

Ατομική Διπλωματική Εργασία

**ΕΞΑΓΩΓΗ ΠΡΟΦΙΛ ΜΗΧΑΝΙΚΩΝ ΛΟΓΙΣΜΙΚΟΥ ΑΠΟ
ΠΛΗΡΟΦΟΡΙΕΣ ΣΕ ΚΟΙΝΩΝΙΚΑ ΔΙΚΤΥΑ**

ΣΤΑΥΡΟΣ ΓΕΩΡΓΙΟΥ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ιανουάριος 2020

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Εξαγωγή προφίλ μηχανικών λογισμικού από πληροφορίες
σε κοινωνικά δίκτυα**

Σταύρος Γεωργίου

Επιβλέπουσα Καθηγήτρια

Δρ. Γεωργία Καπιτσάκη

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Ιανουάριος 2020

Ευχαριστίες

Με την διεκπεραίωση της διπλωματικής μου εργασίας, θα ήθελα να εκφράσω τις ευχαριστίες μου στην υπεύθυνη και επιβλέπουσα καθηγήτρια μου Δρ. Γεωργία Καπιτσάκη για την ευκαιρία που μου έδωσε να αναπτύξω το συγκεκριμένο θέμα αλλά και την στήριξη της στην υλοποίηση της.

Επίσης θα ήθελα να ευχαριστήσω τους καθηγητές για τις γνώσεις που μου έδωσαν αυτά τα χρόνια στο πανεπιστήμιο.

Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου που με στήριξε σε όλη την προσπάθεια που έκανα αλλά και τους φίλους και συμφοιτητές μου που ήταν δίπλα μου καθ' όλη τη διάρκεια των σπουδών μου.

Περίληψη

Η εφαρμογή «Expertise Analyzer Mobile» που υλοποίησα στα πλαίσια της πτυχιακής μου εργασίας αποτελεί ένα εργαλείο τόσο για έναν προγραμματιστή, ο οποίος μπορεί να εξάγει το προφίλ του σε έναν εργοδότη, αλλά και για ένα εργοδότη ο οποίος θα μπορεί να εντοπίσει εργαζόμενους.

Για την άντληση των πληροφοριών έχω χρησιμοποιήσει δύο πλατφόρμες, το δίκτυο ερωτοαπαντήσεων Stack Exchange το οποίο αποτελείτε από 170 ιστοσελίδες και το GitHub το οποίο παρέχει ένα κατακευματισμένο σύστημα ελέγχου.

Υπάρχει η ανάγκη δικτύωσης των μηχανικών λογισμικού και προβολής της δουλειάς τους αλλά και της γνώσης που κατέχουν σχετικά με διάφορα προγραμματιστικά πεδία. Έτσι μέσα από τη προβολή των προφίλ τους, πιθανοί εργοδότες μπορούν να τους προτιμήσουν σε θέσεις εργασίας.

Ο χρήστης μπορεί να δημιουργήσει ένα λογαριασμό στο σύστημα μέσω της Android εφαρμογής και να εισάγει τα στοιχεία τους, όπως για παράδειγμα το όνομα που λογαριασμού που κατέχει στο GitHub και στο Stack Exchange. Μέσω της εφαρμογής ο χρήστης μπορεί να επικοινωνήσει με την χρήση email με άλλους χρήστες.

Ένας εργοδότης μπορεί να δημιουργήσει ένα λογαριασμό στο σύστημα μέσω της Android εφαρμογής και να αναζητήσει διάφορα «ταλέντα» που θα μπορούσε να προσλάβει.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Κίνητρο Διπλωματικής Εργασίας	1
	1.2 Στόχος Διπλωματικής Εργασίας	4
Κεφάλαιο 2	Συστήματα Ελέγχου Εκδόσεων.....	4
	2.1.1 Τοπικά συστήματα ελέγχου εκδόσεων	5
	2.1.2 Συγκεντρωτικά συστήματα ελέγχου εκδόσεων	5
	2.1.3 Κατανεμημένα συστήματα ελέγχου εκδόσεων	6
	2.2 GitHub	6
	2.3 Ιστότοποι Ερωτοαπαντήσεων – Stack Exchange	7
	2.4 Linkedin	8
Κεφάλαιο 3	Λειτουργικό Σύστημα Android.....	9
	3.1 Εισαγωγή στο Android	9
	3.2 Ανάπτυξη Εφαρμογών	10
Κεφάλαιο 4	Spring Boot Εφαρμογή.....	16
	4.1 Εισαγωγή στο Spring Boot	16
	4.2 Εισαγωγή στο Gradle	17
	4.3 Εισαγωγή στη MySQL	18
Κεφάλαιο 5	Εξόρυξη Δεδομένων	20
	5.1 API (Application Programming Interface)	20
	5.2 HTTP (Hypertext Transfer Protocol)	20
	5.3 GitHub API	22
	5.4 Stack Exchange API	22
Κεφάλαιο 6	Ανάπτυξη Εφαρμογής	25
	6.1 Προδιαγραφές και Απαιτήσεις	25

6.2 Υλοποίηση Android Εφαρμογής	28
6.3 Υλοποίηση Web Εφαρμογής	37
6.3.1 Αρχιτεκτονική	37
6.3.2 Βάση Δεδομένων	37
6.4 API Web Services	38
Κεφάλαιο 7 Συμπεράσματα	40
7.1 Συμπεράσματα ΑΔΕ	40
7.2 Αξιολόγηση ΑΔΕ	41
7.3 Επεκτασιμότητα	42
Βιβλιογραφία	44

Κεφάλαιο 1

Εισαγωγή

1.1 Κίνητρο Διπλωματικής Εργασίας	1
1.2 Στόχος Διπλωματικής Εργασίας	3

1.1 Κίνητρο Διπλωματικής Εργασίας

Το κίνητρο μου για να υλοποιήσω την συγκεκριμένη διπλωματική πέραν από τις γνώσεις που θα αποκομίσω από αυτή την εργασία, είναι και η προοπτική χρήσης της από άτομα της Πληροφορικής.

Ζούμε σε μία εποχή όπου η εργασία ξεκινά να αλλάζει. Ειδικότερα στο κλάδο της Πληροφορικής η τηλεργασία έχει έρθει για τα καλά. Πλέον μεγάλοι κολοσσοί της Πληροφορικής (Microsoft, Google) προσλαμβάνουν άτομα τα οποία ξεκινούν να εργάζονται από το σπίτι. Αυτό για τις εταιρείες διευρύνει την δεξαμενή των ταλέντων αφού μπορούν να προσλάβουν άτομα τα οποία όχι μόνο θα δουλεύουν από το σπίτι τους, αλλά θα βρίσκονται σε άλλη χώρα ακόμα και άλλη ήπειρο.

Δεν θα μπορούσε να αλλάξει η εργασία και να μην αλλάξει και η διαδικασία πρόσληψης ενός υπαλλήλου. Πλέον οι εταιρείες έχουν περισσότερους υποψήφιους για εργοδότηση, συνεπώς και περισσότερες απαιτήσεις. Δεν είναι βέβαια τυχαίο πως η πλατφόρμα επαγγελματικής κοινωνικής δικτύωσης LinkedIn είδε μεγάλη άνοδο τα τελευταία χρόνια μετρώντας περισσότερους από 300 εκατομμύρια εγγεγραμμένους χρήστες σε περισσότερες από 200 χώρες.

Η σημασία των social media για τους recruiters κατά τη διαδικασία των προσλήψεων αποτελεί σημαντική αφού τα ενσωματώνουν σε μεγάλο βαθμό κατά τις αξιολογήσεις που κάνουν στα βιογραφικά που λαμβάνουν καθημερινά. Επομένως ένας υποψήφιος χρειάζεται να δημιουργήσει ένα πλήρες επαγγελματικό προφίλ στο οποίο θα παραθέτει δείγματα της δουλειάς του.

Δεν αποκλείεται μελλοντικά να καταργηθεί και το «παραδοσιακό» βιογραφικό αφού ο εργοδότης θα μπορεί να έχει άμεση αντίληψη των δεξιοτήτων και των ικανοτήτων ενός υποψήφιου εργαζόμενου απλώς αναζητώντας των σε ένα μέσο κοινωνικής δικτύωσης. Το γεγονός αυτό είναι ένας επιπλέον λόγος για τη συγκεκριμένη μελέτη.

Σήμερα ένας μηχανικός λογισμικού μπορεί να προβάλει την δουλειά του μέσω κάποιων συστημάτων ελέγχουν εκδόσεων όπως είναι το GitHub. Σε αυτά τα συστήματα οι χρήστες μπορούν να δημιουργήσουν τα δικά τους repositories ώστε να αποθηκεύουν πηγαίους κώδικες.

Μπορεί επίσης να ενταχθεί σε μία διαδικτυακή κοινότητα όπως είναι το Stack Overflow το οποίο προσφέρει στον χρήστη την ικανότητα να απαντούν και να ρωτούν διάφορες ερωτήσεις που σχετίζονται με συγκεκριμένα θέματα. Το Stack Overflow είναι η δημοφιλέστερη σελίδα της κοινότητας Stack Exchange που αποτελείτε από πολλές άλλες ιστοσελίδες ερωτοαπαντήσεων. Σε αυτές τις ιστοσελίδες ένας μηχανικός λογισμικού μπορεί να πάρει φήμη για την συνεισφορά του στην κοινότητα αφού βαθμολογείται θετικά ή αρνητικά για τις ερωτήσεις και τις απαντήσεις του.

Επομένως υπάρχει η ανάγκη δικτύωσης των μηχανικών λογισμικού και προβολής της δουλειάς τους αλλά και της γνώσης που κατέχουν σχετικά με διάφορα προγραμματιστικά πεδία. Έτσι μέσα από τη προβολή των προφίλ τους, πιθανοί εργοδότες μπορούν να τους προτιμήσουν σε θέσεις εργασίας.

Οι πιο πάνω λόγοι αποτέλεσαν το κίνητρο μελέτης των δύο αυτών μέσων κοινωνικής δικτύωσης με σκοπό την δημιουργία μιας εφαρμογής η οποία θα δημιουργεί το προφίλ ενός μηχανικού λογισμικού και θα το προβάλει σε έναν εργοδότη.

1.2 Στόχος Διπλωματικής Εργασίας

Κύριος στόχος της παρούσας διπλωματικής είναι η μελέτη των λογαριασμών GitHub και Stack Exchange ενός χρήστη, ώστε να δημιουργηθεί ένα προφίλ το οποίο ένας εργαζόμενος θα μπορεί να χρησιμοποιήσει για να προβάλει την δουλεία του, όπως επίσης και ένας εργοδότης να αναζητήσει και να επικοινωνήσει με χρήστες τους οποίους κρίνει ότι θα μπορούσαν να ενταχθούν στο δυναμικό του.

Αυτό θα επιτυγχάνεται με τη δημιουργία μίας Android εφαρμογής στην οποία ο χρήστης θα εισάγει τα στοιχεία του και θα δημιουργείται το προφίλ του. Στην εφαρμογή θα μπορεί να αναζητήσει άλλους χρήστες καθώς και να επικοινωνήσει μαζί τους με σκοπό να ανταλλάξουν γνώσεις μεταξύ τους.

Παράλληλα θα μπορούν οι εργοδότες να αναζητήσουν μέσω της Android εφαρμογής χρήστες κατάλληλους για μία θέση, αφού θα μπορούν να βλέπουν το προφίλ τους ανάλογα με τις δραστηριότητες τους στις δύο ιστοσελίδες.

