

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ,  
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ**

Διπλωματική Εργασία

“Αστικό Λεωφορείο Βόλου” σε Android περιβάλλον

Android development of application 'Volos Bus'

Ελεμένογλου Κωνσταντίνος

Επιβλέποντες Καθηγητές:  
Αλκιβιάδης Ακρίτας, Καθηγητής  
Γεώργιος Σταμούλης, Καθηγητής

Βόλος, Φεβρουάριος 2015



## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τους γονείς μου και την γιαγιά μου που βρίσκονταν πάντα δίπλα μου σε ότι χρειαζόμουν. Επίσης ευχαριστώ τους επιβλέποντες καθηγητές μου για όλη την βοήθεια και τις συμβουλές που μου παρείχαν. Τέλος ευχαριστώ τους φίλους και συγγενείς που μου στάθηκαν όποτε τους χρειάστηκα. Με την βοήθεια όλων σας, αυτή η διαδρομή ολοκληρώθηκε.

Κωνσταντίνος,  
Βόλος, 2015

# Περιεχόμενα

Περίληψη.....	1
1 Περιγραφή	
1.1 Περιγραφή της Εφαρμογής.....	2
1.2 Ροή του Προγράμματος.....	3
2 Εργαλεία	
2.1 Android Studio.....	10
2.2 Genymotion.....	12
2.3 Android Software Development Kit (SDK).....	13
3 Κώδικας της Εφαρμογής	
3.1 Υποκλάσεις της Activity.....	14
3.2 Parsing δεδομένων απο το αρχείο Static Data.....	29
3.3 Κλάση SimpleTime.....	31
3.4 Κλάσεις BusLine και BusStop.....	35
3.5 Κλάση ArriveTimesArrayAdapter.....	40
3.6 Strings.....	43
4 Πιθανές Επεκτάσεις	
4.1 Πιθανές Επεκτάσεις(Updates) για το “Αστικό Λεωφορείο Βόλου” .....	44
Βιβλιογραφία.....	46

# Περίληψη

Η ραγδαία εξέλιξη της τεχνολογίας των τηλεπικοινωνιών και πιο συγκεκριμένα των κινητών τηλεφώνων έχει οδηγήσει σε μαζική χρήση των κινητών τελευταίας γενιάς. Με πάρα πολύ μεγάλη χρησιμότητα σε διάφορους τομείς, τα κινητά μέσω των εφαρμογών τους έχουν μετατραπεί σε βασικό εργαλείο της καθημερινότητας μας.

Μία από τις ανάγκες που έχει σήμερα ένας κάτοικος ή ταξιδιώτης στον Βόλο, είναι η χρήση των αστικών συγκοινωνιών. Να μπορεί να γνωρίζει άμεσα, ανά πάσα στιγμή και χωρίς σύνδεση στο internet τα δρομολόγια όλων των γραμμών και τις ώρες αναχωρήσεων/αφίξεων αυτών. Αυτό το κενό φιλοδοξεί να καλύψει η εφαρμογή “Αστικό Λεωφορείο Βόλου”.

