Εφαρμογή ενημέρωσης για τα δρομολόγια των αστικών συγκοινωνιών της πόλης του Βόλου, σε περιβάλλον Android.

Μπακίρης Εμμανουήλ

Προχωρημένη Διαχείριση Δεδομένων 2019-20
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Πανεπιστήμιο Θεσσαλίας, Βόλος
ebakiris@e-ce.uth.gr

Περίληψη Στην παρούσα εργασία, περιγράφεται η λειτουργία μιας εφαρμογής Android για smartphones, καθώς και τα επιμέρους σημεία της υλοποίησης της. Επιπλέον, παρουσιάζονται και στιγμιότυπα από την εκτέλεση της εφαρμογής καθώς επίσης και αναλυτικά συμπεράσματα για την αξιοπιστία της. Τέλος, δίνεται και ξεχωριστή βαρύτητα στο NoSQL σύστημα διαχείρισης βάσεων δεδομένων που χρησιμοποιήθηκε όπως και τα σημαντικότερα μέρη σχολιασμένου κώδικα, με αναλυτική εξήγηση.

Λέξεις Κλειδιά: android, redis server, redis client, lettuce API, NoSQL database

1 Περιγραφή λειτουργίας της εφαρμογής

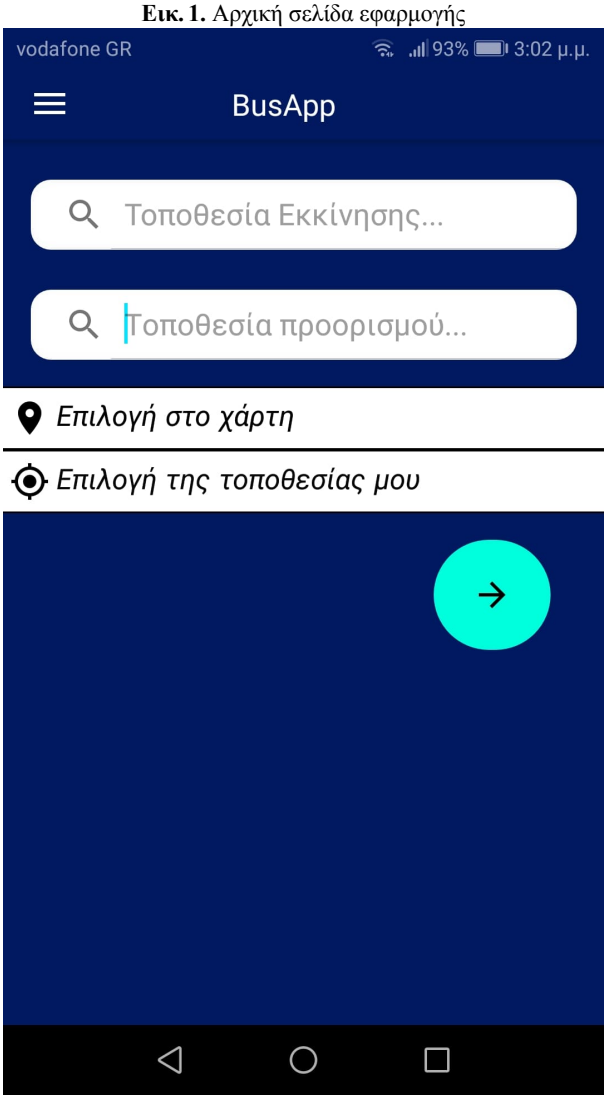
Αρχικά, να τονίσουμε ότι η εφαρμογή αναπτύχθηκε στην πλατφόρμα Android Studio, η οποία είναι δωρεάν και επιτρέπει στους προγραμματιστές την εύκολη και γρήγορη ανάπτυξη εφαρμογών Android για έξυπνα κινητά, τα οποία υποστηρίζουν το συγκεκριμένο λειτουργικό σύστημα. Στην υποενότητα 1.1 παρουσιάζονται τα βασικά μέρη της εφαρμογής με τις επιμέρους λειτουργίες τους και στην υποενότητα 1.2 αναφέρονται κάποιες υποθέσεις και διευκρινήσεις σχετικά με τα δεδομένα που χρησιμοποιήθηκαν στην συγκεκριμένη υλοποίηση.

1.1 Μέρη της εφαρμογής

Η εφαρμογή αποτελείται από δύο βασικά μέρη. Το πρώτο είναι αυτό στο οποίο ο χρήστης εισάγει μία τοποθεσία εκκίνησης της διαδρομής, μια τοποθεσία προορισμού και η εφαρμογή χρησιμοποιεί αυτά τα δεδομένα και εμφανίζει στην οθόνη την καλύτερη επιλογή λεωφορείου για την ολοκλήρωση της διαδρομής.