Κεφάλαιο 2

Πηγές Πληροφοριών

2.1 Συστήματα Ελέγχου Εκδόσεων	4
2.1.1 Τοπικά συστήματα ελέγχου εκδόσεων	5
2.1.2 Συγκεντρωτικά συστήματα ελέγχου εκδόσεων	5
2.1.3 Κατανεμημένα συστήματα ελέγχου εκδόσεων	6
2.2 GitHub	6
2.3 Ιστότοποι Ερωτοαπαντήσεων – Stack Exchange	7
2.4 LinkedIn	8

2.1 Συστήματα Ελέγχου Εκδόσεων

Ο έλεγχος εκδόσεων είναι ένα σύστημα το οποίο καταγράφει αλλαγές σε ένα αρχείο ή σε ένα σύνολο αρχείων έτσι ώστε να μπορούμε να ανακαλέσουμε συγκεκριμένες εκδόσεις. Δηλαδή μπορούμε να επαναφέρουμε αρχεία σε κάποια προγενέστερη κατάσταση, να δούμε ποιος τροποποίησε τελευταίος ένα αρχείο ή και να συγκρίνουμε αλλαγές.

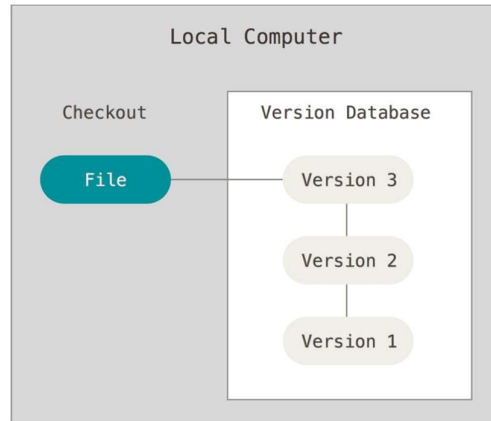
Η χρήση ενός συστήματος ελέγχου εκδόσεων μας προστατεύει από διάφορα προβλήματα που μπορούμε να αντιμετωπίσουμε αφού ανά πάσαν στιγμή μπορούμε να επαναφέρουμε οποιαδήποτε αλλαγή έγινε.

Υπάρχουν τρία είδη συστημάτων ελέγχου εκδόσεων

- Τοπικά συστήματα ελέγχου εκδόσεων
- Συγκεντρωτικά συστήματα ελέγχου εκδόσεων

- Κατανεμημένα συστήματα ελέγχου εκδόσεων.

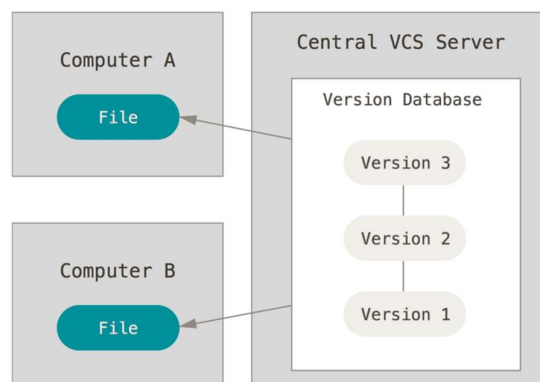
2.1.1 Τοπικά συστήματα ελέγχου εκδόσεων



Εικόνα 4.1.1.1

Τα συστήματα αυτά διαθέτουν μια απλή βάση δεδομένων που κρατάει όλες τις αλλαγές στα αρχεία σε έλεγχο εκδόσεων. Ένα από τα πιο δημοφιλή συστήματα ελέγχου εκδόσεων είναι το σύστημα RCS το οποίο υπάρχει ακόμη σε πολλούς υπολογιστές.

2.1.2 Συγκεντρωτικά συστήματα ελέγχου εκδόσεων

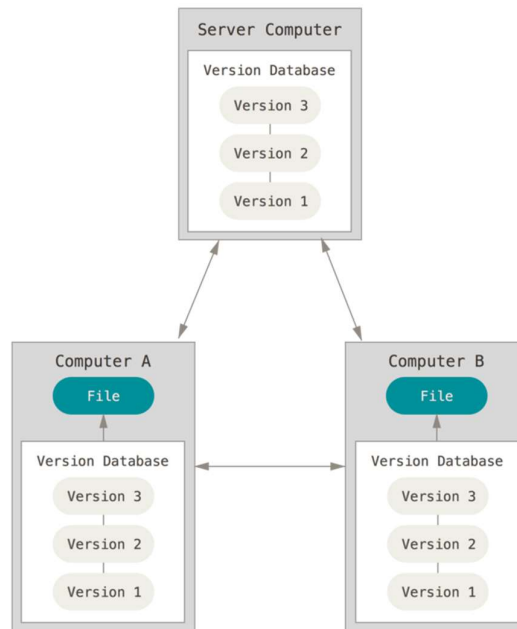


Εικόνα 4.1.2.1

Καθώς η πληροφορική αναπτυσσόταν υπήρχε η ανάγκη για συνεργασία μεταξύ προγραμματιστών σε άλλα συστήματα. Για αυτό δημιουργήθηκαν τα συγκεντρωτικά συστήματα ελέγχου εκδόσεων. Τα συστήματα αυτά όπως το CVS, SVN έχουν ένα διακομιστή ο οποίος είναι υπεύθυνος για την ενημέρωση των αλλαγών. Αυτό το σύστημα

έχει ένα σημαντικό μειονέκτημα αφού αν ο διακομιστής αντιμετωπίζει πρόβλημα δεν μπορεί κανένας χρήστης να ενημερώσει με της αλλαγές του τα αρχεία αλλά και αν δεν γίνουν σωστά τα αντίγραφα ασφαλείας ενδέχεται να χαθούν οι αλλαγές ή και αρχεία.

2.1.3 Κατανεμημένα συστήματα ελέγχου εκδόσεων



Εικόνα 4.1.3.1

Σε ένα κατανεμημένο σύστημα όπως το Git οι χρήστες δεν ενημερώνουν μόνο με τις αλλαγές τους τα αρχεία αλλά δημιουργούν ένα κλώνο τοπικά ολόκληρου του αποθετηρίου, συνεπώς αν υπάρχει πρόβλημα με τον διακομιστή, τα τοπικά αποθετήρια μπορούν να χρησιμοποιηθούν για να τον επαναφέρουν.

2.2 GitHub

Το GitHub είναι μία πλατφόρμα, η οποία προσφέρει στους μηχανικούς λογισμικού τη λειτουργία του κατανεμημένου συστήματος ελέγχου και διαχείρισης πηγαίου κώδικα του Git. Παρέχει έλεγχο πρόσβασης και πολλές δυνατότητες συνεργασίας με άλλους προγραμματιστές όπως εντοπισμό σφαλμάτων και διαχείριση εργασιών.

Όλοι οι μεγάλοι οργανισμοί οι οποίοι παρέχουν ανοικτού πηγαίου κώδικα χρησιμοποιούν το GitHub για τα repositories τους. Η Microsoft από το 2012 είναι ένας από τους μεγαλύτερους χρήστες αφού όλα τα open-source project τα φιλοξενεί στο GitHub. Στις 26 Οκτωβρίου το 2018 η Microsoft αποκτείσαι το GitHub έναντι 7.5 δισεκατομμυρίων δολαρίων.

Το GitHub προσφέρει τις βασικές υπηρεσίες του δωρεάν. Ο κάθε χρήστης έχει το δικό του λογαριασμό ο οποίος περιέχει όλα τα repositories που δημιούργησε καθώς και τα repositories στα οποία συνείσφερε.

2.3 Ιστότοποι Ερωτοαπαντήσεων – Stack Exchange



Εικόνα 4.3.1

Το StackExchange είναι ένα δίκτυο από ιστότοπους ερωτοαπαντήσεων για θέματα σε διαφορετικά πεδία, όπου κάθε ιστότοπος καλύπτει ένα συγκεκριμένο θέμα. Σε όλες τις ιστοσελίδες του StackExchange οι χρήστες μπορούν να ψηφίζουν αρνητικά ή θετικά τόσο τις ερωτήσεις, όσο και τις απαντήσεις που αναρτούν άλλοι χρήστες. Με αυτό το σύστημα οι χρήστες αυτοεπιβλέπονται αφού για κάθε χρήστη υπάρχει φήμη (reputation) ανάλογα με τις θετικές ή αρνητικές ψήφους που έλαβε.

Ανάλογα με την δραστηριότητα του χρήστη στις ιστοσελίδες του StackExchange μπορεί να πάρει και εμβλήματα (badges). Τα εμβλήματα χωρίζονται σε τρεις κατηγορίες οι οποίες είναι χρυσά, αργυρά και χάλκινα. Ανάλογα με την φήμη και τα εμβλήματα που έχει κάποιος χρήστης του δίνονται και κάποια προνόμια όπως το κλείσιμο κάποιων ερωτήσεων. Οι τρεις μεγαλύτερες ιστοσελίδες ερωτοαπαντήσεων είναι το Stack Overflow, το Super User και το Ask Ubuntu.

2.4 LinkedIn

Το LinkedIn είναι ιστοχώρος επαγγελματικής κοινωνικής δικτύωσης. Από το Μάιο του 2020, το LinkedIn είχε 706 εκατομμύριο εγγεγραμμένα μέλη από 150 χώρες. Από το Δεκέμβριο του 2016 ανοίκει στην Microsoft. Στην παρούσα διπλωματική δεν θα ασχοληθούμε με το LinkedIn αφού δεν μπορούμε να πάρουμε πληροφορίες για τους χρήστες χωρίς να είμαστε συνεργάστες τους.

Κεφάλαιο 3

Λειτουργικό Σύστημα Android

3.1 Εισαγωγή στο Android	9
3.2 Ανάπτυξη Εφαρμογών	10

3.1 Εισαγωγή στο Android



Εικόνα 3.1.1

Το Android είναι ένα λειτουργικό σύστημα για Smartphones, tablets και άλλες φορητές συσκευές που αναπτύχθηκε από την Google το 2005 όταν αγόρασε τα δικαιώματα από την ομώνυμη εταιρεία Android Inc. Κυκλοφόρησε επίσημα το 2007 ως λειτουργικό σύστημα και το 2008 κυκλοφόρησε η πρώτη συσκευή που είχε εγκατεστημένο εργοστασιακά το λειτουργικό σύστημα Android.

Αν και είναι κατά κύριο λόγο σχεδιασμένο για συσκευές με οθόνη αφής, όπως τα smartphones και τα tablets έχει χρησιμοποιηθεί και για τηλεοράσεις, αυτοκίνητα και ρολόγια χειρός. Το λειτουργικό σύστημα υποστηρίζει φωνητικές λειτουργίες, εικονικό πληκτρολόγιο και άλλες βασικές λειτουργίες που πρέπει να έχει ένα κινητό τηλέφωνο. 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.

Ένα μεγάλο μέρος του κώδικα του λειτουργικού συστήματος Android είναι ανοικτός κώδικας και είναι διαθέσιμος για επεξεργασία από τους κατασκευαστές smartphone συσκευών. Ο πυρήνας του Android βασίζεται στον πυρήνα Linux για τις βασικές του λειτουργίες και η κάθε κατασκευάστρια εταιρεία παρέχει τα ανάλογα drivers για τα υποσυστήματα της συσκευής, λειτουργίες διαχείρισης μνήμης, διαχείριση διεργασιών και δικτύωσης. Η κάθε εταιρεία έχει την δυνατότητα να «πειράξει» τον πυρήνα αν κρίνει απαραίτητο με βάση τις ανάγκες της συσκευής.