# 1 Περιγραφή

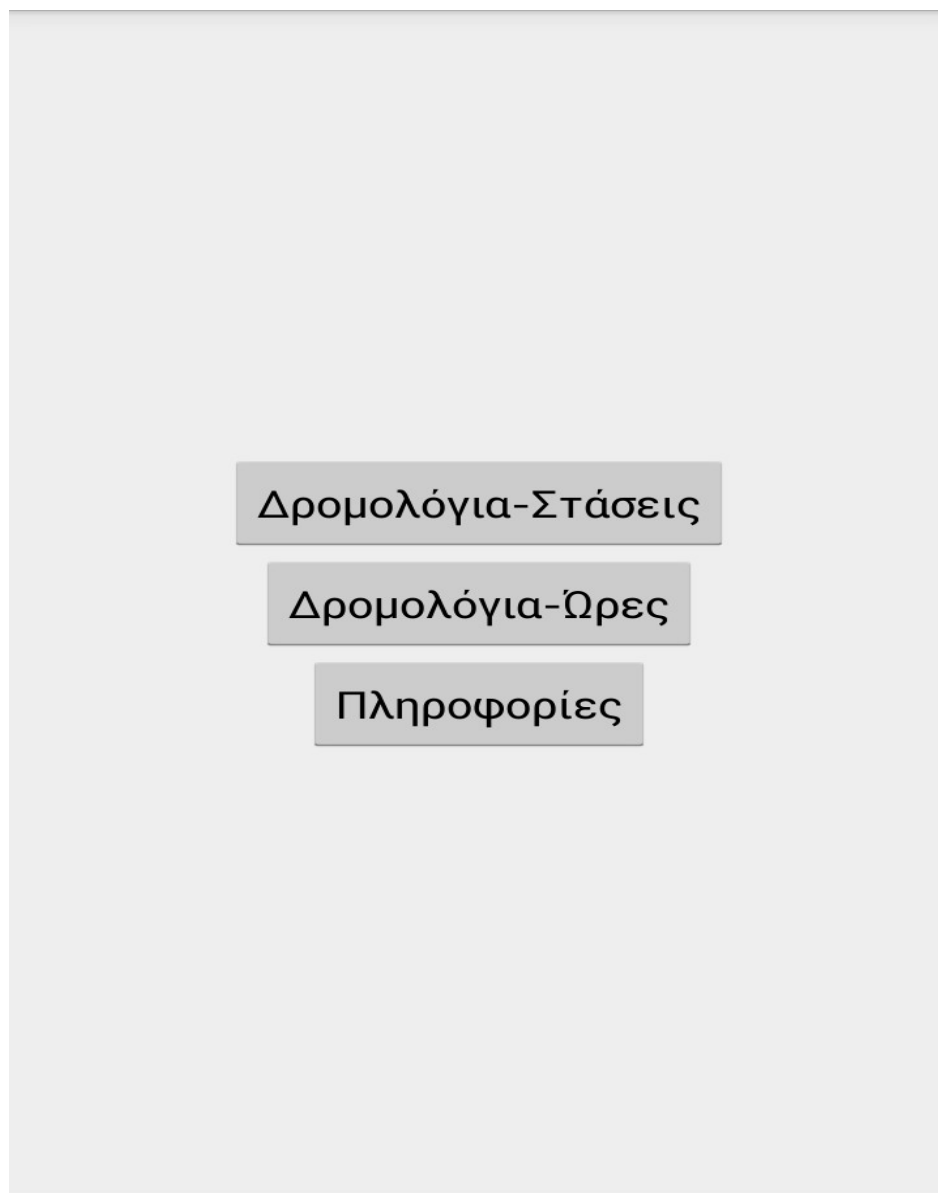
## 1.1 Περιγραφή της Εφαρμογής

Η εφαρμογή “Αστικό Λεωφορείο Βόλου” έχει σκοπό να παρέχει όσες περισσότερες πληροφορίες γύρω από τις αστικές συγκοινωνίες στην περιοχή του Βόλου. Οι βασικές δυνατότητες που έχει είναι:

- 1.Πληροφορίες δρομολογίων
- 2.Πληροφορίες γραμμής
- 3.Πληροφορίες στάσης
- 4.Επόμενο δρομολόγιο
- 5.Ώρες αφίξεων/αναχωρήσεων

## 1.2 Ροή του Προγράμματος

Στην αρχική σελίδα της εφαρμογής ο χρήστης καλείται να επιλέξει ανάμεσα σε 3 επιλογές. Το “Δρομολόγια-Στάσεις” για να μάθει περισσότερα γύρω απο τις γραμμές και τις στάσεις, το “Δρομολόγια-Ώρες” για να μάθει τις ώρες αφίξεων των γραμμών, ενώ με την επιλογή “Πληροφορίες” μπορεί να πληροφορηθεί σχετικά με την εφαρμογή.



Με την επιλογή “Δρομολόγια-Στάσεις”, τότε ανακατευθύνεται στην νέα οθόνη όπου μπορεί να δει το σύνολο των γραμμών των λεωφορείων καθώς και την διαδρομή που κάνουν(Αφετηρία-Τέρμα).

No1

Άναυρος - Ν. Ιωνία

---

No1

Ν. Ιωνία - Άναυρος

---

No2

Αφετηρία - Αμπελόκηποι

---

No2

Αμπελόκηποι - Αφετηρία

---

No3

Άναυρος - Ν. Ιωνία

---

No3

Ν. Ιωνία - Άναυρος

---

No4

Αφετηρία - Αηδονοφωλιές

---

No4

Αηδονοφωλιές - Αφετηρία

---

No4

Αφετηρία - Κατηχώρι/Σταγιάτες

---

No4

Κατηχώρι/Σταγιάτες - Αφετηρία

---

No5

Αφετηρία - Τέρμα Λεχώνια/  
Πλατανίδια

---



Εφόσον επιλέξει μια γραμμή, τότε μπορεί να δει τους δρόμους απο τους οποίους περνάει η συγκεκριμένη γραμμή, καθώς και το σύνολο των στάσεων που κάνει.