Το δεύτερο μέρος της εφαρμογής αποτελείται από διάφορες επιλογές με τις γραμμές λεωφορείων και το αποτέλεσμα είναι το δρομολόγιο με τις ώρες και τις μέρες της εκάστοστε γραμμής λεωφορείου.

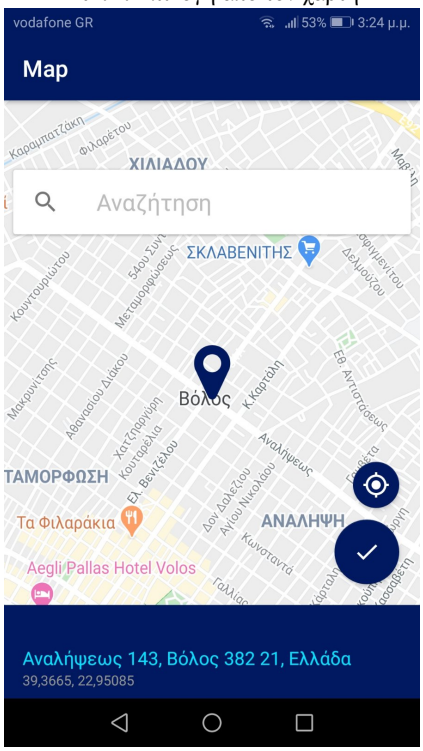
Εικ. 1. Αρχική σελίδα εφαρμογής



Μέρος πρώτο Με την εκκίνηση της εφαρμογής παρουσιάζεται στον χρήστη η Εικόνα 1. Αρχικά, αυτός θα επιλέγει την τοποθεσία εκκίνησης βασιζόμενος σε 2 επιλογές.Να επιλέξει από τον χάρτη ή να επιλέξει την τοποθεσία του.

Εάν αποφασίσει να επιλέξει από τον χάρτη εμφανίζεται η Εικόνα 2. Δίνεται η δυνατότητα στον χρήστη να μετακινήσει την πινέζα σε όποιο σημείο θέλει αυτός πάνω στο χάρτη ή ακόμα και να κάνει αναζήτηση ένα σημείο.

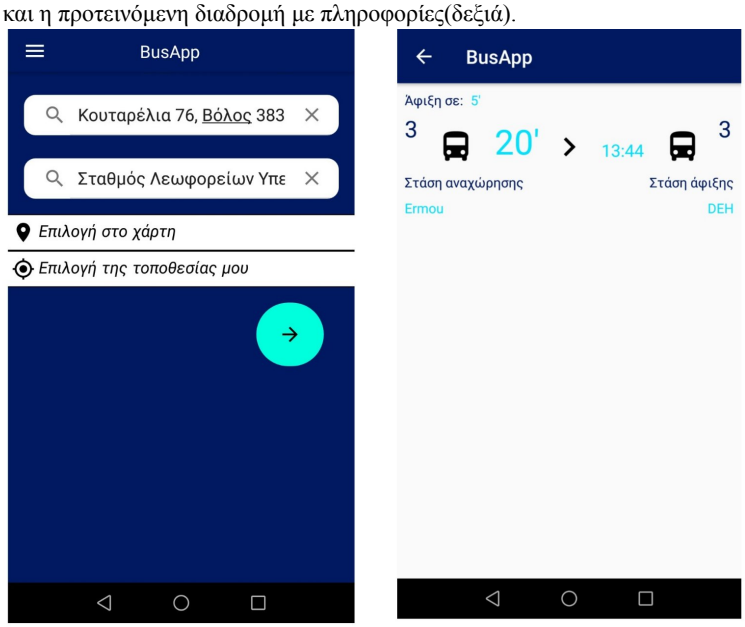
Εικ. 2. Επιλογή από τον χάρτη



Εφόσον περάσει την διαδικασία επιλογής τοποθεσίας εκκίνησης αλλά και προορισμού επιλέγοντας δύο σημεία στο χάρτη, παίρνει ένα αποτέλεσμα, την εικόνα 3(αριστερά), όπου έχουν συμπληρωθεί αυτόματα οι τοποθεσίες και έτσι μπορούμε να συνεχίσουμε περαιτέρω στην διαδικασία αναζήτησης της κατάλληλης διαδρομής. Παρόμοια είναι και η λειτουργία κατά την επιλογή της τοποθεσίας του χρήστη, όπου ζητείται πρόσβαση στην τοποθεσία της συσκευής και εν συνέχεια συμπληρώνεται η τοποθεσία εκκίνησης κατάλληλα. Λεπτομέρειες σχετικά με τον κώδικα και πως αντλούνται τα απαιραίτητα δεδομένα αναλύονται λεπτομερώς στην ενότητα 2.