Μια «καθαρή» έκδοση του Android όπως αρχικά γράφεται από την Google χωρίς τροποποιήσεις υπάρχει στις συσκευές Pixel και σε όσες συσκευές υποστηρίζουν το Android One.

3.2 Ανάπτυξη Εφαρμογών

Η ανάπτυξη εφαρμογών για το λειτουργικό σύστημα Android γίνεται με τη χρήση της γλώσσας προγραμματισμού Java και την Kotlin που υποστηρίζεται από την Google. Οι δύο γλώσσες τρέχουν με την χρήση JVM και μπορούν να χρησιμοποιηθούν παράλληλα για μία ανάπτυξη εφαρμογής. Μπορούν να χρησιμοποιηθούν και άλλες γλώσσες προγραμματισμού όπως η C++, Python και C# αλλά και η Dart η οποία όμως χρησιμοποιείτε μαζί με το Flutter.

Το Flutter είναι ένα open-source SDK από την Google με σκοπό την ανάπτυξη εφαρμογών για Android, iOS, Linux, Mac, Window, Google Fuchsia και Web με την χρήση ενός πηγαίου κώδικα.

Η υλοποίηση του προγράμματος που έκανα στα πλαίσια της πτυχιακής μου έγινε με την χρήση της γλώσσας προγραμματισμού Kotlin (τελευταία έκδοχή 1.4.21) και JVM 1.8. Το IDE που χρησιμοποίησα είναι το Android Studio (έκδοχή 4.1.1). Για την ανάπτυξη εφαρμογής χρησιμοποίησα το MVVM μοντέλο (Model-View-View Model) αρχιτεκτονικής.

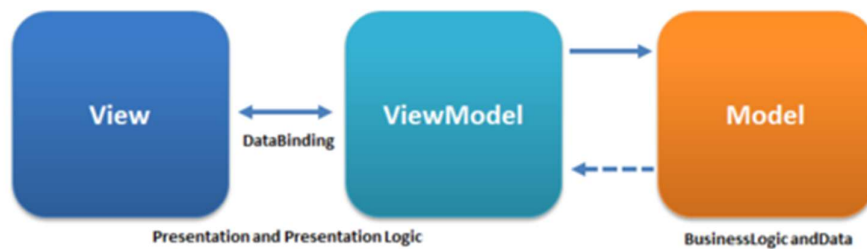
Χρησιμοποίησα τη γλώσσα προγραμματισμού Kotlin η οποία σχεδιάστηκε από την ομάδα JetBrains. Από το 2019 είναι η επίσημη γλώσσα προγραμματισμού για εφαρμογές Android. Η JetBrains είναι εταιρεία ανάπτυξης λογισμικού της οποίας τα εργαλεία απευθύνονται σε προγραμματιστές λογισμικού. Το IntelliJ IDEA είναι το πρώτο IDE της εταιρείας το οποίο στοχεύει σε τεχνολογίες βασισμένες σε JVM. Η εταιρεία παρέχει open-source μία έκδοση του IDE. Το Android Studio IDE είναι βασισμένο στην open-source έκδοση του IntelliJ IDEA.

Χαρακτηριστικά	Java	Kotlin
Null Safe	Η Java δεν παρέχει στον προγραμματιστή προστασία για το Null Pointer Exception, αφού μπορούμε να δηλώσουμε όλες τις μεταβλητές ως null	Η Kotlin δεν θα σε αφήσει να δηλώσεις μια μεταβλητή null αφού ο κώδικας θα εμφανίσει το πρόβλημα στο compile-time. Εξ ορισμού της Kotlin όλες οι μεταβλητές είναι non-nullable. Αν χρειάζεται να δηλώσεις μία μεταβλητή με null θα πρέπει να την δηλώσεις ως nullable.
Extensions Functions	Στην Java αν θέλουμε να κάνουμε extent μία υπάρχων κλάση, πρέπει να δημιουργήσουμε μία καινούρια κλάση η οποία θα κληρονομεί από αυτήν. Επομένως τα Extensions functions δεν υποστηρίζονται.	Η Kotlin παρέχει την δυνατότητα να κάνει extension function μίας υπάρχων κλάσης. Για παράδειγμα μπορούμε να δημιουργήσουμε ένα καινούριο function για την κλάση String, η οποία θα μπορεί να χρησιμοποιηθεί από όλες της μεταβλητές String.
Data Classes	Στην Java αν θέλουμε να δημιουργήσουμε μια κλάση η οποία θα είναι υπεύθυνη για να κρατά δεδομένα, πρέπει να δηλώσουμε όλους τους	Στην Kotlin αν θέλουμε να δημιουργήσουμε μια κλάση για να κρατά δεδομένα απλώς την δηλώνουμε ως data class A(). Συνεπάγεται λιγότερος boilerplate

	getters και setters, τους constructor και τις μεθόδους hashCode(), toString() και equals()	code και λιγότερος χρόνος υλοποίησης.
Type Inference	Στην Java πρέπει να δηλώνουμε ρητά τον τύπο κάθε μεταβλητής κατά δήλωση της	Στην Kotlin δεν είναι απαραίτητο να δηλώνουμε τον τύπο της μεταβλητής

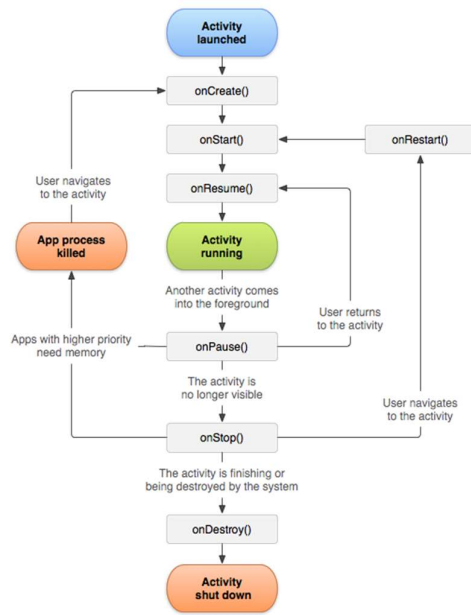
Πίνακας 3.2.1

Όπως βλέπουμε από το πιο πάνω πίνακα (3.2.1) υπάρχουν αρκετές διαφορές μεταξύ των δύο γλωσσών προγραμματισμού. Εκτός από την Android εφαρμογή χρησιμοποιήσαμε την Kotlin και για την υλοποίηση της Web εφαρμογής, για την οποία θα αναφερθούμε στην συνέχεια.

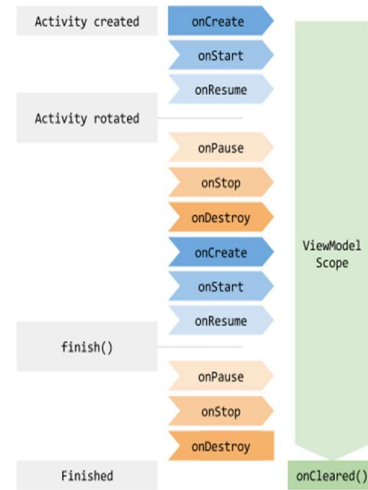


Σχήμα 3.2.2 MVVM από Google Developers

Το MVVM μοντέλο διευκολύνει το διαχωρισμό της ανάπτυξης της γραφικής διεπαφής του χρήστη από την ανάπτυξη της επιχειρηματικής λογικής (Model) έτσι ώστε η προβολή (View) να μην εξαρτάται από κάποιο συγκεκριμένο μοντέλο. Το View Model καθορίζει κάποια ή ακόμη και όλη την λογική του View. Το MVVM μοντέλο είναι μια παραλλαγή του σχεδιαστικού μοντέλου «Presentation Model» του Martin Fowler. Εφευρέθηκε από τους Ken Cooper και Ted Peters της Microsoft με σκοπό να απλοποιήσει τον προγραμματισμό των διεπαφών χρήστη.



Σχήμα 3.2.3 Activity Lifecycle Σχήμα
από Google Developers



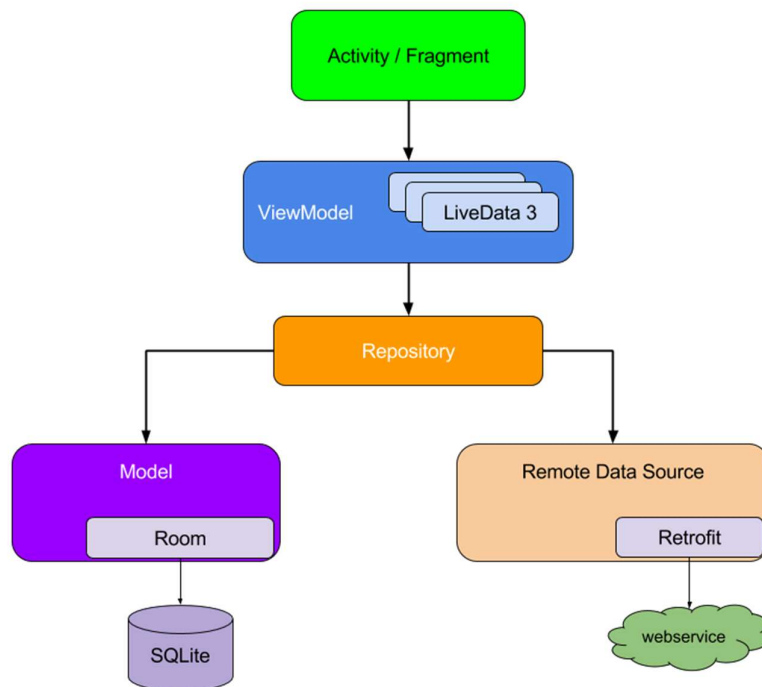
3.2.4 View Model Lifecycle
από Google Developers

Το Android Framework διαχειρίζεται τα lifecycle των ελεγκτών διεπαφής χρήστη όπως τα activities και τα fragments. Το framework ενδέχεται να αποφασίσει να καταστρέψει ή να ξαναδημιουργήσει ένα activity ή fragment (UI controllers) σε συγκεκριμένες ενέργειες του χρήστη ή σε συμβάντα που είναι εντελώς εκτός ελέγχου (low memory). Αν το σύστημα καταστρέψει ή δημιουργήσει εκ νέου ένα UI controller, τυχόν προσωρινά δεδομένα που σχετίζονται με το περιβάλλον εργασίας του χρήστη θα χαθούν.

Στο σχήμα 3.2.3 βλέπουμε το lifecycle ενός UI Controller, στην περίπτωση μας ενός activity. Για παράδειγμα, στην εφαρμογή μου υπάρχει μια λίστα από στοιχεία που πρέπει ο χρήστης να συμπληρώσει με σκοπό να γραφτεί στο σύστημα μας ως user. Το activity «Registration Form» τρέχει μεταξύ της onResume() και της onPause(). Αν ο χρήστης για παράδειγμα αλλάξει εφαρμογή, υπάρχει πιθανότητα το λειτουργικό σύστημα αν δεν υπάρχει ο κατάλληλος χώρος στην μνήμη να καταστρέψει το activity και όταν επιστρέψει πίσω στην εφαρμογή μας να χάσει τα δεδομένα που εισήγαγε, αφού το activity θα ξαναδημιουργηθεί.

Με την χρήση του MVVM μπορούμε να λύσουμε αυτό το πρόβλημα αφού εξ ορισμού του, διαχωρίζουμε το View από το Business Logic και τα Data. Όπως βλέπουμε στο Σχήμα 3.2.4 το View Model έχει δικό του lifecycle το οποίο συνάπτει με του activity με την διαφορά ότι δεν καταστρέφεται ή ξαναδημιουργείται αλλά γίνεται «clear» όταν γίνει «finish».