← No1 Άναυρος - Ν. Ιωνία

Οδοί

ΠΟΛΥΜΕΡΗ - ΔΗΜΗΤΡΙΑΔΟΣ -  
2ας ΝΟΕΜΒΡΙΟΥ - ΕΙΡΗΝΗΣ -  
ΕΘΝ. ΑΓΩΝΩΝ

Στάσεις

Άναυρος

---

Νοσοκομείο

---

Στρατηγού Σαράφη

---

Φιλίππου Ιωάννου

---

Βλαχάβα

---

Κασσαβέτη

---

Επιλέγοντας συγκεκριμένη στάση μπορεί να δει και ποιες άλλες γραμμές περνάνε από την στάση που διάλεξε και σε πόση ώρα.

## ← Στάση Νοσοκομείο

### Γραμμές που περνάν από τη στάση Νοσοκομείο (επόμενο λεωφορείο)

No1

Άναυρος - Ν. Ιωνία

Καθημερινές: 18:18

Σάββατα: 18:25

Κυριακές: 18:30

---

No1

Ν. Ιωνία - Άναυρος

Καθημερινές: 18:53

Σάββατα: 18:47

Κυριακές: 18:51

---

No5

Αφετηρία - Τέρμα Λεχώνια/  
Πλατανίδια

Καθημερινές: 18:40

Σάββατα: 18:30

Κυριακές: 18:30

---

No5

Τέρμα Λεχώνια/Πλατανίδια -

Εάν η επιλογή του στην αρχική οθόνη ήταν “Δρομολόγια-Ωρες”, τότε θα ανακατευθυνόταν και πάλι σε μία νέα οθόνη όπου θα μπορούσε να δει όλες τις γραμμές και τις διαδρομές τους(Αφετηρία-Τέρμα).