Πατώντας το βελάκι, ο χρήστης μεταφέρεται στην εικόνα 3(δεξιά) όπου βλέπει τις εξής πληροφορίες : την γραμμή λεωφορείου που είναι κατάλληλη για την διαδρομή που επέλεξε(από τοποθεσία εκκίνησης σε προορισμού), χρόνος άφιξης λεωφορείου στην

Εικ. 3. Ο χρήστης είναι έτοιμος για την αναζήτηση διαδρομής(αριστερά).Το αποτέλεσμα της αναζήτησης και η προτεινόμενη διαδρομή με πληροφορίες(δεξιά).



κοντινότερη στάση, συνολικός χρόνος διεκπαιρέωσης της διαδρομής, η στάση αναχώρησης, υπολογισμένη εκτιμώμενη ώρα άφιξης στον προορισμό και στάση άφιξης στον προορισμό.Λεπτομέρειες σχετικά με την υλοποίηση και την επιλογή της καταλληλότερης γραμμής παρουσιάζονται στις ενότητες 2 και 3.

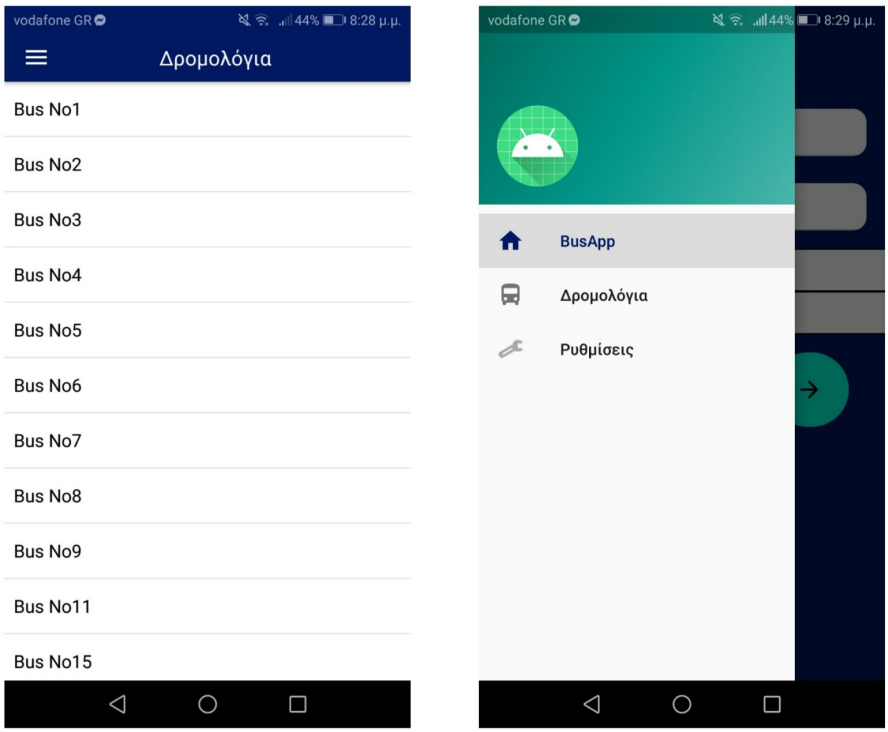
Μέρος δεύτερο Στο δεύτερο μέρος της εφαρμογής, ο χρήστης μπορεί να επιλέξει εκείνη την γραμμή λεωφορείου που θέλει, για να δει πληροφορίες όσον αφορά τις στάσεις και τις ώρες εκκίνησης από την αφετερία.Για να βρεθεί σε αυτήν την εικόνα 4(αριστερά), επιλέγει την κατάλληλη επιλογή μέσω του μενού που έχει διαμορφωθεί και φαίνεται στην εικόνα 4(δεξιά).

Στην συνέχεια μετά την επιλογή της γραμμής εμφανίζεται η εικόνα 5(αριστερά). Όπως γνωρίζουμε κάθε γραμμή έχει αφετερία και τέρμα. Έτσι, έχουμε δυο κατευθύνσεις με ξεχωριστές στάσεις και ωράρια.Επομένως, ο χρήστης θα έχει και τις κατάλληλες επιλογές.Η λειτουργία εμφάνισης των στάσεων και των ωραρίων είναι έτσι σχεδιασμένη ούτως ώστε αν δεν χωράνε στην οθόνη ο χρήστης μπορεί να σκρολλάρει προς τα κάτω και πάνω(εικόνα 5 κέντρο και δεξιά).