Αυτό γίνεται αφού το View Model δεν έχει UI Components με αποτέλεσμα να μην καταναλώνει αρκετούς πόρους από το σύστημα. Στο παράδειγμα μας όταν ο UI controller ξαναδημιουργηθεί θα εισαχθούν τα δεδομένα του χρηστή ξανά αφού θα υπάρχουν αποθηκευμένα στο View Model και όχι στον UI controller και έτσι ο χρήστης δεν θα χάσει τα δεδομένα του.



Σχήμα 3.2.5 Αρχιτεκτονική Android από Google Developers

Στο σχήμα 3.2.5 προχωρούμε ακόμη ένα βήμα και βλέπουμε το Repository το οποίο διαλέγει από πού θα πάρει τα δεδομένα που θα δώσει στο View Model για να τα εμφανίσει ο UI controller μας. Είτε θα πάρει τα δεδομένα από την βάση δεδομένων του

τηλεφώνου με την χρήση της Room (Object Relation Mapping library) είτε από webservice με την χρήση της Retrofit library.

Υπάρχει επίσης και η επιλογή να χρησιμοποιηθούν και οι δύο τρόποι αφού μια εφαρμογή μπορεί να εμφανίζει του χρήστη τα δεδομένα που υπάρχουν στην βάση του τηλεφώνου (offline data) μέχρι να ενημερωθούν από μια απομακρυσμένη βάση δεδομένων (online data).

Στην εφαρμογή που υλοποίησα δεν χρησιμοποιώ βάση δεδομένων στο τηλέφωνο με αποτέλεσμα ο χρήστης να χρειάζεται διαδίκτυο για να χρησιμοποιήσει την εφαρμογή.

Κεφάλαιο 4

Spring Boot Εφαρμογή

4.1 Εισαγωγή στο Spring Boot	16
4.2 Εισαγωγή στο Gradle	17
4.3 Εισαγωγή στη MySQL	18

4.1 Εισαγωγή στο Spring Boot

Το Spring Framework παρέχει ολοκληρωμένη υποστήριξη για την ανάπτυξη εφαρμογών Java. Έχει αρκετά χαρακτηριστικά όπως το «Dependency Injection» και «Out of the box Modules» όπως:

- Spring JDBC
- Spring MVC
- Spring Security
- Spring AOP
- Spring ORM
- Spring Test

Αυτά τα modules μπορούν να μειώσουν δραστικά τον χρόνο ανάπτυξης μίας εφαρμογής. Για παράδειγμα, στις πρώτες μέρες της ανάπτυξης Java εφαρμογών, χρειαζόταν να γράψουμε πολύ κώδικα boilerplate για να εισαγάγουμε μια εγγραφή σε μια βάση δεδομένων. Αλλά με τη χρήση του JDBCTemplate του Spring JDBC module μπορούμε να το μειώσουμε σε μερικές γραμμές κώδικα με λίγες μόνο διαμορφώσεις.

Το Spring Boot είναι ένα open-source web framework βασισμένο στην Java. Είναι βασικά μία επέκταση του Spring framework που εξαλείφει τις διαμορφώσεις του boilerplate που απαιτούνται για την δημιουργία μιας εφαρμογής.

Το Spring Boot είναι πρωταρχικής σημασίας για μια ομάδα ανάπτυξης που θέλει να αφιερώσει το χρόνο της στην ανάπτυξη παρά να διαμορφώνει και να ρυθμίζει συνεχώς. Διαθέτει πολλές δυνατότητες που το καθιστούν ιδανικό για γρήγορη ανάπτυξη εφαρμογών Java, όπως αυτόματη διαμόρφωση, ελέγχους υγείας και μετρήσεις απόδοσης του συστήματος και εξαρτήσεις (dependencies).

Πλεονεκτήματα του Spring Boot Framework

Χαρακτηριστικό	Πλεονέκτημα
Αυτόνομη Εφαρμογή	Μπορεί πολύ εύκολα να παράγει το jar της εφαρμογής και να το εκτελέσει χωρίς να χρειάζεται οποιαδήποτε ρύθμιση
Ενσωματωμένοι διακομιστές	Έρχεται με προ-εγκατεστημένους διακομιστές εφαρμογών (Tomcat, Jetty και Undertow) που δεν χρειάζονται εγκατάσταση για την χρήση τους. Παρέχει επίσης ταχύτερες και αποδοτικότερες εφαρμογές
Auto-Configurable	Το Spring Boot framework διαμορφώνονται αυτόματα
Χαρακτηριστικά παραγωγής	Παρέχει ελέγχους υγείας και μετρήσεις απόδοσης του συστήματος
Εξαρτήσεις (Dependencies)	Παρέχει εξαρτήσεις οι οποίες έχουν διαμορφωθεί για την απλοποίηση της διαμόρφωσης του build. Επίσης παρέχει πλήρη build εργαλεία όπως είναι το Maven και το Gradle.

Πίνακας 4.1.1

4.2 Εισαγωγή στο Gradle

Το Gradle είναι ένα εργαλείο αυτοματοποίησης κατασκευής για την ανάπτυξη λογισμικού πολλαπλών γλωσσών. Οι υποστηριζόμενες γλώσσες περιλαμβάνουν την Java (Kotlin, Groovy, Scala), τη C/C++ και τη JavaScript. Ελέγχει όλες τις διαδικασίες ανάπτυξης μιας εφαρμογής, όπως το testing, το deployment και το publishing.

Ορισμένα από τα πιο σημαντικά χαρακτηριστικά του είναι:

- Υψηλή Απόδοση

Το Gradle αποφεύγει την άσκοπη εργασία εκτελώντας μόνο τις εργασίες που πρέπει να εκτελεστούν αφού ελέγχει αν οι είσοδοι ή οι έξοδοι μιας διεργασίας έχει αλλάξει.

- JVM Foundation

Το Gradle εκτελείται σε JVM, αυτό είναι ένα σημαντικό χαρακτηριστικό αφού μπορεί να εκτελεστεί σε διαφορετικές πλατφόρμες. Είναι σημαντικό να αναφέρουμε ότι το Gradle δεν περιορίζεται στο να κατασκευάζει JVM projects, αλλά υπάρχει και η δυνατότητα να υποστήριξη και native projects.

- Επεκτασιμότητα

Το Gradle μπορεί εύκολο να επεκταθεί για να παρέχει διαφορετικούς τύπους διεργασιών ή και ακόμη διαφορετικό μοντέλο. Το Android Gradle παρέχει πρόσθετες διεργασίες όπως είναι τα flavors και τα build types.

- Υποστήριξη IDE

Αρκετά μεγάλα IDE (Android Studio, IntelliJ IDEA, Eclipse και NetBeans) επιτρέπουν την εισαγωγή Gradle builds και αλληλοεπιδρούν μεταξύ τους.

4.3 Εισαγωγή στη MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων ανοικτού κώδικα. Μια σχεσιακή βάση δεδομένων οργανώνει δεδομένα σε ένα ή και περισσότερους πίνακες δεδομένων των οποίων οι τύποι τους ενδέχεται να σχετίζονται μεταξύ τους.

Η MySQL συνήθως χρησιμοποιείται με άλλα προγράμματα για να υλοποιηθούν εφαρμογές που χρειάζονται σχεσιακή βάση δεδομένων. Είναι ένα στοιχείο της στοίβας λογισμικού εφαρμογών LAMP το οποίο σημαίνει Linux, Apache, MySQL,

Perl/PHP/Python. Χρησιμοποιείται από πολλές εφαρμογές ιστού που βασίζονται σε σχεσιακή βάση δεδομένων όπως τα Drupal, Joomla, phpBB και WordPress. Πολλές γλώσσες προγραμματισμού περιλαμβάνουν βιβλιοθήκες για πρόσβαση σε βάσεις δεδομένων MySQL όπως είναι οι MySQL Connector/Net για .NET και JDBC για JAVA.

Στην παρούσα διπλωματική εργασία θα χρησιμοποιήσουμε MySQL για την διαχείριση της σχεσιακής βάσης της Web Εφαρμογής μας. Ο JDBC driver της Java μπορεί να χρησιμοποιηθεί και από την Kotlin.

Κεφάλαιο 5

Εξόρυξη Δεδομένων

5.1 API (Application Programming Interface)	20
5.2 HTTP (Hypertext Transfer Protocol)	20
5.3 GitHub API	22
5.4 Stack Exchange API	22

5.1 API (Application Programming Interface)

Για την εξόρυξη των δεδομένων από το Stack Exchange και το GitHub χρησιμοποιήθηκαν τα APIs της κάθε πηγής. Το API είναι διαμεσολαβητής (interface) λογισμικού το οποίο επιτρέπει σε δύο εφαρμογές να «μιλούν» μεταξύ τους. Το REST API είναι interface εφαρμογής το οποίο κληρονομεί τους περιορισμούς του αρχιτεκτονικού στύλ REST και επιτρέπει την αλληλεπίδραση με άλλες RESTful υπηρεσίες.

5.2 HTTP (Hypertext Transfer Protocol)

Το HTTP είναι ένα πρωτόκολλο επικοινωνίας, το οποίο αποτελεί το κύριο πρωτόκολλο για την μεταφορά δεδομένων ενός διακομιστή (server) και ενός πελάτη (client).

Μέθοδοι αίτησης του HTTP:

- **GET**

Η μέθοδος GET ζητά από τον διακομιστή να ανακτήσει οποιαδήποτε πληροφορία ανάλογα με τη διεύθυνση URL που θα εισάγει ο πελάτης

- **POST**

Η μέθοδος POST στέλνει δεδομένα ενός διακομιστή και τα διαχειρίζεται ανάλογα της διεύθυνσης URL. Επιστρέφει πίσω πληροφορία που περιέχει το αποτέλεσμα της μεθόδου.

- **PUT**

Η μέθοδος PUT ζητά από τον διακομιστή να αποθηκεύσει δεδομένα ανάλογα της διεύθυνσης URL. Αν στην διεύθυνση URL υπάρχουν δεδομένα τότε τροποποιούνται αλλιώς δημιουργούνται

- **DELETE**

Η μέθοδος DELETE ζητά από τον διακομιστή να διαγράψει δεδομένα ανάλογα της διεύθυνσης URL.

Κάθε αίτηση λαμβάνει μία απάντηση στην οποία υπάρχει ένα status line με έναν κωδικό και μία φράση.

Το πρώτο ψηφίου του κωδικού χρησιμοποιείται για την υποδιαίρεση των αιτήσεων σε 5 κατηγορίες:

- 1xx: Informational
- 2xx: Successful
- 3xx: Redirection
- 4xx: Client Error
- 5xx: Server Error

5.3 GitHub API

Για την συλλογή των δεδομένων από το GitHub δημιούργησα ένα Rest Client τον οποίο χρησιμοποιώ για να αντλήσω τα δεδομένα.

GitHub API	Details
/users/{id}	Το χρησιμοποιώ με σκοπό να πάρω τα δεδομένα ενός χρήστη του GitHub. Παίρνει ως παράμετρο στο PATH του URL το user id του χρήστη.
/users/{id}/repos	Το χρησιμοποιώ με σκοπό να πάρω μία λίστα με όλα τα repositories ενός χρήστη του GitHub. Παίρνει ως παράμετρο στο PATH του URL το user id του χρήστη.
/repos/{id}/{repo}/languages	Το χρησιμοποιώ με σκοπό να πάρω ένα HashMap<String,Long> για ένα συγκεκριμένο repository ενός χρήστη του GitHub. Παίρνει ως παράμετρο στο PATH του URL το user id και το repository του χρήστη.