No1

Άναυρος - Ν. Ιωνία

---

No1

Ν. Ιωνία - Άναυρος

---

No2

Αφετηρία - Αμπελόκηποι

---

No2

Αμπελόκηποι - Αφετηρία

---

No3

Άναυρος - Ν. Ιωνία

---

No3

Ν. Ιωνία - Άναυρος

---

No4

Αφετηρία - Αηδονοφωλιές

---

No4

Αηδονοφωλιές - Αφετηρία

---

No4

Αφετηρία - Κατηχώρι/Σταγιάτες

---

No4

Κατηχώρι/Σταγιάτες - Αφετηρία

---

No5

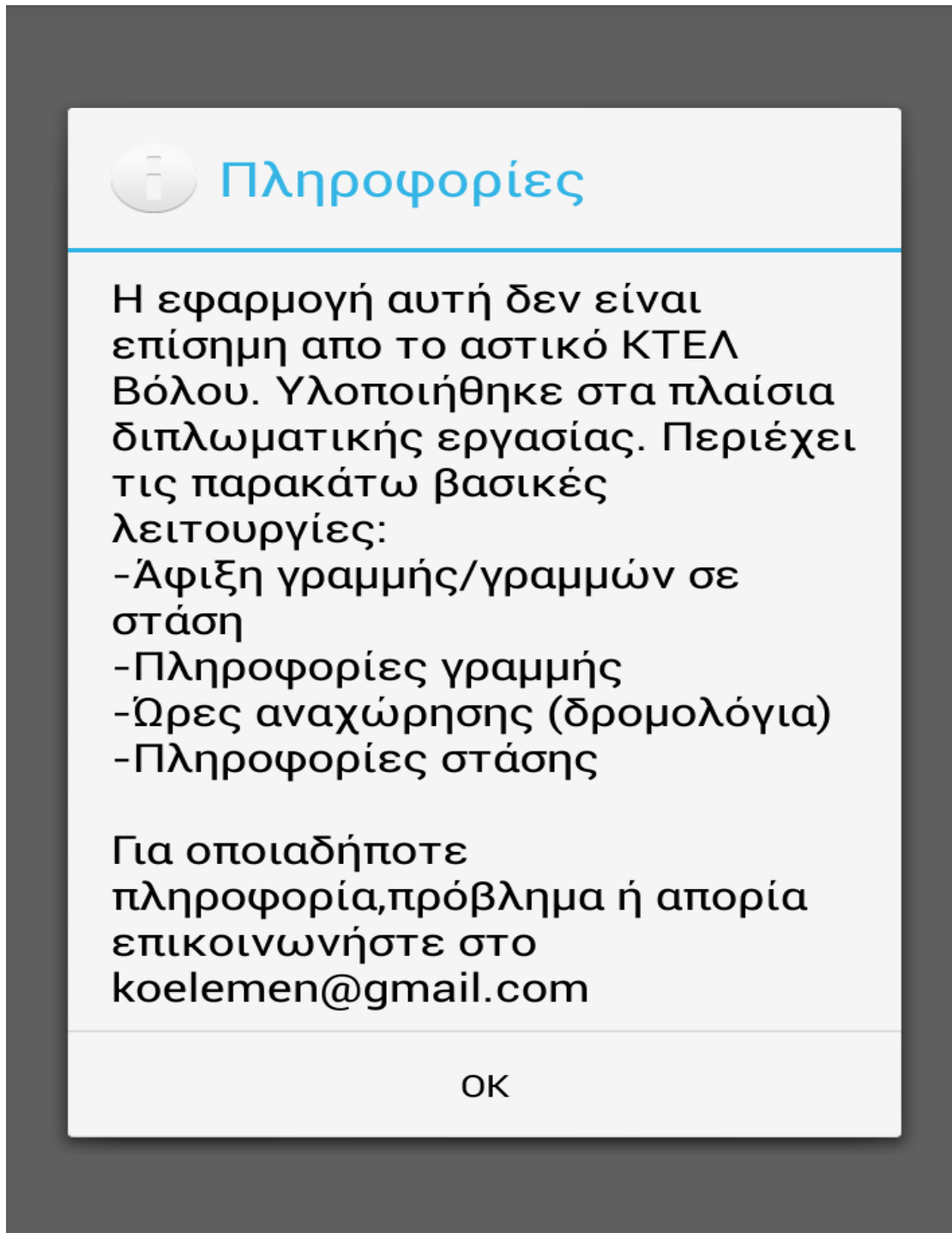
Αφετηρία - Τέρμα Λεχώνια/  
Πλατανίδια

---

Έχοντας ο χρήστης επιλέξει την γραμμή που επιθυμεί, μπορεί να δει ολόκληρο το εβδομαδιαίο πρόγραμμα με όλες τις ώρες αναχώρησης.

← Ώρες αναχώρησης Νο2 Αφετη...		
Καθημερινές	Σάββατα	Κυριακές
06:30	07:00	07:20
06:45	07:25	07:50
07:00	07:45	08:20
07:12	08:00	08:45
07:24	08:15	09:10
07:36	08:30	09:35
07:48	08:45	10:00
07:57	09:00	10:25
08:06	09:15	10:50
08:18	09:30	11:15

Τέλος, εάν επιλέξει να πατήσει “Πληροφορίες” από την αρχική οθόνη, τότε εμφανίζεται το παρακάτω μήνυμα με πληροφορίες σχετικά με την εφαρμογή και email επικοινωνίας.