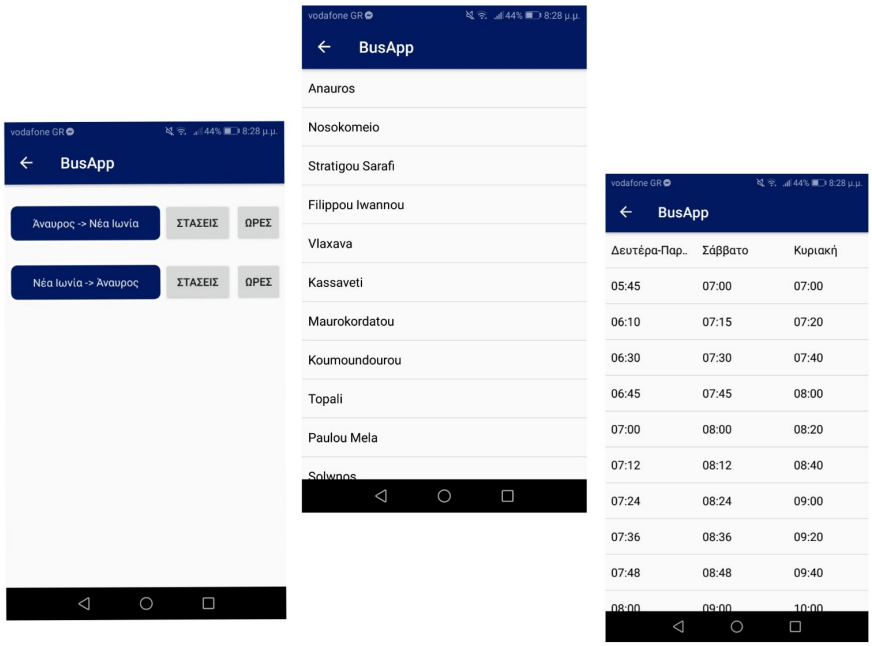
1.2 Υποθέσεις και διευκρινήσεις σχετικά με τα δεδομένα

Τα δεδομένα αντλήθηκαν από το επίσημο ιστότοπο των αστικών συγκοινωνιών της πόλης του Βόλου[1].Λόγω έλλειψης δεδομένων όσο αφορά τις τοποθεσίες των στά-

Εικ. 4. Οι γραμμές των αστικών λεωφορείων στην πόλη του Βόλου αριστερά και δεξιά το μενού πλοήγησης.



Εικ. 5. Τέρμα αριστερά οι δυο κατευθύνσεις της εκάστοτε γραμμής.Στο κέντρο όταν επιλέγουμε τις στάσεις από Άναυρο για Νέα Ιωνία και δεξιά οι ώρες της ίδιας γραμμής.



σεων(longitude,latitude), αυτά πάρθηκαν με πειραματικό τρόπο, διασχίζωντας συγκεκριμένα τη γραμμή 3 (και τις δυο κατευθύνσεις).Επομένως, η εφαρμογή λαμβάνει υπόψιν μόνο τις συντεταγμένες των τοποθεσιών των στάσεων της γραμμής 3.

2 Υλοποίηση - Σύστημα Διαχείρισης Δεδομένων - Εγκατάσταση

2.1 Περιβάλλον ανάπτυξης της εφαρμογής

Η ανάπτυξη της εφαρμογής πραγματοποιήθηκε στην πλατφόρμα Android Studio. Αυτή προσφέρει πλήθος εργαλείων που αυτοματοποιεί και διευκολύνει τον προγραμματισμό τόσο του γραφικού περιβάλλοντος, όσο και των λειτουργιών αυτής. Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η Java. Επίσης, τα αρχεία XML είναι αυτά που χρησιμοποιούνται για την διεπαφή χρήστη και την σχεδίαση της εφαρμογής, πως δηλαδή αυτή θα φαίνεται στον χρήστη.

2.2 Δομή των καταλόγων, των αρχείων και η λειτουργία τους

Αρχικά, κατά την δημιουργία ενός νέου project, το περιβάλλον Android Studio, δημιουργεί αυτόματα την δομή και τους καταλόγους που θα χρησιμοποιηθούν κατά την ανάπτυξη της εφαρμογής.Στην συγκεκριμένη εργασία δεν χρειάστηκε να δημιουργήσουμε άλλους καταλόγους, αυτοί που δημιουργήθηκαν αυτόματα από το περιβάλλον ήταν αρκετοί.Όσον αφορά τα αρχεία, δημιουργούνται κάποια βασικά για το ξεκίνημα, στην συνέχεια ο προγραμματιστής προσθέτει τα δικά του στον κατάλληλο φάκελο, όπως θα εξηγήσουμε παρακάτω.