Πίνακας 5.3.1

Χρησιμοποιώντας τα APIs του πίνακα 5.3.1 μπορούμε να εξάγουμε τα τελικά δεδομένα από το GitHub τα οποία θα χρησιμοποιηθούν από την εφαρμογή μας για τη δημιουργία του προφίλ.

5.4 Stack Exchange API

Για την συλλογή των δεδομένων από το Stack Exchange δημιούργησα ένα Rest Client τον οποίο χρησιμοποιώ για να αντλήσω τα δεδομένα. Σε αυτή την περίπτωση χρειάστηκε να δημιουργήσω μία εφαρμογή για να μου δοθεί ένα key και ένα client id. Αυτό το key

μας παρέχει μεγαλύτερο αριθμό αιτημάτων, αλλά επίσης και την εξαγωγή δεδομένων που δεν μας παρέχονται για ένα χρήστη διαφορετικά.

Stack Exchange API	Details
<code>/me?access_token=</code>	Το χρησιμοποιώ με σκοπό να πάρω τα δεδομένα ενός χρήστη στο Stack Exchange
<code>/users/{id}/questions?sort=votes</code>	Το χρησιμοποιώ με σκοπό να πάρω όλες της ερωτήσεις ενός χρήστη στο Stack Exchange
<code>/users/{id}/answers?sort=votes</code>	Το χρησιμοποιώ με σκοπό να πάρω όλες την απαντήσεις ενός χρήστη στο Stack Exchange
<code>/users/{id}/associated/</code>	Το χρησιμοποιώ με σκοπό να βρω όλες της σελίδες από το δίκτυο Stack Exchange στις οποίες ο χρήστης είναι μέλος.

Πίνακας 5.4.1

Βλέποντας τον πίνακα 5.4.1 διακρίνουμε ότι χρησιμοποιούμε το API `/me?access_token`. Το `access_token` είναι το token που παράγεται μετά τον OAuth έλεγχο ταυτότητας. Για τον έλεγχο ταυτότητας χρησιμοποιήσαμε το `client id` που μας δόθηκε. Μετά τον επιτυχή έλεγχο ταυτότητας παίρνουμε το `access_token` το οποίο χρειαζόμαστε για να συλλέξουμε το `unique id` του user που έχει για τη Stack Exchange πλατφόρμα.

Τα APIs του Stack Exchange παίρνουν ως Query παράμετρο το «site», στο οποίο εισάγουμε την ιστοσελίδα που θέλουμε να πάρουμε τις πληροφορίες. Για παράδειγμα «stackoverflow» αν θέλουμε να πάρουμε πληροφορίες για το Stack Overflow. Αυτό όμως προκαλεί πρόβλημα αφού το `id` δεν είναι το ίδιο για όλα τα site ενός χρήστη, επομένως

ανάλογα με το site που θέλουμε να πάρουμε πληροφορία θα πρέπει να εισάγουμε και το σωστό id.

Για να λύσουμε αυτό το πρόβλημα χρησιμοποιήσαμε το API `/users/{id}/associated` όπου εισάγουμε ως παράμετρο το unique id που πήραμε από το `/me` και παίρνουμε ως αποτέλεσμα όλες τις ιστοσελίδες όπως ο χρήστης είναι ενεργός, μαζί με το διαφορετικό id της κάθε μίας.

Κεφάλαιο 6

Ανάπτυξη Εφαρμογής

6.1 Προδιαγραφές και Απαιτήσεις	25
6.2 Υλοποίηση Web Εφαρμογής	28
6.2.1 Αρχιτεκτονική	28
6.2.2 Βάση Δεδομένων	29
6.3 API Web Services	30
6.4 Υλοποίηση Android Εφαρμογής	31

6.1 Προδιαγραφές και Απαιτήσεις

Η Android εφαρμογή αναπτύχθηκε με την χρήση του IDE Android Studio και στην γλώσσα προγραμματισμού Kotlin. Υποστηρίζει εκδόσεις Android από την 7.1.1 μέχρι και την τελευταία που είναι η 11. Η Android εφαρμογή μου ονομάζεται «Expertise Analyzer Mobile». Έγινε προσεχτική σχεδίαση, αξιολόγηση αναγκών και απαιτήσεων μέχρι την τελική της υλοποίηση.

Στην συγκεκριμένη Android εφαρμογή προσπάθησα να δώσω μεγάλη βαρύτητα στη διεργασία επαφής και στην αλληλεπίδραση με τον χρήστη. Ακολούθησα όλα τα πρότυπα σχεδίασης της Google που έχουν να κάνουν με το γραφικό περιβάλλον του χρήστη και σωστές πρακτικές προγραμματισμού.

Η εφαρμογή απευθύνεται σε χρήστες των GitHub και Stack Exchange, οι οποίοι στο μεγάλο τους μέρος είναι άτομα της Πληροφορικής. Υπάρχει φυσικά και ένα μέρος των χρηστών οι οποίοι δεν έχουν σχέση με την Πληροφορική και χρησιμοποιούν το δίκτυο του Stack Exchange, το οποίο αποτελείται από πολλούς ιστότοπους ερωτοαπαντήσεων.

Ο χρήστης μπορεί να εισάγει τα στοιχεία του στην Android εφαρμογή και καταχωρείται στο σύστημα.

Το σύστημα απευθύνεται και σε εργοδότες οι οποίοι αναζητούν μηχανικούς λογισμικού για να προσλάβουν. Μέσα από το σύστημα τους παρέχεται η δυνατότητα να αναζητήσουν μηχανικούς λογισμικού, βλέποντας τα στοιχεία τους αναλυτικά.

Για να μπορεί κάποιος εργοδότης να επικοινωνήσει με ένα χρήστη θα πρέπει να υπάρχουν τα κατάλληλα στοιχεία επικοινωνίας. Γι' αυτό το λόγο αν κάποιος χρήστης θέλει να χρησιμοποιήσει το σύστημα μας χρειάζεται να κάνει εγγραφή με ένα email.

Οι κύριες λειτουργίες της Android εφαρμογής καθορίζονται ως εξής:

- Εγγραφή χρήστη στο σύστημα (Register)
- Εισδοχή χρήστη στο σύστημα (Login)
- Εμφάνιση τις πρώτες πέντε γλώσσες προγραμματισμού που χρησιμοποίησε στο GitHub
- Εμφάνιση όλων των γλωσσών προγραμματισμού με σκοπό την αναζήτηση των χρηστών που τις χρησιμοποιούν
- Εμφάνιση όλων των ιστότοπων του δικτύου Stack Exchange που ο χρήστης έχει λογαριασμό και παρουσίαση για κάθε ένα ξεχωριστά τα badges που έχει.
- Εμφάνιση όλων των repositories που ο χρήστης έχει συνυμφαίρει καθώς και τις γλώσσες προγραμματισμού που χρησιμοποίησε.
- Εμφάνιση όλων των χρηστών του συστήματος καθώς και τις ανάλογες πληροφορίες που σχετίζονται με αυτούς.
- Ανταλλαγή πληροφοριών μεταξύ της Android εφαρμογής και της Web εφαρμογής.

Η Web εφαρμογή αναπτύχθηκε με την χρήση του IDE IntelliJ IDEA και στην γλώσσα προγραμματισμού Kotlin με την χρήση JVM (1.8). Έγινε προσεχτική σχεδίαση, αξιολόγηση αναγκών και απαιτήσεων μέχρι την τελική της υλοποίηση.

Plugins	Version
org.springframework.boot	2.4.1
io.spring.dependency-management	1.0.10.RELEASE

Πίνακας 6.1.1

Dependencies	Version
org.springframework.boot:spring-boot-starter-data-jpa	2.4.1
org.springframework.boot:spring-boot-starter-tomcat	2.4.1
org.springframework.boot:spring-boot-starter-web	2.4.1
org.jetbrains.kotlin:kotlin-reflect	1.4.21
org.jetbrains.kotlin:kotlin-stdlib-jdk8	1.4.21
mysql:mysql-connector-java	8.0.22
com.squareup.retrofit2:retrofit	2.9.0
com.squareup.retrofit2:converter-gson	2.9.0

Πίνακας 6.1.2

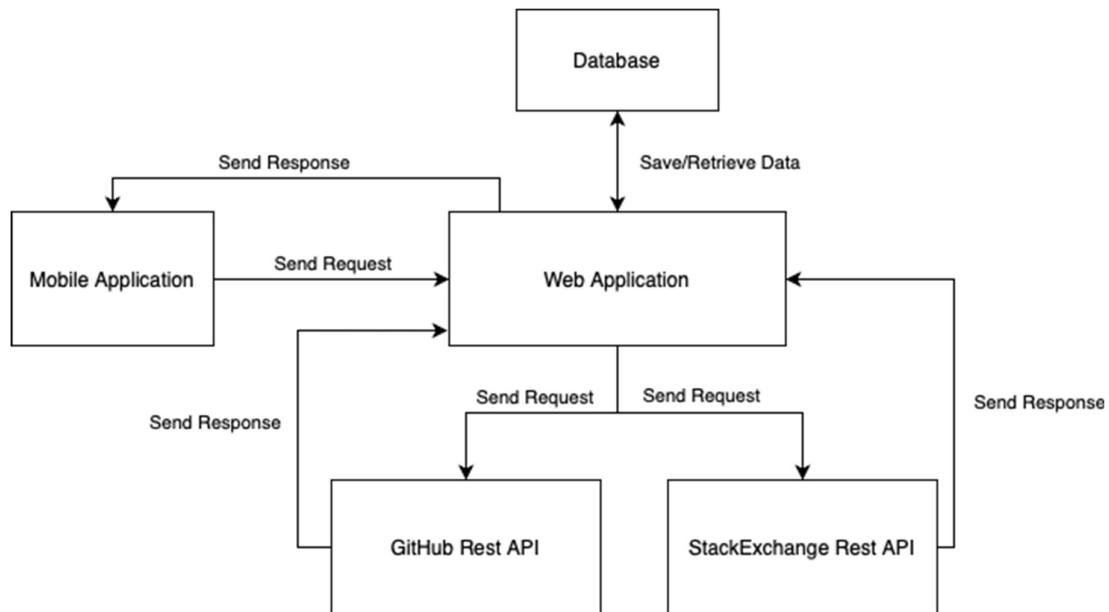
Στους πιο πάνω πίνακες 6.1.1 και 6.1.2 βλέπουμε τα Plugins και τα Dependencies που χρησιμοποιήθηκαν για την δημιουργία της Web εφαρμογής.

Οι κύριες λειτουργίες της Web εφαρμογής καθορίζονται ως εξής:

- Εξόρυξη και αποθήκευση πληροφοριών του χρήστη σχετικά με το GitHub
- Εξόρυξη και αποθήκευση πληροφοριών του χρήστη σχετικά με το δίκτυο Stack Exchange
- Ενημέρωση των πληροφοριών του χρήστη αν υπάρχουν αλλαγές κάθε φορά που εισέρχεται στο σύστημα
- Ανταλλαγή πληροφοριών μεταξύ της Android εφαρμογής και της Web εφαρμογής
- Αποθήκευση προσωπικών πληροφοριών του χρήστη (email και password)
- Εγγραφή χρήστη στο σύστημα
- Εισδοχή χρήστη στο σύστημα

6.2 Υλοποίηση Web Εφαρμογής

6.2.1 Αρχιτεκτονική



Σχήμα 6.2.1.1

Το σύστημα σχεδιάστηκε με βάση την αρχιτεκτονική πελάτη – εξυπηρετητή, αφού ο πελάτης χρησιμοποιώντας την Android εφαρμογή στέλνει αιτήματα στον εξυπηρετητή και ο εξυπηρετητής ανταποκρίνεται πίσω στον πελάτη εμφανίζοντας του τα κατάλληλα αποτελέσματα. Ο πελάτης στην δική μας περίπτωση είναι η Android εφαρμογή και ο εξυπηρετητής η Web εφαρμογή.