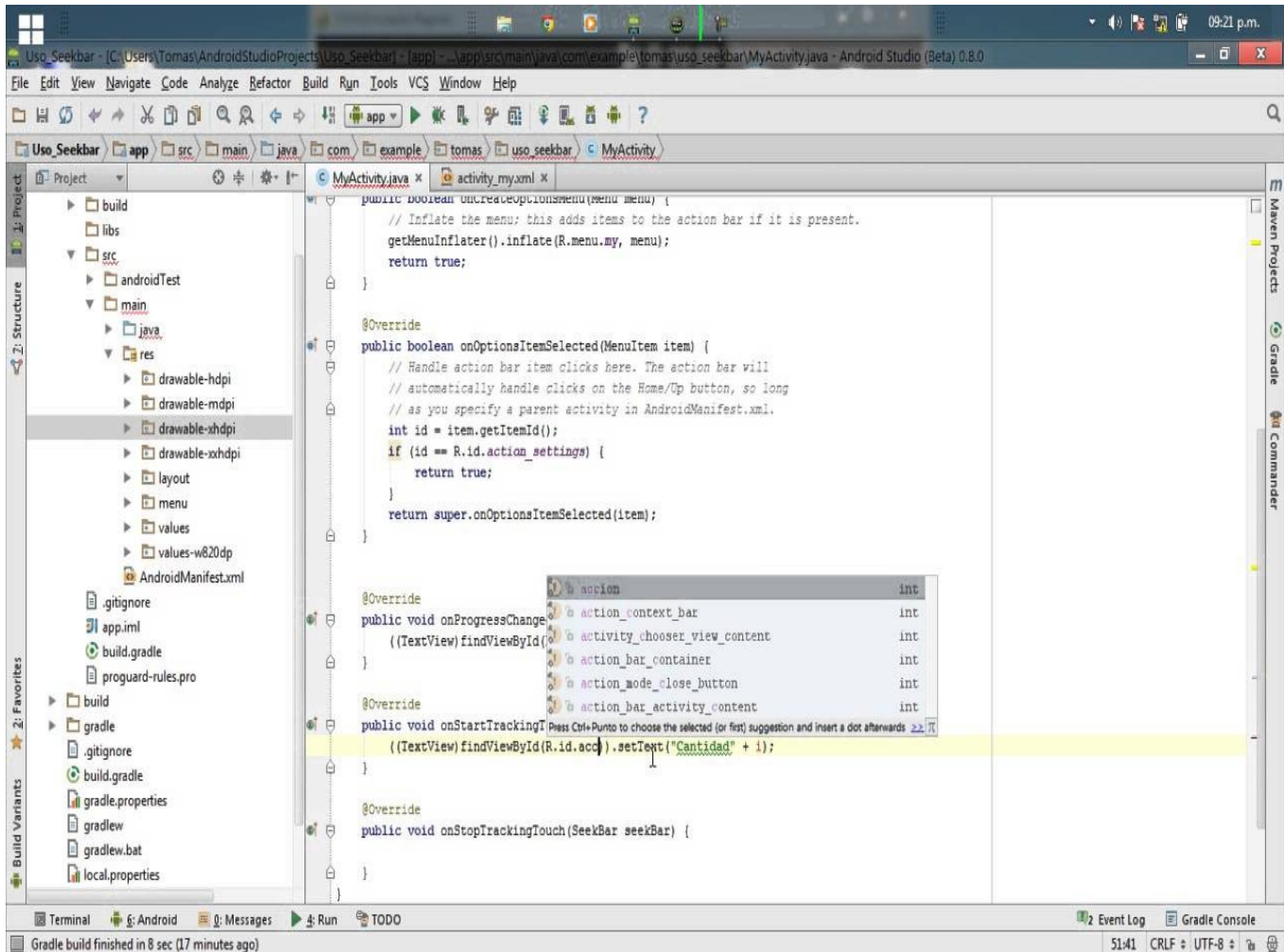
## 2 Εργαλεία

### 2.1 Android Studio

Το Android Studio είναι το επίσημο IDE για την ανάπτυξη εφαρμογών Android, βασισμένο στο IntelliJ IDEA. Εκτός από τις δυνατότητες του IntelliJ, το Android Studio προσφέρει:

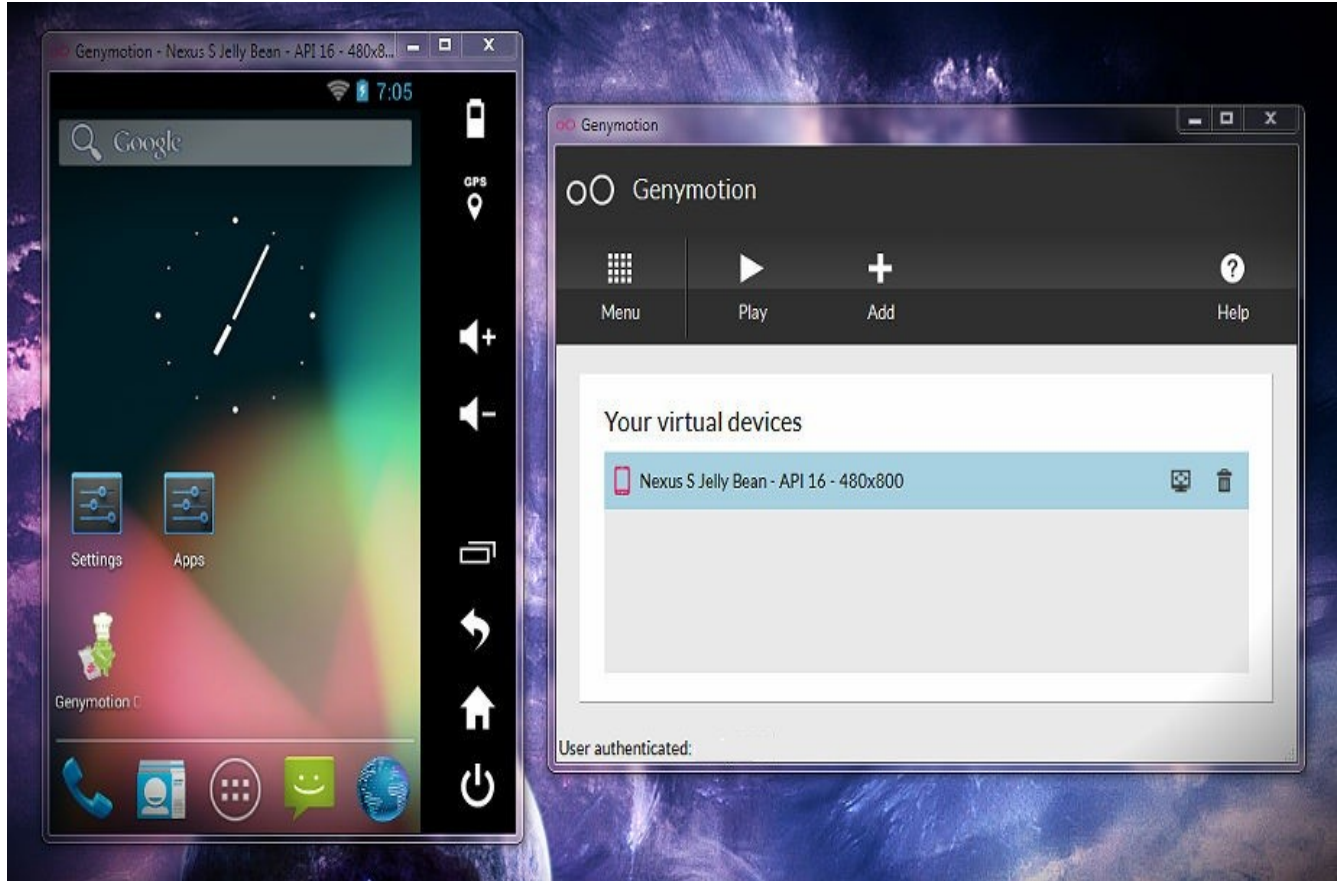
- Ευέλικτο σύστημα compiling/building (Gradle)
- Κατασκευή πολλαπλών εκδόσεων αρχείων apk
- Πρότυπα κώδικα για να βοηθήσει να οικοδομηθούν κοινά χαρακτηριστικά
- Πλούσιο επεξεργαστή διάταξης με υποστήριξη με drag and drop
- Εργαλεία lln για την παρακολούθηση απόδοσης, τη χρηστικότητα, τη συμβατότητας έκδοσης, και άλλων προβλημάτων