Τα αρχεία που μας ενδιαφέρουν περισσότερο για τη συγκεκριμένη εργασία βρίσκονται στο μονοπάτι /Busapp/app/src/main.

Android Manifest Αυτό το αρχείο βρίσκεται στο μονοπάτι που αναφέρθηκε προηγουμένως. Είναι το αρχείο στο οποίο ουσιαστικά έχουμε δηλώσει τις Activities(.java αρχεία) καθώς και τις άδειες που θα ζητηθούν από το χρήστη κατά την πρώτη εκκίνηση της εφαρμογής.

Οι Activities είναι ουσιαστικά εκείνα τα αρχεία στα οποία ορίζεται η λειτουργία της εφαρμογής.Για κάθε διαφορετική οθόνη που εμφανίζεται, ένα αρχείο .java , η Activity δηλαδή είναι αυτή που διαχειρίζεται αυτή την οθόνη ανάλογα με τις ενέργειες του χρήστη ή θα μεταφέρει τα δεδομένα σε μία νέα Activity, η οποία θα έχει να διαχειριστεί άλλο layout(XML αρχείο).Αναλυτική εξήγηση θα δωθεί παρακάτω.

Οι άδειες, μπορεί να είναι άδεια παρακολούθησης τοποθεσίας, απαραίτητη ενεργοποίηση του Wifi ή των δεδομένων, πρόσβαση στα πολυμέσα της συσκευής κ.α.Στην παρούσα εργασία ζητάμε, ουσιαστικά, μόνο άδεια στην πρόσβαση της τοποθεσίας.

Activities - Java αρχεία Στον φάκελο /BusApp/app/src/main/java/com/example/busapp/ έχουμε μία κύρια Activity η οποία είναι η βασική της εφαρμογής και ρυθμίζει την εναλλαγή μέσω του menu και την εκκίνηση της εφαρμογής.

Στον φάκελο /BusApp/app/src/main/java/com/example/busapp/ui/home έχουμε 3 αρχεία.Ουσιαστικά μόνο τα 2 μας ενδιαφέρουν ιδιαίτερα.Στο HomeFragment.java ορίζεται η λειτουργία της αρχικής οθόνης, δηλαδή αυτής της εικόνας 1.Το αρχείο Result Activity.java είναι εκείνο που είναι υπεύθυνο για το τελικό αποτέλεσμα, δηλαδή την δεξιά οθόνη στην εικόνα 3.Αυτό το αρχείο λαμβάνει τα δεδομένα εισόδου από το Home Fragment.java και αναλόγως τυπώνει τα αποτελέσματα στο layout αρχείο που είναι συνδεδεμένο με την ResultActivity.

Στον φάκελο /BusApp/app/src/main/java/com/example/busapp/ui/gallery έχουμε 5 αρχεία.Το GalleryFragment διαχειρίζεται την οθόνη αριστερά της εικόνας 4, τα Bus RouteActivity, BusStationActivity και BusScheduleActivity αναφέρονται αντίστοιχα στα layout αριστερά, κέντρο και δεξιά της εικόνας 5.

Layout - XML αρχεία Κάθε Activity αναφέρεται σε ένα layout αρχείο,το οποίο προβάλλεται στην οθόνη του χρήστη και συνολικά όλα συνδέονται μέσω των Activities διαμορφώνοντας την εφαρμογή.

Στον φάκελο /BusApp/app/src/main/res/layout είναι τα κύρια layout αρχεία.Από την ονομασία εύκολα καταλαβαίνουμε σε ποιά Activity αναφέρονται.Για παράδειγμα το αρχείο fragment-home αναφέρεται στη HomeFragment.java και συνδυαστικά προβάλλουν την εικόνα 1, το αρχείο activity-busschedule αναφέρεται στην BusScheduleActivity και συνδυαστικά προβάλλεται και διαχειρίζεται η δεξιά οθόνη της εικόνας 5 κ.ο.κ.