Η Web εφαρμογή στην συνέχεια συμπεριφέρεται και αυτή ως πελάτης αφού στέλνει αιτήματα στον εξυπηρετητή (GitHub ή Stack Exchange) και ο εξυπηρετητής ανταποκρίνεται πίσω στον πελάτη εμφανίζοντας τα κατάλληλα αποτελέσματα. Στην συνέχεια η web εφαρμογή αποθηκεύει τα αποτελέσματα στην βάση δεδομένων και απαντάει πίσω στην Android εφαρμογή τα καινούρια αποτελέσματα.

Επομένως η Android εφαρμογή επικοινωνεί με τα APIs της Web μας εφαρμογής η οποία στην συνέχεια επικοινωνεί με τα APIs του GitHub ή του Stack Exchange.

6.2.2 Βάση Δεδομένων

Η βάση δεδομένων περιέχει τους πίνακες user, stack_exchange_associated_sites, stack_exchange_model, stack_exchange_model_stack_exchange_associated_sites, languages, badge_counts, git_hub_model και git_hub_model_repo_languages.

Πίνακας	Περιγραφή
badge_counts	Περιέχει τον αριθμό των badge (bronze, gold, silver) για κάθε ιστοσελίδα από το δίκτυο Stack Exchange
Git_hub_model	Περιέχει ένα id μαζί με τη εικόνα του προφίλ
user	Περιέχει το email και το password του χρήστη όπως επίσης και τα id των GitHub και Stack Exchange
stack_exchange_model_stack_exchange_associated_sites	Περιέχει την σχέση του stack_exchange_model_id και του stack_exchange_associated_sites_id. Για ένα Stack Exchange id μπορεί να υπάρχουν τουλάχιστο ένα associated_sites_id
Stack_exchange_model	Περιέχει το id του Stack Exchange
Stack_exchange_asslckated_sites	Υπάρχουν όλες οι πληροφορίες για κάθε ιστοσελίδα του δικτύου Stack Exchange
Languages	Περιέχει τις πληροφορίες σχετικά με το τις γλώσσες προγραμματισμού για κάθε χρήστη
Git_hub_model_repo_language	Περιέχει την σχέση του git_hub_model_id και του repo_language_id. Ένας χρήστης GitHub έχει τουλάχιστο μία γλώσσα προγραμματισμού

Πίνακας 6.2.2.1

6.3 API Web Services

Για να μπορέσουν οι δύο εφαρμογές να ανταλλάξουν πληροφορίες χρειάστηκε να υλοποιήσω APIs στην Web εφαρμογή.

API Interface	Method	Χαρακτηριστικά
/login	POST	Δέχεται ως παράμετρο στο BODY το email και το password του χρήστη.
/register	POST	Δέχεται ως παράμετρο στο BODY το email, το password, το firstName και το surname του χρήστη
/register/{id}	PUT	Δέχεται ως παράμετρο στο PATH το id του χρήστη και στο BODY το gitHubName του χρήστη
/register/{id}/stackoverflow/{accessToken}	POST	Δέχεται ως παράμετρο στο PATH το id του χρήστη και το accessToken (από τον oauth έλεγχο ταυτότητας)
/user	GET	Δέχεται ως παράμετρο στο QUERY «search» (optional) το όνομα του χρήστη και επιστέφει μία λίστα με τα στοιχεία των χρηστών που ικανοποιούν την συνθήκη «search».
/user/github/{id}	GET	Δέχεται ως παράμετρο στο PATH το id του χρήστη και επιστρέφει όλες τις πληροφορίες σχετικά με το GitHub
/user/languages/	GET	Επιστρέφει όλα τα στοιχεία των γλωσσών προγραμματισμού σε μορφή HashMap <String,Long>
/user/languages/{id}	GET	Δέχεται ως παράμετρο στο PATH το id της γλώσσας προγραμματισμού και επιστρέφει μία λίστα με τα στοιχεία των χρηστών.

Πίνακας 6.3.1

Από τον πιο πάνω πίνακα (6.3.1) βλέπουμε ένα «documentation» για τα APIs που υποστηρίζει η Web εφαρμογή.

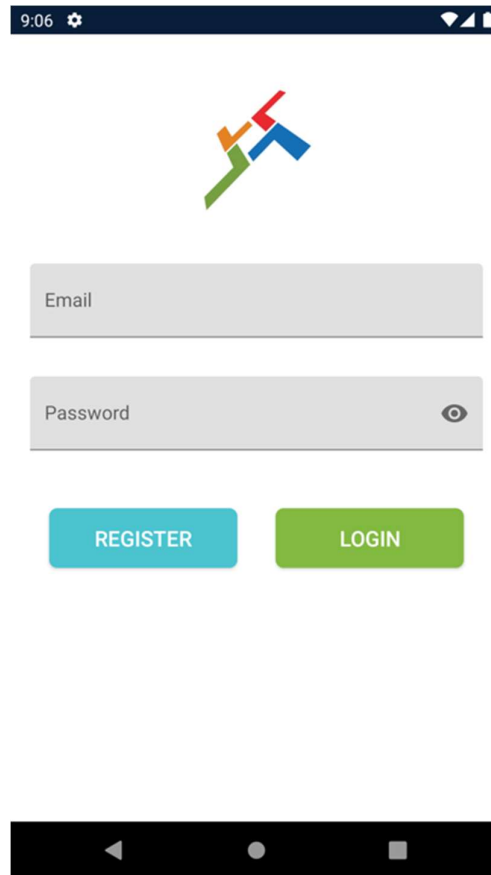
6.4 Υλοποίηση Android Εφαρμογής

Η εφαρμογή μου για τους σκοπούς της πτυχιακής μου υλοποιήθηκε μέσω του Android Studio και φέρει το όνομα «Expertise Analyzer Mobile» και είναι συμβατή μέχρι και τη τελευταία έκδοση του (Android 11).



Εικόνα 6.4.1

Ανοίγοντας την εφαρμογή βλέπουμε από την πιο πάνω εικόνα (6.4.1) το λογότυπο της εφαρμογής όπως επίσης και το λογότυπο του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου. Μόλις φορτωθούν τα δεδομένα της εφαρμογής πηγαίνει στην αρχική σελίδα.



Εικόνα 6.4.2

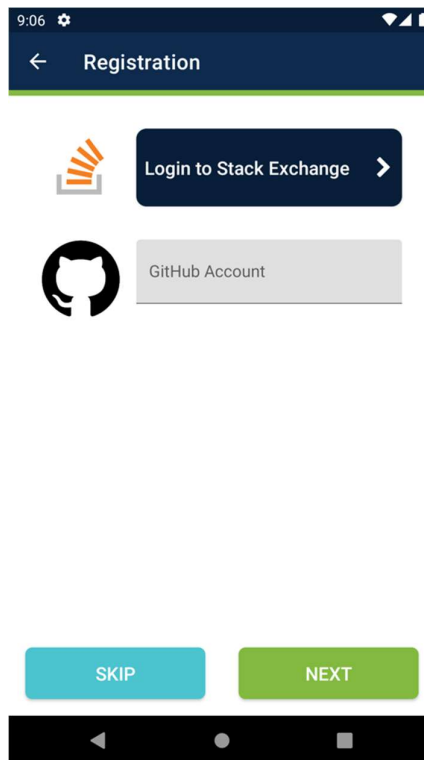
Στην συνέχεια ο χρήστης θα μεταφερθεί στην αρχική σελίδα της εφαρμογής (6.4.2). Όπως βλέπουμε από την πιο πάνω εικόνα ο χρήστης έχει την δυνατότητα να κάνει Login στο σύστημα μας με το Email του και το Password. Αν δεν έχει δημιουργήσει user στο σύστημα μας θα πρέπει να πάει στην σελίδα Register.

Αν ο χρήστης έχει ξανα κάνει login στο σύστημα μας τότε θα τον εισάγουμε στο σύστημα αυτόματα χωρίς να χρειαστεί να εισάγει ξανά τα στοιχεία του.

Εικόνα 6.4.3

Όταν πατήσει στο κουμπί Register ο χρήστης μεταφέρεται στην σελίδα Registration στην οποία θα πρέπει να συμπληρώσει τα στοιχεία του (First Name, Surname, Email Address, Password και Confirm Password). Το σύστημα μας ελέγχει αν το email υπάρχει ήδη στο σύστημα μας και εμφανίζει ένα μήνυμα λάθους αν ισχύει. Αν ο κωδικός είναι ο ίδιος και στις δύο περιπτώσεις (Password και Confirm Password) τότε ο user του δημιουργείται στο σύστημα μας και προχωρά στην επόμενη οθόνη.

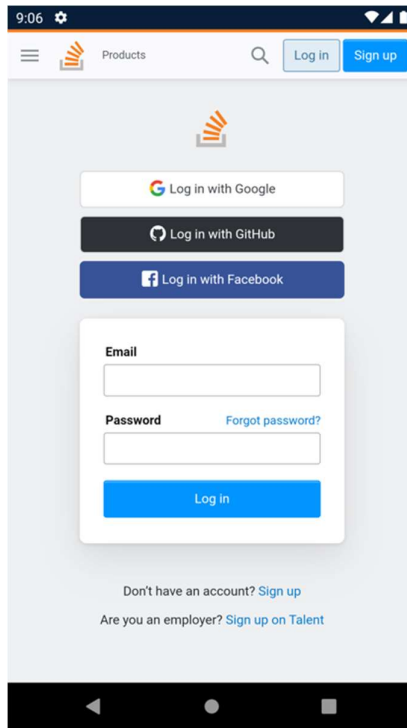
Στην επόμενη οθόνη (6.4.4) βλέπουμε ότι υπάρχουν 2 επιλογές «Skip» και «Next». Αν ο user πατήσει στο κουμπί «Skip» τότε η εγγραφή του στο σύστημα ολοκληρώνεται και μεταφέρεται στην αρχική οθόνη όπου μπορεί να κάνει Login. Ένας χρήστης μπορεί να παραλείψει να εισάγει τα στοιχεία του για το GitHub και το Stack Exchange αφού μπορεί να είναι εργοδότης. Αν θελήσει να τα εισάγει μπορεί να το κάνει και στην συνέχεια αφού υπάρχει αυτή η λειτουργία μέσα στην εφαρμογή, φτάνει φυσικά να κάνει Login.



Εικόνα 6.4.4

Όταν ο χρήστης πατήσει στο κουμπί “Login to Stack Exchange” μεταφέρεται στην ιστοσελίδα του Stack Exchange (6.4.5) όπου πρέπει να κάνει login στον λογαριασμό του και να δώσει άδεια στην εφαρμογή μας να συλλέξει τα στοιχεία που χρειαζόμαστε για να δημιουργήσουμε το προφίλ του.

