθμηση κλπ.
- **File Controller** – περιλαμβάνει τις μεθόδους που αφορούν την εξυπηρέτηση αιτημάτων σχετικά με συνημμένα αρχεία σε hotspots.
- **Stats Controller** – περιλαμβάνει τις μεθόδους που αφορούν την εξυπηρέτηση αιτημάτων σχετικά με στατιστικά δεδομένα.
- **Views Controller** – περιλαμβάνει τις μεθόδους που αφορούν την εξυπηρέτηση αιτημάτων σχετικά με τα views ενός hotspot.





## Κεφάλαιο 3

# Έλεγχος

Εδώ λέμε ότι θα ακολουθήσουν τεχνικές λεπτομέρειες της διπλωματικής.

### 3.1 Λεπτομέρειες υλοποίησης

Εδώ περιγράφουμε λεπτομερώς θέματα της διπλωματικής που έχουν τεχνικό ενδιαφέρον. Προσδιορίστε επομένως τα θέματα αυτά, βάλτε μια ενότητα για κάθε ένα και περιγράψτε τα αναλυτικά. Η περιγραφή μπορεί να γίνει βάζοντας κομμάτια κώδικα ή ψευδοκώδικα, και περιγράφοντάς τα με λόγια. Μην ξεχνάτε να δίνετε πάντα παραδείγματα για το πώς τρέχει ένα κομμάτι κώδικα π.χ. για έναν αλγόριθμο.

#### 3.1.1 ‘Τίτλος θέματος 1’

Γράψτε το κείμενό σας εδώ ...

#### 3.1.2 ‘Τίτλος θέματος 1’

Γράψτε το κείμενό σας εδώ ...

### 3.2 Πλατφόρμες και προγραμματιστικά εργαλεία

Εδώ περιγράφονται τα χαρακτηριστικά της συγκεκριμένης υλοποίησης, όπως η πλατφόρμα ανάπτυξης και εκτέλεσης, τα προγραμματιστικά εργαλεία, οι απαιτήσεις της εφαρμογής σε hardware, κ.λ.π. Επίσης, περιγράφεται λεπτομερώς η διαδικασία εγκατάστασης της διπλωματικής σε υπολογιστή. Προσέξτε να δίνονται όλες οι λεπτομέρειες, το απαραίτητο λογισμικό και οι αναγκαίες ρυθμίσεις.



## Κεφάλαιο 4

# Επίλογος

Εδώ εξηγούμε ότι θα συνοψίσουμε την μελέτη που εκπονήθηκε στα πλαίσια της διπλωματικής.

### 4.1 Σύνοψη και συμπεράσματα

Εδώ συνοψίζουμε τα αποτελέσματα της διπλωματικής και περιγράφουμε τα συμπεράσματα που προέκυψαν, αρνητικά και θετικά. Επιβεβαιώνουμε την συνεισφορά της διπλωματικής στα προβλήματα που αναφέραμε στην εισαγωγή.

### 4.2 Μελλοντικές επεκτάσεις

Εδώ δίνουμε ιδέες για επέκταση της διπλωματικής.



# Βιβλιογραφία

- [1] Jeff Sondermann, *American Press Institute*. Mobile and social media are intricately linked. <https://www.americanpressinstitute.org/publications/reports/white-papers/mobile-and-social-media/> 2014. Last accessed on 07/03/2019.
- [2] Ricardo Bilton. New data shows just how much social sharing has decreased since 2015 (and News Feed tweaks are just one factor). <https://www.venturelean.team/hello-world/> 2018. Last accessed on 08/03/2019.
- [3] Jessica Brown, *BBC Future*. Is social media bad for you? The evidence and the unknowns. <http://www.bbc.com/future/story/20180104-is-social-media-bad-for-you-the-evidence-and-the-unknowns> 2018. Last accessed on 08/03/2019.
- [4] CrowdSourcingWeek. What is Crowdsourcing? <https://crowdsourcingweek.com/what-is-crowdsourcing/> 2018. Last accessed on 08/03/2019.
- [5] ISLAB Team, NTUA. About WITH <http://withcrowd.eu/about> 2018. Last accessed on 08/03/2019.
- [6] Foursquare Developer Team. Foursquare API <https://developer.foursquare.com/docs/api/endpoints> 2018. Last accessed on 08/03/2019.
- [7] Grace Fearon, *iStudent*. Our need to maintain social approval is actually making us lose what is best about ourselves - our individuality. <https://www.independent.co.uk/student/istudents/our-need-to-maintain-social-approval-is-actually-making-us-lose-what-is-best-about-ourselves-our-individuality-1818181.html> 2016. Last accessed on 08/03/2019.
- [8] Mansi Beniwal. Social Media and its Impact in Interpersonal Relationships. <https://jarvee.com/social-media-impact-interpersonal-relationships/> 2018. Last accessed on 08/03/2019.
- [9] Ullah, Asmat. Client Side Scripting for Web Applications [https://www.sqa.org.uk/e-learning/SiteHomeCD/page\\_26.htm](https://www.sqa.org.uk/e-learning/SiteHomeCD/page_26.htm) 2007. Last accessed on 08/03/2019.