- Σύστημα app-signing (Proguard)
- Η ενσωματωμένη υποστήριξη για την cloud πλατφόρμα της Google, καθιστώντας εύκολη την ενσωμάτωση του Google Cloud Messaging και App Engine



## 2.2 Genymotion

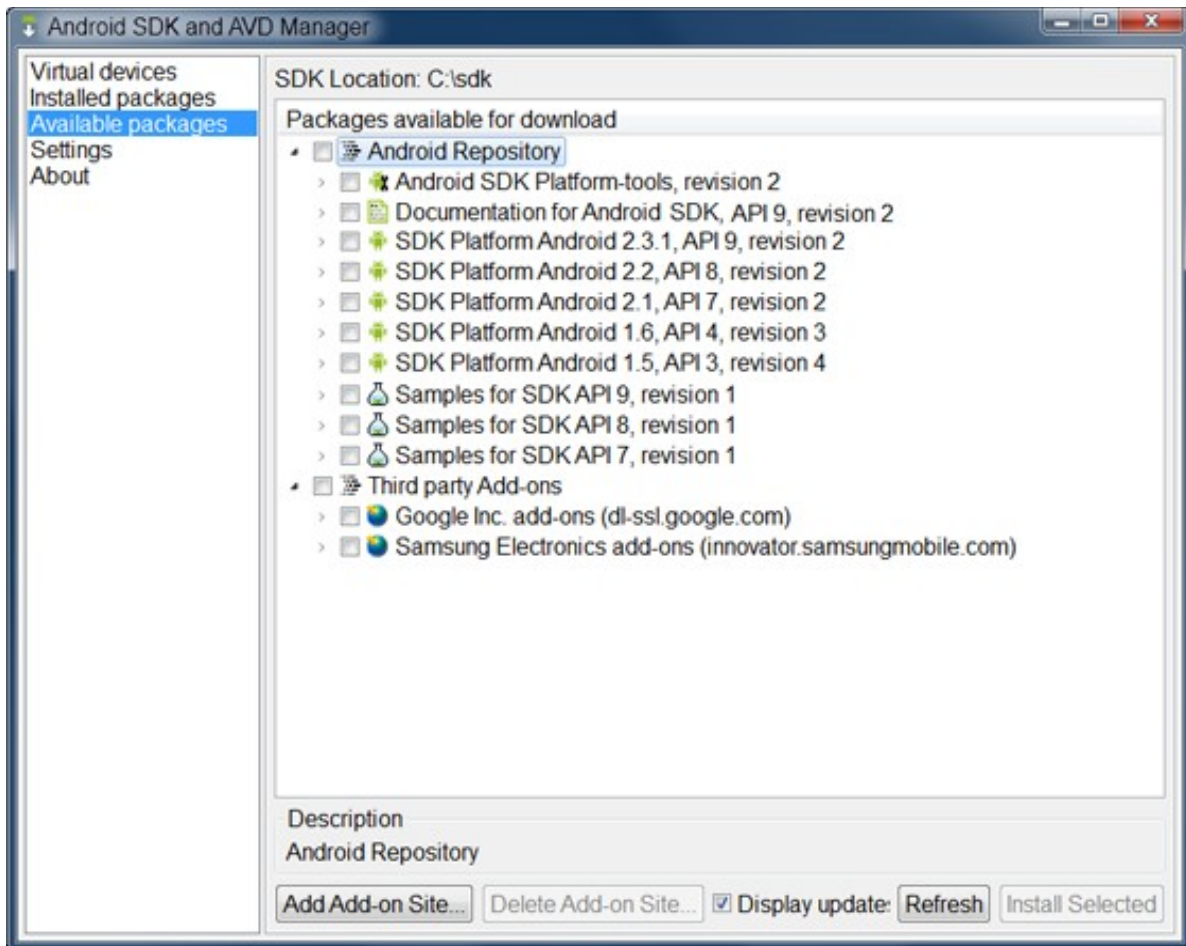
Το Genymotion είναι ένας εξομοιωτής του Android το οποίο περιλαμβάνει ένα πλήρες σύνολο των αισθητήρων και τα χαρακτηριστικά για να αλληλεπιδράσει με το εικονικό περιβάλλον Android. Με το Genymotion, δύναται να δοκιμαστούν οι Android εφαρμογές σε ένα ευρύ φάσμα των εικονικών συσκευών για αναπτυξιακούς σκοπούς, δοκιμή και επίδειξη. Το Genymotion είναι γρήγορη, απλό στην εγκατάσταση και ισχυρό χάρη στα φιλικά προς το χρήστη γραφικά στοιχεία αισθητήρα και δυνατότητες αλληλεπίδρασης. Είναι διαθέσιμο για Windows, Mac OS X και Linux λειτουργικά συστήματα.





## 2.3 Android Software Development Kit (SDK)

Το Android Software Development Kit (SDK) περιέχει όλα τα εργαλεία χαμηλού επιπέδου για να “χτιστεί”, να “τρέξει” αλλά και να “τεσταριστεί” κάθε Android εφαρμογή.



## 3 Κώδικας της Εφαρμογής

### 3.1 Υποκλάσεις της Activity

Η φιλοσοφία που ακολουθήθηκε είναι πως σε κάθε "οθόνη" που βλέπει ο χρήστης αντιστοιχεί μία υποκλάση της κλάσης Activity καθώς και ένα αρχείο xml το οποίο περιγράφει τα γραφικά στοιχεία. Έτσι λοιπόν έχουμε υλοποιήσει 6 υποκλάσεις και αντίστοιχα 6 αρχεία xml με παρόμοιο όνομα (στον φάκελο layout).

1. MenuActivity - Εδώ υπάρχει η υλοποίηση της αρχικής οθόνης με τις τρεις βασικές επιλογές. Αξίζει να σημειωθεί ότι στη γραμμή 28 γίνεται χρήση της κλάσης AlertDialog για την εμφάνιση του modal pop-up window που εμφανίζεται με το πάτημα του κουμπιού "Πληροφορίες".

```

package edu.uth.gr.ktelvolou;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class MenuActivity extends Activity
{
    private Button btnBuslineBusstops;
    private Button btnBuslineHours;
    private Button btnAbout;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menu);

        btnBuslineBusstops = (Button)
findViewById(R.id.menu_btn_buslines_busstops);
        btnBuslineHours = (Button)
findViewById(R.id.menu_btn_buslines_hours);
        btnAbout = (Button) findViewById(R.id.menu_btn_about);

        final AlertDialog aboutDialog = new AlertDialog.Builder(this)
            .setCancelable(true)
            .setIcon(android.R.drawable.ic_dialog_info)
            .setTitle(R.string.menu_btn_about)
            .setMessage(R.string.app_info)
            .setPositiveButton(android.R.string.ok, new
DialogInterface.OnClickListener(){

                @Override
                public void onClick(DialogInterface dialog, int which)
                { /* just close the dialog */ }
            })
        .create();
    }
}