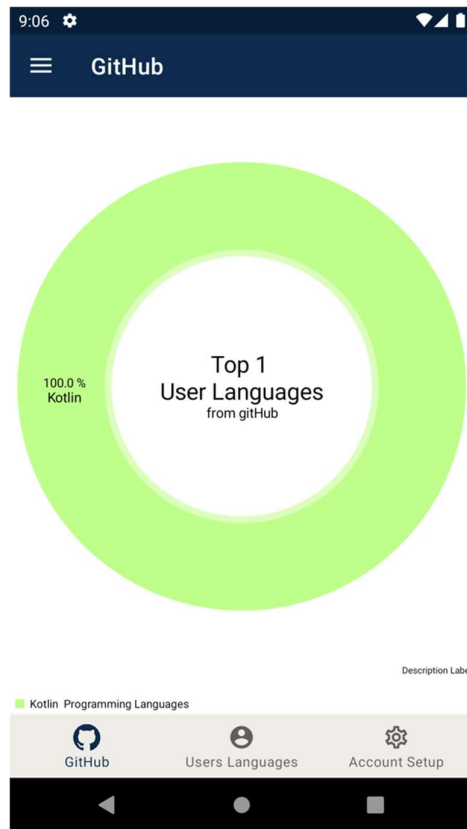
Σε αυτή την περίπτωση χρησιμοποιείται oauth έλεγχος ταυτότητας, του παρέχουμε συγκεκριμένο client id και redirection URL και μας επιστρέφει access_token.



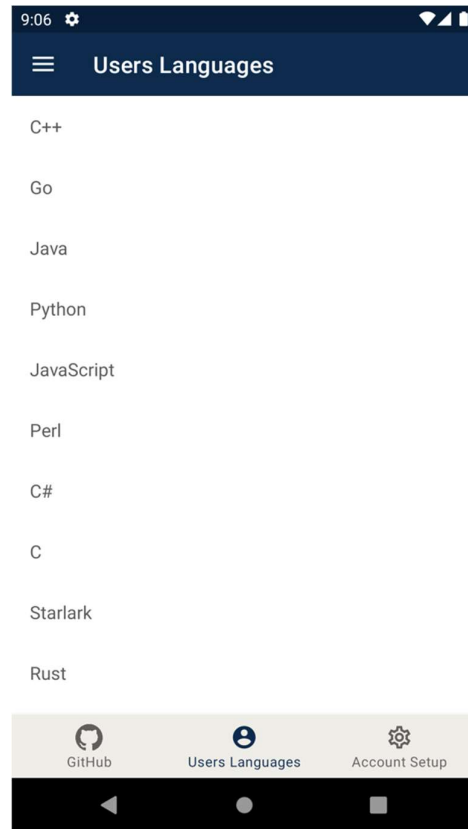
Εικόνα 6.4.5

Αφού ο χρήστης εισάγει τα στοιχεία του GitHub (user name) ολοκληρώνεται και η εγγραφή του στο σύστημα και είναι έτοιμος να κάνει Login στην εφαρμογή μας. Η εφαρμογή μας «ακούει» για το redirect URL που δηλώσαμε για τον oauth έλεγχο ταυτότητας και όταν προσπαθήσει να φορτώσει το URL το σταματούμε και κλείνουμε την οθόνη αφού το access_token ανακτήθηκε.

Στην συνέχεια ο χρήστης εισάγει το email και το password που όρισε και κάνει login στο σύστημα μας.



Εικόνα 6.4.6



Εικόνα 6.4.7

Στις πιο πάνω εικόνες (6.4.6 και 6.4.7) βλέπουμε τις δυο βασικές λειτουργίες που βλέπει ένας χρήστης μόλις κάνει login. Βλέπουμε ότι στο κάτω Navigation Bar υπάρχουν τρεις επιλογές, «GitHub», «Users Languages» και «Account Setup».

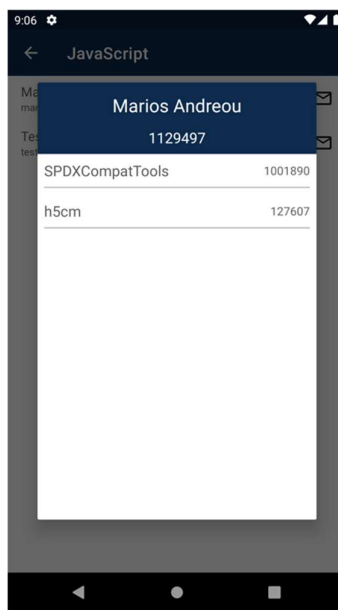
Στην πρώτη εικόνα βλέπουμε την επιλογή «GitHub» στην οποία ο χρήστης βλέπει στοιχεία σχετικά με το πρόσωπο του, βλέπει τις πρώτες 5 γλώσσες προγραμματισμού που χρησιμοποίησε σε όλα τα repositories του. Στην εικόνα βλέπουμε “Top 1” επειδή ο χρήστης που χρησιμοποίησα για το παράδειγμα μας έχει μόνο ένα repository με μόνο μια γλώσσα προγραμματισμού (Kotlin).

Στην δεύτερη εικόνα (6.4.7) βλέπουμε την επιλογή «Users Languages», εδώ εμφανίζονται όλες οι γλώσσες προγραμματισμού που εισάχθηκαν κατά την δημιουργία των προφίλ των χρηστών του συστήματος μας. Επομένως για κάθε γλώσσα προγραμματισμού υπάρχει τουλάχιστον ένας χρήστης ο οποίος χρησιμοποίησε την γλώσσα σε ένα repository του. Αν επιλέξουμε μια γλώσσα προγραμματισμού θα

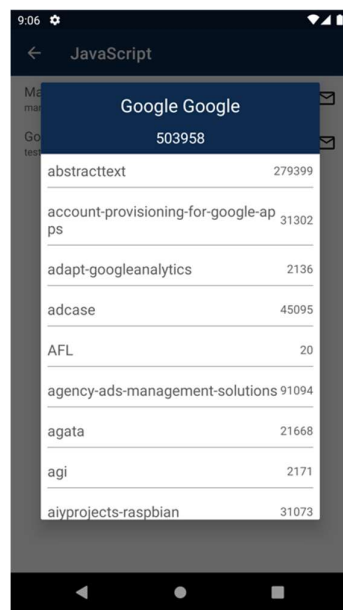
εμφανιστούν οι χρήστες κατά φθίνουσα σειρά με βάση την συνεισφορά τους για την συγκεκριμένη γλώσσα προγραμματισμού σε όλα τα repositories.



Εικόνα 6.4.8

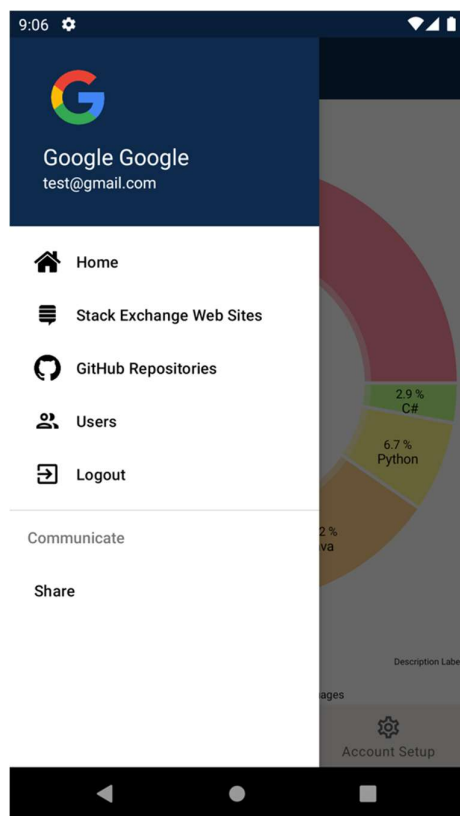


Εικόνα 6.4.9

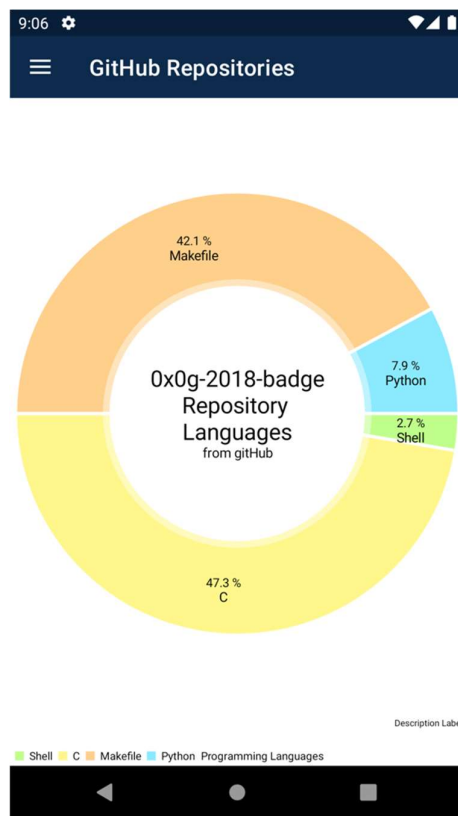


Εικόνα 6.4.10

Από τις πιο πάνω εικόνες (6.4.8 – 6.4.10) βλέπουμε ότι ο χρήστης «Marios Andreou» ενώ έχει συνεισφέρει σε λιγότερα repositories από τον χρήστη «Google Google» έχει χρησιμοποιήσει περισσότερο την γλώσσα προγραμματισμού «JavaScript» αφού το τελικό του score είναι «1129497» έναντι «503958». Από αυτή την λειτουργία της εφαρμογής μας κάποιος εργοδότης μπορεί να εντοπίσει χρήστες οι οποίοι ανταποκρίνονται στις απαιτήσεις του, αφού μπορεί να φιλτράρει τους χρήστες του συστήματος μας βάση μιας γλώσσας προγραμματισμού.



Εικόνα 6.4.11



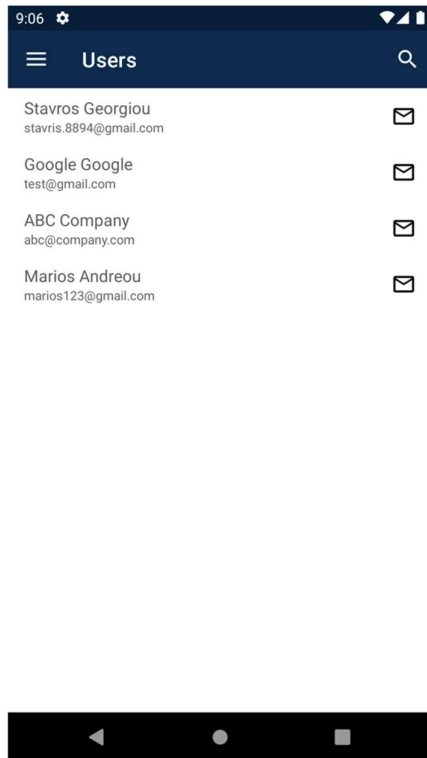
Εικόνα 6.4.12

Ο χρήστης από όλες τις σελίδες μπορεί να έχει πρόσβαση στο Navigation Drawer στα αριστερά της σελίδας (6.4.11) όπου μπορεί να δει την προφίλ εικόνα που έχει στο GitHub όπως επίσης το ονοματεπώνυμο του και το email του. Υπάρχουν επίσης και οι επιλογές να επιστρέψει στην “Home” σελίδα (6.4.6), να πάει στην σελίδα «Stack Exchange Web Sites» στην οποία μπορεί να δει τα badge του για κάθε ιστοσελίδα του δικτύου Stack Exchange. Επιπλέον υπάρχει η επιλογή να πάει στην σελίδα «GitHub Repositories» και «Users» αλλά και να κάνει Logout από το σύστημα μας.