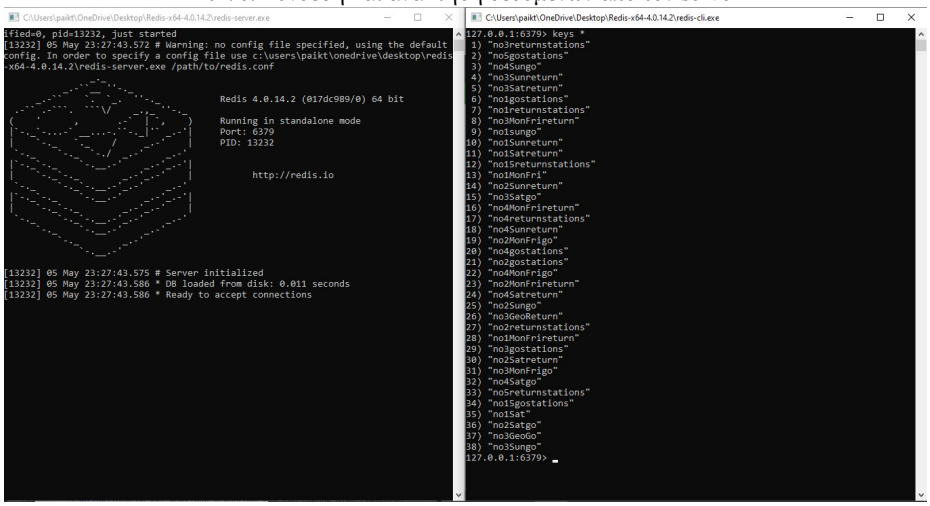
Στον φάκελο /BusApp/app/src/main/res/ υπάρχουν και κάποια άλλα αρχεία. Αυτά είναι για την αυτοματοποίηση της διαδικασίας.Για παράδειγμα το /res/value/colors ορίζονται τα χρώματα που χρησιμοποιούνται και μπορούμε να αναφερόμαστε σε αυτά μέσα από άλλα αρχεία.Στο αρχείο /res/value/strings ορίζονται κάποια strings με τις τιμές τους τα οποία χρησιμοποιούσαμε συχνά.Άλλοι φάκελοι περιέχουν εικόνες που χρησιμοποιήσαμε σε διάφορα layout(πχ το λεωφορείο στο activity-result layout, ή η εικόνα της τοποθεσίας και το βελάκι στην αρχική οθόνη).

2.3 Σύστημα Διαχείρισης Δεδομένων

Το σύστημα διαχείρισης δεδομένων που χρησιμοποιήθηκε είναι ένα είδος NoSQL βάσεων δεδομένων και συγκεκριμένα Key-value store.Το σύστημα Redis είναι αυτό που επιλέχτηκε για τη συγκεκριμένη εφαρμογή και κρίθηκε κατάλληλο.

Λήψη και εγκατάσταση Redis Στην επίσημη ιστοσελίδα του Redis αναφέρονται οδηγίες εγκατάστασης μόνο για Linux.Για την εγκατάσταση σε Windows, αρκεί η λήψη της τελευταίας έκδοσης από το GitHub[2]. Μετά την λήψη, μεταβαίνουμε στο φάκελο και απλά εκτελούμε το αρχείο redis-server.Έτσι, εκτελείται τοπικά ο server ο οποίος θα περιέχει τα δεδομένα που θα εισάγουμε, εικόνα 6(αριστερά).Εκτελώντας το αρχείο redis-client μπορούμε να εκτελέσουμε διάφορες εντολές εισαγωγής,διαγραφής ανάκτησης δεδομένων από τον server, εικόνα 6(δεξιά). Στο συγκεκριμένο στιγμιότυπο εκτελέστηκε η εντολή keys * , η οποία εμφανίζει όλα τα κλειδιά που υπάρχουν στη βάση. Στη προκειμένη περίπτωση, είναι λίστες και geodata, δηλαδή κλειδιά με συντεταγμένες και members.

Εικ. 6. Σύνδεση και ανάκτηση δεδομένων από τον server



Πρόσβαση στο server μέσω της εφαρμογής Για να έχουμε πρόσβαση στα δεδομένα από την εφαρμογή, χρειαζόμαστε ένα API του συστήματος Redis για γλώσσα προγραμματισμού Java.Από τον επίσημο ιστότοπο [3] μπορούμε να δούμε ποιο είναι κατάλληλο για εμάς.Υπάρχουν πολλά για τη συγκεκριμένη γλώσσα, οπότε επιλέχτηκε το Lettuce.Εύκολα γίνεται λήψη από το GitHub[5] και ακολουθώντας τις οδηγίες το εισάγουμε στο Android Studio[6].

Με οδηγό το documentation [4], μπορούμε να καλέσουμε τις απαραίτητες συναρτήσεις στα αρχεία μας για να συνδεθούμε στο server και να χρησιμοποιήσουμε τα δεδομένα.