- [10] Stroustrup, Bjarne. Bjarne Stroustrup's FAQ: What do you think of C++/CLI? [http://www.stroustrup.com/bs\\_faq.html#CppCLI](http://www.stroustrup.com/bs_faq.html#CppCLI) 2009. Last accessed on 08/03/2019.
- [11] Gregory, Kate. Managed, Unmanaged, Native: What Kind of Code Is This? <https://www.developer.com/net/cplusplus/article.php/2197621> 2003. Last accessed on 08/03/2019.
- [12] Meijer, Erik and Peter Drayton. Static Typing Where Possible, Dynamic Typing When Needed: The End of the Cold War Between Programming Languages. Microsoft Corporation, 2005. Last accessed on 08/03/2019.
- [13] L. Tratt. Dynamically Typed Languages. *Advances in Computers*, 77(2):149–184, July 2009. Last accessed on 08/03/2019.
- [14] Apple Inc., *swift.org*. The Swift Linux Port <https://swift.org/blog/swift-linux-port/> 2016. Last accessed on 10/03/2019.
- [15] Chris Lattner. Chris Lattner's Homepage <http://nondot.org/sabre/> 2014. Last accessed on 10/03/2019.
- [16] Apple Worldwide Developers Conference, Session 102. Platforms State of the Union 2016. Last accessed on 10/03/2019.
- [17] Rachel Metz, *MIT Technology*. Apple Seeks a Swift Way to Lure More Developers <https://www.technologyreview.com/s/527821/apple-seeks-a-swift-way-to-lure-more-developers/> 2014. Last accessed on 10/03/2019.
- [18] Harrison Weber, *VentureBeat*. Apple announces 'Swift', a new programming language for macOS iOS <https://venturebeat.com/2014/06/02/apple-introduces-a-new-programming-language-swift-objective-c-without-the-c/> 2014. Last accessed on 10/03/2019.
- [19] J. Gosling, J. Bill, G. Steele, G. Bracha, A. Buckley. The Java® Language Specification *Java SE 8th edition*, 2014. Last accessed on 11/03/2019.
- [20] Computer Weekly. Write once, run anywhere? <http://www.computerweekly.com/Articles/2002/05/02/186793/write-once-run-anywhere.htm> 2002. Last accessed on 11/03/2019.
- [21] Oracle Inc.. Design Goals of the Java™ Programming Language <https://www.oracle.com/technetwork/java/intro-141325.html> 2013. Last accessed on 11/03/2019 .
- [22] R. McMillan, *wired.com*. Is Java Losing Its Mojo? <https://www.wired.com/2013/01/java-no-longer-a-favorite/> 2013. Last accessed on 11/03/2019.