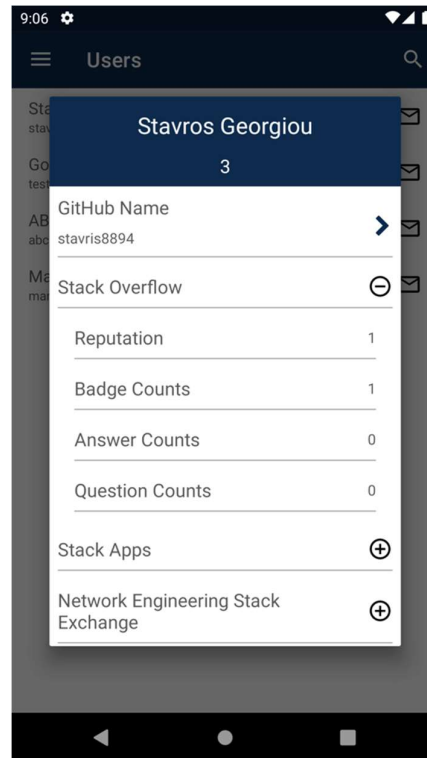
Τέλος υπάρχει η επιλογή «Share» με την οποία μπορεί να δημοσιεύσει το μήνυμα « Please checkout my account in Expertise Analyzer Mobile : @userEmail ».

Στην σελίδα «GitHub Repositories» (6.4.12) ο χρήστης μπορεί να δει όλα τα repositories που έχει καθώς και να δει ένα σχήμα πίττας με τα ποσοστά των γλωσσών προγραμματισμού που χρησιμοποιεί. Για παράδειγμα στην εικόνα 6.4.12 βλέπουμε ότι η

συνολική προσφορά του χρήστη στο «0x0g-2018-badge» repository είναι 47.3% σε γλώσσα C, 42.1% σε Makefile, 7.9% σε γλώσσα Python και 2.7% σε Shell.



Εικόνα 6.4.13



Εικόνα 6.4.14

Ο χρήστης στην σελίδα «Users» (6.4.13) μπορεί να εντοπίσει όλους τους χρήστες του συστήματος και πατώντας στο εικονίδιο του email να τους στείλει προσωπικό μήνυμα για να επικοινωνήσουν. Αυτή η λειτουργία είναι χρήσιμη τόσο για τους απλούς χρήστες αφού θα μπορούν να επικοινωνούν μεταξύ τους, όσο και για τους εργοδότες αφού θα μπορούν να στείλουν στους χρήστες email για να τους ενημερώσουν για τυχόν θέσεις εργασίας.

Επίσης στην σελίδα «Users» μπορεί κάποιος να δει περισσότερες πληροφορίες για ένα χρήστη (6.4.14), όπως για παράδειγμα το όνομα του user του στο GitHub, το Reputation που κατέχει σε μία ιστοσελίδα του δικτύου «Stack Exchange», το συνολικό αριθμό των badge, όπως επίσης και τους συνολικούς αριθμούς των ερωτήσεων και των απαντήσεων που έχει στις συγκεκριμένες ιστοσελίδες. Πατώντας στο κουμπί «GitHub Name»

εμφανίζονται όλα τα repository του χρήστη καθώς και τα ποσοστά ανά γλώσσα προγραμματισμού που συνείσφερε στο κάθε ένα repository ξεχωριστά.

Κεφάλαιο 7

Συμπεράσματα

7.1 Συμπεράσματα ΑΔΕ	40
7.2 Αξιολόγηση ΑΔΕ	41
7.3 Επεκτασιμότητα	42

7.1 Συμπεράσματα ΑΔΕ

Το θέμα που μου ανατέθηκε για τη διπλωματική μου εργασία ήταν η εξαγωγή προφίλ μηχανικών λογισμικού από πληροφορίες σε κοινωνικά δίκτυα και η ανάπτυξη εφαρμογής για συσκευές Android.

Μελετήσαμε και εξηγήσαμε την Web Εφαρμογή όπως επίσης και την Android Εφαρμογή που δημιουργήθηκαν για τους σκοπούς της παρούσας διπλωματικής εργασίας. Αποτελούν ένα ολοκληρωμένο σύστημα το οποίο προσφέρεται για δημόσια χρήση όχι μόνο από τους μηχανικούς λογισμικού αλλά και όποιο χρησιμοποιεί το GitHub και το Stack Exchange, αφού η υλοποίηση μας προσφέρεται για όλες τις σελίδες του δικτύου Stack Exchange.

Το σύστημα βοηθά το χρήστη να γνωρίσει και να κατανοήσει σε πιο επίπεδο βρίσκεται σε σχέση με άλλους χρήστες. Βοηθά επίσης και τους εργοδότες που αναζητούν μηχανικούς λογισμικού αφού τους παρέχεται η δυνατότητα αναζήτησης βάση γλώσσας προγραμματισμού. Η λεπτομερής γραφική παρουσίαση του προφίλ των χρηστών θα

βοηθήσει τόσο τους εργοδότες όσο και τους μηχανικούς λογισμικού αφού μπορούν να συγκρίνουν τα προφίλ τους εύκολα με άλλους χρήστες.

Η Android εφαρμογή και η Web εφαρμογή γράφτηκαν στην γλώσσα προγραμματισμού Kotlin με την οποία είχα σχετική εμπειρία.

7.2 Αξιολόγηση ΑΔΕ

Ο στόχος της πτυχιακής μου ήταν να εξάγω πληροφορίες από κοινωνικά δίκτυα και να τα παρουσιάσω τόσο σε χρήστες όσο και σε εργοδότες. Αυτό έγινε με επιτυχία αφού παρουσιάζονται με ακρίβεια όλα τα στοιχεία του χρήστη που υπάρχουν στο GitHub και σε όλους τους ιστότοπους του δικτύου Stack Exchange.

Ακόμα προστέθηκε η λειτουργία δημιουργίας λογαριασμού στο σύστημα μας με email και password. Αυτό βοηθά στον έλεγχο και στην ασφάλεια των APIs της Web εφαρμογής αφού δεν μπορεί οποιοσδήποτε να αντλήσει πληροφορίες αν δεν είναι χρήστης του συστήματος.

Το γραφικό περιβάλλον πρόκειται για ένα πολύ σημαντικό στοιχείο της εφαρμογής μας αφού δεν θέλουμε σε κανένα σημείο ο χρήστης να κουραστεί και έτσι επενδύθηκε πολύς χρόνος σε αυτό το κομμάτι με την δημιουργία λογοτύπων, εικονιδίων και ειδικά σχεδιασμένου περιβάλλοντος που εμφανίζονταν οι πληροφορίες.

Παρουσίασα την πτυχιακή μου εργασία σε δύο συμφοιτητές μου και τους ζητήθηκε να αξιολογήσουν την εφαρμογή σε συγκεκριμένα πεδία. Αυτά είναι η κατανόηση, η ευχρηστία και η χρησιμότητα του συστήματος. Και στις δυο περιπτώσεις έλαβα θετικά σχόλια τόσο για την κατανόηση του συστήματος όσο και για την ευχρηστία αφού δήλωσαν ότι βρήκαν εύκολη και ευχάριστη την πλοήγηση στο σύστημα. Μπόρεσαν να κάνουν εγγραφή και σύνδεση στο σύστημα χωρίς κάποιο πρόβλημα. Ο ένας δήλωσε ότι βρήκε οικεία την πλοήγηση της εφαρμογής αφού όπως ανέφερε το σύστημα έχει αρκετά κοινά χαρακτηριστικά με άλλες Android εφαρμογές που χρησιμοποιεί. Αυτό ευθύνεται γιατί ακολουθήθηκαν όλες οι κατευθυντήριες γραμμές για ανάπτυξη εφαρμογής από την Google. Στην χρησιμότητα του συστήματος δήλωσαν και οι δύο ότι θα μπορούσαν να

χρησιμοποιήσουν την εφαρμογή με σκοπό την δικτύωση τους με άλλους μηχανικούς λογισμικού με σκοπό να ανταλλάξουν διάφορες απορίες ή και γνώση.

Τέλος ένα σχόλιο που έλαβα ήταν η υλοποίηση ενός φόρουμ με βάση την χώρα των μηχανικών λογισμικού που βρίσκονται με σκοπό την δικτύωση τους με άτομα που ζουν στην ίδια χώρα.

7.3 Επεκτασιμότητα

Όπως σε όλα τα συστήματα υπάρχουν περιθώρια βελτίωσης και επέκτασης. Συγκεκριμένα μπορούν να προστεθούν αρκετές λειτουργίες που θα μπορέσουν να βελτιώσουν την εμπειρία χρήσης του συστήματος από τους χρήστες.

Για αρχή θα μπορούσε να γίνει η εφαρμογή και για iOS συσκευές αφού η Web εφαρμογή χρησιμοποιεί APIs τα οποία μπορούν να χρησιμοποιηθούν χωρίς δυσκολία και από άλλα συστήματα. Όπως επίσης θα μπορούσε να γίνει και μια ιστοσελίδα με την χρήση του Framework Angular το οποίο θα χρησιμοποιήσει τα υπάρχον APIs της Web εφαρμογής. Αυτόματα το σύστημα μας μεγαλώνει αφού αυξάνεται και το πλήθος των ατόμων που μπορούν να την χρησιμοποιήσουν.

Να γίνει καλύτερη η εμπειρία του εργοδότη με περισσότερες επιλογές αναζήτησης όπως επίσης και η επιλογή να κάνει block χρήστες τους οποίους δεν θέλει να εμφανίζονται στις αναζητήσεις του. Αυτό θα βοηθήσει τους εργοδότες να βρουν τα άτομα που χρειάζονται για τις ανάγκες τους.

Μπορεί να εισαχθεί σύστημα «Alert» το οποίο θα μπορεί να ενημερώνει τους χρήστες με ειδοποιήσεις, είτε με email είτε με Push Notification στην συσκευή του, ότι κάποιος άλλος χρήστης παρακολούθησε το προφίλ του.

Ακόμη θα μπορούσε να εισαχθεί σύστημα ανταλλαγής μηνυμάτων μέσω της εφαρμογής για να υπάρχει καλύτερη εμπειρία χρήσης του συστήματος με τον χρήστη.

Επιπρόσθετα, εάν το σύστημα χρησιμοποιηθεί από ένα μεγάλο αριθμό χρηστών θα μπορούν να συλλεχτούν πολλά στοιχεία σε μεγάλους όγκους δεδομένων. Επομένως τα δεδομένα θα αναλύονται και θα εξάγουν χρήσιμες πληροφορίες όπως και συμπεράσματα. Όπως επίσης και να αποθηκεύονται όλες οι ενέργειες των χρηστών με σκοπό την βελτίωση του συστήματος, αφού θα μπορούν να αναλύσουν ποιες λειτουργίες του συστήματος χρησιμοποιούνται το περισσότερο και ποιες όχι.

Τέλος μπορεί να εισαχθούν οι επιλογές σύνδεσης του λογαριασμού του χρήστη με το Facebook, το Google και την Apple με σκοπό τον εντοπισμό φύλων.

Βιβλιογραφία

- [1] Android, Design for Android
<https://developer.android.com/design>
- [2] Spring Boot
<https://spring.io>
- [3] Wikipedia “GitHub”
<https://en.wikipedia.org/wiki/GitHub>
- [4] Material Design
<https://material.io/design>
- [5] Android, “Android Studio – Documentation for app developers”
<https://developer.android.com/docs/>
- [6] Spring Boot, Tutorial
https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm
- [7] Stack Exchange API
<https://api.stackexchange.com/docs>
- [8] GitHub, REST API
<https://docs.github.com/en/free-pro-team@latest/rest>
- [9] Kotlin, Documentation
<https://kotlinlang.org/docs/reference/>
- [10] Spring, Serving Web Content with Spring MVC
<https://spring.io/guides/gs/serving-web-content/>

- [11] MySQL
<https://en.wikipedia.org/wiki/MySQL>
- [12] Android, Retrofit
<https://square.github.io/retrofit/>