Στην εικόνα 7, γραμμή 59 του κώδικα δημιουργούμε τον client που θα συνδεθεί στη βάση δεδομένων, στην γραμμή 60 δημιουργούμε την σύνδεση και στην γραμμή 61 ορίζουμε τις εντολές redis. Στην γραμμή 65 και 66 εκτελούμε εντολές redis τις syncCommands.llen και syncCommands.lindex. Με την πρώτη παίρνουμε το μέγεθος της λίστας και την δεύτερη το περιεχόμενο ενός κόμβου από τη λίστα με βάση το δείκτη.Στο συγκεκριμένο τμήμα κώδικα, αντλούμε τις στάσεις μιας συγκεκριμένης γραμμής για να τις εμφανίσουμε στον χρήστη, εικόνα 5(κέντρο).

2.4 Εγκατάσταση εφαρμογής

Αρχικά θα πρέπει να γίνει εγκατάσταση του Android Studio.Στην συνέχεια, από το μενού επιλέγουμε File->Open και επιλέγουμε το αρχικό κατάλογο της εφαρμογής δηλαδή BusApp. Για να την εγκαταστήσουμε στο κινητό πρέπει να γίνουν πρώτα κάποιες ρυθμίσεις.Στο κινητό πηγαίνουμε στις Ρυθμίσεις->Σύστημα->”Επιλογή για προγραμματιστές” και ενεργοποιούμε τον ”εντοπισμό σφαλμάτων USB”.Στην συνέχεια συνδέουμε το κινητό με το μηχάνημα(desktop,laptop) και στην ερώτηση που θα εμφανιστεί στο κινητό πατάμε ok. Για να εγκαταστήσουμε τώρα την εφαρμογή μέσω του Android

Εικ. 7. Σύνδεση και ανάκτηση δεδομένων από τον server στην εφαρμογή

```
public void connectToRedis(String[] stations){

    int i;

    RedisClient redisClient = RedisClient.create("redis://@192.168.1.10:6379/0");
    StatefulRedisConnection<String, String> connection = redisClient.connect();
    RedisCommands<String, String> syncCommands = connection.sync();

    if(position.contains("togo")) {
        i = 0;
        while (i < syncCommands.llen(stations[Integer.parseInt(String.valueOf(position.charAt(0)))])) {
            arrayList.add(syncCommands.lindex(stations[Integer.parseInt(String.valueOf(position.charAt(0)))], i));
            i++;
        }
    }
    else{
        i = 0;
        while (i < syncCommands.llen(stations[Integer.parseInt(String.valueOf(position.charAt(0)))])) {
            arrayList.add(syncCommands.lindex(stations[Integer.parseInt((String.valueOf(position.charAt(0))))], i));
            i++;
        }
    }

    connection.close();
    redisClient.shutdown();
}
```

Studio πατάμε το κουμπί ”Play” κάτω από τη γραμμή εργαλειών(θα πρέπει να έχει εμφανιστεί το όνομα της συσκευής αριστερά).

3 Αξιολόγηση εφαρμογής - Μελλοντική βελτίωση

Η εφαρμογή για να λειτουργήσει γρήγορα και αποτελεσματικά θα πρέπει να βασίζεται σε μεγάλο πλήθος δεδομένων. Κατά τη διάρκεια ανάπτυξης της παρουσιάστηκαν κάποιες ελλείψεις σε δεδομένα τα οποία επηρεάζουν, κυρίως, την αξιοπιστία της.

3.1 Αξιοπιστία

Η αξιοπιστία επηρεάζεται με ποίκιλους τρόπους.Αρχικά, δεν υπάρχουν δεδομένα για τις στάσεις σε συγκεκριμένες γραμμές, δεν αναφέρονται τα ονόματα των στάσεων για τις γραμμές 6,7,8,9,11 διότι ούτε στον επίσημο ιστότοπο των αστικών συγκοινωνιών Βόλου υπάρχουν.Επίσης, για τις υπόλοιπες γραμμές(εκτός της 3) δεν έχουν καταχωρηθεί οι συντεταγμένες των τοποθεσιών των στάσεων, το οποίο όμως μπορεί να γίνει μελλοντικά με πειραματκή παρακολούθηση των γραμμών.Επόμενως, στην παρούσα φάση αν μια άλλη γραμμή εκτός της 3, μπορεί να χρησιμοποιηθεί από τον πολίτη για τη βέλτιστη διαδρομή του, δεν θα εμφανιστεί στην εφαρμογή, παρά μόνο η βέλτιση χρησιμοποιώντας την γραμμή 3.