- [23] Stephen O’Grady, *RedMonk*. The RedMonk Programming Language Rankings: January 2015 <https://redmonk.com/sograd/2015/01/14/language-rankings-1-15/> 2015. Last accessed on 11/03/2019.
- [24] *langpop.com*. Programming Language Popularity <https://web.archive.org/web/20090116080326/http://www.langpop.com/> 2009. Last accessed on 11/03/2019.
- [25] Tata McGraw. Object-oriented Programming with Java: Essentials and Applications. *Hill Education*, 1:30–35, 2009. Last accessed on 11/03/2019.
- [26] Sun Microsystems, *WaybackMachine*. JAVASOFT SHIPS JAVA 1.0 - *Programming environment available free for developers* <https://web.archive.org/web/20070310235103/http://www.sun.com/smi/Press/sunflash/1996-01/sunflash.960123.10561.xml> 2018. Last accessed on 11/03/2019.
- [27] Alex Mullis, *Android Authority*. How to install the Android SDK (Software Development Kit) <https://www.androidauthority.com/how-to-install-android-sdk-software-development-kit-21137/> 2016. Last accessed on 11/03/2019.
- [28] Android Developers. Introduction to Android <https://developer.android.com/guide/index.html> 2017. Last accessed on 11/03/2019.
- [29] Android Developers. Tools Overview <https://developer.android.com/studio/command-line/> 2012. Last accessed on 11/03/2019.
- [30] Stoyan Stefanov. JavaScript Patterns. *O’Reilly Media, Inc.*, 1(1): 5–10, 2010. Last accessed on 14/03/2019.
- [31] *w3techs.com*. Usage Statistics of JavaScript for Websites <https://w3techs.com/technologies/details/cp-javascript/all/all> 2015. Last accessed on 14/03/2019.
- [32] *pabbly.com*. NodeJS Event Loops <https://www.pabbly.com/tutorials/node-js-event-loops/>. 2015. Λαστ αςςεσσεδ ον 14/03/2019.
- [33] Jesse James Garrett, *AdaptivePath*. Ajax: A New Approach to Web Applications <https://adaptivepath.org/ideas/ajax-new-approach-web-applications/> 2005. Last accessed on 14/03/2019.
- [34] Mozilla Developer Network Web Docs, *MDN.com*. Ajax - Web developer guides <https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX> 2018. Last accessed on 14/03/2019.
- [35] David Flanagan. JavaScript - The Definitive Guide. *O’Reilly, Sebastpool, CA*, 1(5):495–450, 2006.

- 
- [36] Steve Burbeck. Applications Programming in Smalltalk-80:How to use Model-View-Controller (MVC) <https://web.archive.org/web/20120729161926/http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html> 1992. Last accessed on 14/03/2019.
- [37] Facebook, *reactjs.org*. Components and Props <https://reactjs.org/docs/components-and-props.html#props-are-read-only> 2018. Last accessed on 17/03/2019.
- [38] React Blog, *reactjs.org*. Refs and DOM <https://reactjs.org/docs/refs-and-the-dom.html> 2018. Last accessed on 17/03/2019.
- [39] Facebook. JSX Specification <https://facebook.github.io/jsx/> 2018. Last accessed on 17/03/2019.
- [40] *reactjs.org*, tutorials. Tutorial: intro to React <https://reactjs.org/tutorial/tutorial.html> 2017. Last accessed on 17/03/2019.
- [41] Bonnie Eisenman. Learning React Native. *O'Reilly Media, Inc.*, chapter 1, 2015. Last accessed on 18/03/2019.
- [42] Tal Kol, *Wix.com*. Building a React Native App for 80 Million Users. From *ReactNext Conference 2016, Tel Aviv*, <https://www.youtube.com/watch?v=abSNo2P9mMM&feature=youtu.be&t=17m34s>, minutes 17:37–33:20, 2016. Last accessed on 18/03/2019.
- [43] Natalia Chrzanowska. 13 Great Examples of React Native Apps <https://www.netguru.com/blog/13-great-apps-written-with-react-native> 2019. Last accessed on 18/03/2019.
- [44] Marvin Frachet, *hackernoon.com*. Understanding the React Native bridge concept <https://hackernoon.com/understanding-react-native-bridge-concept-e9526066ddb8> 2018. Last accessed on 18/03/2019.
- [45] Tomislav Capan, *toptal.com*. Why Use Node.js? <https://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js> 2013. Last accessed on 19/03/2019.
- [46] Kenneth Peeples, *dzone.com*. What are the Benefits of Node.js? <https://dzone.com/articles/what-are-benefits-nodejs> 2015. Last accessed on 19/03/2019.
- [47] Node.js Foundation, *nodejs.org*. Node.js Docs <https://nodejs.org>. Last accessed on 19/03/2019.