Από την άλλη, στον υπολογισμό του χρόνου άφιξης ενός λεωφορείου στη στάση αναχώρησης, θα μπορούσαμε να λάβουμε υπόψιν ένα ερώτημα που θα εμφανίζεται στην οθόνη του χρήστη που περιμένει σε προηγούμενη ή επόμενη στάση,ρωτώντας τον αν περιμένει ακόμα ή επιβιβάστηκε ,ούτως ώστε να ενημερώνεται αυτόματα η βάση με μέσους όρους άφιξης του λεωφορείου στις στάσεις αναχώρησης ή για την real-time ενημέ-

ρωση επιβατών που περιμένουν χρησιμοποιώντας την εφαρμογή σε επόμενες στάσεις, από επιβάτες που επιβιβάστηκαν ή περιμένουν σε προηγούμενες στάσεις.

Επιπλέον, θα μπορούσαμε να καταγράψουμε με πειραματική παρακολούθηση για κάθε διαφορετική μέρα της εβδομάδας,για διαφορετικές ώρες(ένα διάστημα το πρωί, μεσημέρι, απόγευμα και βράδυ) και για κάθε γραμμή την μέση ταχύτητα των λεωφορείων και να αποθηκεύεται στη βάση.Έτσι, χρησιμοποιώντας αυτή τη μέση ταχύτητα για κάθε μέρα της εβδομάδος και αναλόγως την ώρα της διαδρομής να ανακτάται η μέση ταχύτητα που είναι αποθηκευμένη στη βάση και να είναι ακριβέστερη η εκτιμώμενη ώρα άφιξης αλλά και διάρκεια της διαδρομής που θα προτείνεται τελικά στον χρήστη.Στην συγκεκριμένη εργασία, οι υπολογισμοί αυτοί γίνονται κατά μέσο όρο.

3.2 Απόδοση

Η λειτουργία της εφαρμογής στηρίζεται αποκλειστικά στην εκτέλεση ενός και μοναδικού νήματος(thread).Για την γρηγορότερη απόκριση και για καλύτερη απόδοση μπορεί να υποστηρίξει την πολυνημάτωση(multi-threading).Έτσι, θα αποφευχθούν φαινόμενα κρασαρίσματος της εφαρμογής και θα μειωθεί ο φόρτος εργασίας στο κινητό του χρήστη, που μπορεί να αξιοποιηθεί για multitasking καθώς και για πολυπλοκότερους υπολογισμούς και ανακτήσεις δεδομένων για τη βελτίωση της αξιοπιστίας, ταυτόχρονα με τη βελτίωση της απόδοσης.

4 Συμπέρασμα

Η εφαρμογή βρίσκεται ακόμα σε πρώιμο στάδιο.Στην παρούσα φάση δεν μπορεί να χρησιμοποιηθεί από τους πολίτες, καθώς υπάρχουν κυρίως ελλείψεις σε δεδομένα και προβλήματα απόδοσης που οδηγούν σε μεγάλο χρόνο απόκρισης.Τα αναφερόμενα προβλήματα, όμως, επιδέχονται βελτίωσης.Το σύστημα διαχείρισης δεδομένων που χρησιμοποιήθηκε είναι απλό, ανακτά, επεξεργάζεται και αποθηκεύει δεδομένα με γρήγορη ταχύτητα, και λόγω της φύσης του(NoSQL) μπορεί να είναι κατανεμημένο βελτιώνοντας ακόμα περισσότερο την αποτελεσματικότητα,την απόδοση, την ευελεξία και την ικανότητα της εφαρμογής να διαχειρίζεται τα Big Data.Ο βασικός κορμός έχει φτιαχτεί, οι λειτουργίες που προσφέρει θα βοηθήσουν σημαντικά τους χρήστες στην καθημερινότητα τους, διευκολύνωντας τις μετακινήσεις τους και προσφέρωντας τους μια δυνατότητα που ποτέ πριν δεν είχαν.

Αναφορές

1. http://www.astikovolou.gr/
2. https://github.com/tporadowski/redis/releases
3. https://redis.io/clients
4. https://lettuce.io/core/release/api/
5. https://github.com/lettuce-io/lettuce-core/releases
6. https://lettuce.io/docs/getting-started.html
7. https://lettuce.io/core/release/api/
8. https://stackoverflow.com/

9. GPS Based Bus Tracking System Leeza Singla, Dr. Parteek Bhatia
10. The Prediction of Bus Arrival Time Using Global Positioning System Data And Dynamic Traffic Information Tongyu Zhu, Fajin Ma, Tao Ma, Congcong Li State Key Laboratory of Software Development Environment