- [48] Expolarions of an Engineer, *timcosta.io*. The Node.js Event Loop <https://www.timcosta.io/the-node-js-event-loop/> 2016. Last accessed on 19/03/2019.
- [49] RESTful Web Services Architecture. Of the *World Wide Web Consortium*, Chapter 3.1.3: Relationship to the World Wide Web and REST Architectures, February 2004. Last accessed on 19/03/2019.
- [50] R. T. Fielding Architectural Styles and the Design of Network-based Software Architectures (Ph.D.). *Chapter 5: Representational State Transfer (REST)*. University of California, Irvine. 2000. Last accessed on 19/03/2019.
- [51] L. Richardson, M. Amundsen RESTful Web APIs, *O'Reilly Media*. 2013 ISBN 978-1-449-35806-8.
- [52] Davis Kerby Why MongoDB is the Way to Go, *DZone*. <https://dzone.com/articles/why-mongodb-is-worth-choosing-find-reasons> 2015. Last accessed on 20/03/2019.
- [53] *ClusterHQ*. Ridiculously fast MongoDB replica recovery Part 1 of 2 <http://clusterhq.com/2016/03/14/ridiculously-fast-mongodb-replica-recovery-with-flocker/>. Last accessed on 20/03/2019.
- [54] *Severalnines*. Turning MongoDB Replica Set to a Sharded Cluster <https://severalnines.com/blog/turning-mongodb-replica-set-sharded-cluster> 2013. Last accessed on 20/03/2019.
- [55] *Compose*. GridFS & MongoDB: Pros And Cons <https://www.compose.com/articles/gridfs-and-mongodb-pros-and-cons/> 2014. Last accessed on 20/03/2019.
- [56] Md. Malick, *ExpertsTown*. MongoDB Overview [https://www.hugedomains.com/domain\\_profile.cfm?d=expertstown&e=com](https://www.hugedomains.com/domain_profile.cfm?d=expertstown&e=com) 2013. Last accessed on 20/03/2019.
- [57] MongoDB Docs, *docs.mongodb.com*. Aggregation — MongoDB Manual <https://docs.mongodb.com/manual/aggregation/>. Last accessed on 20/03/2019.
- [58] MongoDB Docs, *docs.mongodb.com*. Map-Reduce — MongoDB Manual <https://docs.mongodb.com/manual/core/map-reduce/>. Last accessed on 20/03/2019.
- [59] J. Bradley, N. Sakimura, M. Jones, JSON Web Token (JWT) *Mastering Identity and Access Management with Microsoft Azzure*, 1(1):84, ISBN: 9781785887888.
- [60] N. Dalvi, C. Ré, and D. Suciu. Probabilistic databases: Diamonds in the dirt. *Communications of the ACM*, 52(7):86–94, July 2009.

- [61] R. H. Güting, M. H. Böhlen, M. Erwig, C. S. Jensen, N. A. Lorentzos, M. Schneider, and M. Vazirgiannis. A foundation for representing and querying moving objects. *ACM Transactions on Database Systems (TODS)*, 25(1):1–42, 2000.
- [62] N. Jain, S. Mishra, A. Srinivasan, J. Gehrke, J. Widom, H. Balakrishnan, U. Çetintemel, M. Cherniack, R. Tibbetts, and S. Zdonik. Towards a streaming SQL standard. In *Proceedings of the VLDB Endowment*, 1(2):1379–1390, 2008.
- [63] K. Mouratidis, D. Papadias, and M. Hadjieleftheriou. Conceptual partitioning: an efficient method for continuous nearest neighbor monitoring. In *Proceedings of the 24th ACM SIGMOD International Conference on Management of Data*, pages 634–645, 2005.
- [64] Oracle, Inc. Complex event processing CQL language reference. [http://docs.oracle.com/cd/E16764\\_01/doc.1111/e12048/intro.htm](http://docs.oracle.com/cd/E16764_01/doc.1111/e12048/intro.htm) 2009. Last accessed on 15/09/2013.
- [65] K. Patroumpas and T. Sellis. Subsuming multiple sliding windows for shared stream computation. In *Advances in Databases and Information Systems*, volume 6909 of Springer *Lecture Notes in Computer Science*, pages 56–69, 2011.
- [66] P. Rigaux, M. Scholl, and A. Voisard. *Spatial databases: with application to GIS*. Morgan Kaufmann, 2001.
- [67] Σεραφείμ Παπαδιάς. Απόρριψη φόρτου από ρεύματα τροχιάς κινούμενων αντικειμένων. Διπλωματική Εργασία *DIPL-2015-02* στο Εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων, Εθνικό Μετσόβιο Πολυτεχνείο, Ιούλιος 2015.

# Κατάλογος Ακρονύμων

<b>IT</b>	Information Technology
<b>NGO</b>	Non Governmental Organization
<b>API</b>	Application programming Interface
<b>URL</b>	Uniform Resource Locator
<b>SDK</b>	Software Development Kit
<b>CLR</b>	Common Language Runtime
<b>CIL</b>	Common Intermediate Language
<b>OS</b>	Operating System
<b>IDE</b>	Integrated Development Environment
<b>WWDC</b>	Worldwide Development Conference
<b>OO</b>	Object Oriented
<b>WORA</b>	Write Once Run Anywhere
<b>JS</b>	JavaScript
<b>ES</b>	EcmaScript
<b>HTML</b>	Hypertext Markup Language
<b>CSS</b>	Cascading Style Sheets
<b>DOM</b>	Document Object Model
<b>AJAX</b>	Asynchronous Javascript And XML
<b>I/O</b>	Input Output
<b>SPA</b>	Single Page Application
<b>MVC</b>	Model View Controller

[illegible]

# Γλωσσάριο

## Αγγλικός όρος

social media

facebook

instagram

snapchat

twitter

tinder

New York Times

crowdsourcing

withcrowd

portal

repository

youtube

foursquare

endpoint

web application

native application

programming language

managed

compiled

dynamic

native

compiler

high-level

overhead

low-level

cross-platform

## Ελληνικός όρος

μέσα κοινωνικής δικτύωσης

μία από τις δημοφιλέστερες εταιρείες και υπηρεσίες κοινωνικής δικτύωσης σήμερα

υπηρεσία κοινωνικής δικτύωσης για κοινή χρήση φωτογραφιών και βίντεο

εφαρμογή ανταλλαγής μηνυμάτων πολυμέσων αμερικανική ηλεκτρονική υπηρεσία ειδήσεων και κοινωνικής δικτύωσης

εφαρμογή για κοινωνική αναζήτηση βασισμένη σε τοποθεσίες

αμερικανική εφημερίδα

πληθοπορισμός

ψηφιακή πλατφόρμα αρχειοθέτησης ευρυμάτων της ευρωπαϊκής πολιτισμικής κληρονομιάς

πύλη

ψηφιακή αποθήκη

εφαρμογή για κοινή χρήση βίντεο

εφαρμογή τοπικής αναζήτησης κοινωνικών τόπων τελικό σημείο

εφαρμογή που τρέχει σε φυλλομετρητή

μητρική εφαρμογή

γλώσσα προγραμματισμού

διαχειριζόμενη

μεταγλωττισμένη

δυναμική

μητρική

μεταγλωττιστής

υψηλού επιπέδου

χρόνος καθυστέρησης

χαμηλού επιπέδου

μεταξύ πολλών πλατφόρμων

bytecode	κώδικας ψηφίων
ios	λειτουργικό σύστημα που ανήκει στην Apple Inc.
android	λειτουργικό σύστημα που ανήκει στην Google Inc.
developer	προγραμματιστής
concurrent	ταυτόχρονος
class	κλάση
web browser	φυλλομετρητής ιστού
web page	ιστοσελίδα
object	αντικείμενο
integer	ακέραιος αριθμός
floating point	μεταβλητή κινητής υποδιαστολής
boolean value	μεταβλητή λογικής τιμής
smartphone	έξυπνη κινητή συσκευή
tablet	φορητή συσκευή
debugger	διορθωτής σφαλμάτων
emulator	προσομοιωτής
tablet	φορητή συσκευή
functional	συναρτησιακός
event-driven	οδηγούμενο από γεγονότα
client-side	πλευρά πελάτη
object-oriented	αντικειμενοστραφής
fade out/in	ξεθώριασμα
slide out/in	ολίσθηση
script-objects	αντικείμενα κώδικα
host-objects	αντικείμενα φιλοξενίας
server	εξυπηρετητής/διακομιστής
requests	αιτήματα
event loop	βρόχος συμβάντων
non-blocking execution	εκτέλεση χωρίς μπλοκάρισμα
callback function	συνάρτηση ανάκλησης
callback content	περιεχόμενο ανάκλησης
run to completion logic	λογική πλήρους διεκπαιρέωσης
browser	περιηγητής
backend	η πλευρά του εξυπηρετητή
frontend	η πλευρά του πελάτη
desktop application	εφαρμογή επιφάνειας εργασίας
model	μοντέλο
view	όψη
controller	ελεγκτής
container class	κλάση-δοχείο
presentation logic	λογική παρουσίασης

mobile application	εφαρμογή για φορητή συσκευή
component	στοιχείο
component-based architecture	αρχιτεκτονική βασισμένη σε στοιχεία
property	ιδιότητα
one-way data binding	μονόδρομη ροή δεδομένων
state	κατάσταση
stateful component	στοιχείο με κατάσταση
child-component	στοιχείο-παιδί
parent-component	στοιχείο-πατέρας
hook	συναρτήσεις-γάντζοι
lifecycle method	συνάρτηση που εκτελείται σε κάποιο συγκεκριμένο χρονικό σημείο κατά τον κύκλο ζωής της εφαρμογής
framework	πλαίσιο
developer experience	προγραμματιστική εμπειρία
rebuild	επαναμεταγλώττιση
design	σχεδίαση
Chrome	περιηγητής της Google
Safari	περιηγητής της Apple
text editor	επεξεργαστής κειμένου
code-reuse	επαναχρησιμότητα κώδικα
knowledge sharing	κοινή χρήση γνώσης
resource saving	εξοικονόμηση πόρων
airbnb	ιδιωτική ηλεκτρονική εταιρεία που προσφέρει υπηρεσίες διαμονής και φιλοξενίας
skype	εφαρμογή τηλεπικοινωνίας
bridge	γέφυρα
realm	βασιλείο, σφαίρα, εδώ: κατηγορία
interoperable language	διαλειτουργική γλώσσα
node	κόμβος
event queue	ουρά συμβάντων
worker-thread	νήμα-εργάτης
worker-thread pool	δεξαμενή νημάτων εργασίας
interpreting	μεταγλώττιση
repersentational state transfer	παραστατική μεταβίβαση κατάστασης
router	δρομολογητής
route	διαδρομή
response payload	φορτίο απόκρισης
key	κλειδί
value	τιμή
string	συμβολοσειρά
boolean	λογική τιμή

array	πίνακας
null	κενός χαρακτήρας
non-relational	μη-σχετικιστικός
document-oriented	εγγραφοστραφής
schema	μοντέλο εγγράφου σε MongoDB
collection	συλλογή
ad hoc query	αναζήτηση συγκεκριμένου σκοπού
field search	αναζήτηση με βάση πεδίο
range search	αναζήτηση με βάση συνθήκη
regular expression search	αναζήτηση με βάση μοτίβο
replication	αναπαραγωγή/αντιγραφή
set	σύνολο, ομάδα
indexing	δεικτοδότηση
load balancing	εξισορρόπηση φορτίου
sharding	κοπή, τεμαχιοποίηση
hashing	διαδικασία κρυπτογράφησης δεδομένων
file storage	αποθήκευση αρχείων
aggregation	άθροιση
pipeline	αγωγός
map-reduce function	μέθοδος ανθροιστικής χαρτογράφησης
single-purpose aggregation	άθροιση μονού σκοπού
pagination	σελιδοποίηση
open authentication	σύστημα ταυτοποίησης ανοικτού τύπου
access token	σύμβολο πιστοποίησης
username	όνομα χρήστη
email	ηλεκτρονική διεύθυνση
password	κωδικός πρόσβασης
refresh token	σύμβολο ανανέωσης
register/signup form	φόρμα εγγραφής
handshake	χειραψία, διαδικασία ανταλλαγής
logout	αποσύνδεση
login/signin form	φόρμα σύνδεσης
authorization header	επικεφαλίδα εξουσιοδότησης
passport strategy	στρατηγική ταυτοποίησης χρηστών
boilerplate	βασική δομή κώδικα
profile	προφίλ χρήστη
hotspot	σημείο ενδιαφέροντος στην εφαρμογή
comment	σχόλιο
point of interest	σημείο ενδιαφέροντος
performance	απόδοση
recoverability	ανάκτηση



[illegible]