```

```

        btnBuslineBusstops.setOnClickListener(new OnClickListener(){

            @Override
            public void onClick(View v)
            {
                Intent intent = new Intent(MenuActivity.this,
BuslineListActivity.class);
                startActivity(intent);
            }
        });

        btnBuslineHours.setOnClickListener(new OnClickListener(){

            @Override
            public void onClick(View v)
            {
                Intent intent = new Intent(MenuActivity.this,
ArriveTimesListActivity.class);
                startActivity(intent);
            }
        });

        btnAbout.setOnClickListener(new OnClickListener(){

            @Override
            public void onClick(View v)
            {
                aboutDialog.show();
            }
        });
    }
}

```

2 & 3. BuslineListActivity & ArriveTimesListActivity - Ουσιαστικά εδώ γίνεται ο διαχωρισμός για το αν θα έχουμε twopane layout ή ξεχωριστή οθόνη για την εμφάνιση της λίστας ( των γραμμών ή των ωρών άφιξης αντίστοιχα)

```

package edu.uth.gr.ktelvolou;

import android.content.Intent;
import android.os.Bundle;
import android.support.v4.app.FragmentActivity;
import android.support.v4.app.NavUtils;
import android.view.MenuItem;

/**
 * An activity representing a list of Bus lines. This activity has different
 * presentations for handset and tablet-size devices. On handsets, the activity
 * presents a list of items, which when touched, lead to a
 * {@link BuslineDetailActivity} representing item details. On tablets, the
 * activity presents the list of items and item details side-by-side using two
 * vertical panes.
 * <p>
 * The activity makes heavy use of fragments. The list of items is a
 * {@link BuslineListFragment} and the item details (if present) is a
 * {@link BuslineDetailFragment}.
 * <p>
 * This activity also implements the required
 * {@link BuslineListFragment.Callbacks} interface to listen for item
 * selections.
 */
public class BuslineListActivity extends FragmentActivity implements
    BuslineListFragment.Callbacks
{
    /**
     * Whether or not the activity is in two-pane mode, i.e. running on a tablet
     * device.
     */
    private boolean mTwoPane;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
    }

```

```

setContentView(R.layout.activity_busline_list);

if (findViewById(R.id.busline_detail_container) != null)
{
    // The detail container view will be present only in the
    // large-screen layouts (res/values-large and
    // res/values-sw600dp). If this view is present, then the
    // activity should be in two-pane mode.
    mTwoPane = true;

    // In two-pane mode, list items should be given the
    // 'activated' state when touched.
    ((BuslineListFragment)
getSupportFragmentManager().findFragmentById(R.id.busline_list)).setActivateOnItem
Click(true);
}

// TODO: If exposing deep links into your app, handle intents here.
}

/**
 * Callback method from {@link BuslineListFragment.Callbacks} indicating
 * that the item with the given ID was selected.
 */
@Override
public void onItemSelected(String id)
{
    if (mTwoPane)
    {
        // In two-pane mode, show the detail view in this activity by
        // adding or replacing the detail fragment using a
        // fragment transaction.
        Bundle arguments = new Bundle();
        arguments.putString(BuslineDetailFragment.ARG_ITEM_ID, id);
        BuslineDetailFragment fragment = new BuslineDetailFragment();
        fragment.setArguments(arguments);
        getSupportFragmentManager().beginTransaction()
ail_container, fragment)
                                .replace(R.id.busline_det
                                .commit());
    }
}

```

```

        else
        {
            // In single-pane mode, simply start the detail activity
            // for the selected item ID.
            Intent detailIntent = new Intent(this, BuslineDetailActivity.class);
            detailIntent.putExtra(BuslineDetailFragment.ARG_ITEM_ID, id);
            startActivity(detailIntent);
        }
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item)
    {
        int id = item.getItemId();
        if (id == android.R.id.home)
        {
            // This ID represents the Home or Up button. In the case of this
            // activity, the Up button is shown. Use NavUtils to allow users
            // to navigate up one level in the application structure. For
            // more details, see the Navigation pattern on Android Design:
            //
            // http://developer.android.com/design/patterns/navigation.html#up-
            vs-back
            //
            NavUtils.navigateUpTo(this, new Intent(this, MenuActivity.class));
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}

```

```

package edu.uth.gr.ktelvolou;

import android.content.Intent;
import android.os.Bundle;
import android.support.v4.app.FragmentActivity;
import android.support.v4.app.NavUtils;
import android.view.MenuItem;

/**

```

```

* An activity representing a list of Items. This activity has different
* presentations for handset and tablet-size devices. On handsets, the activity
* presents a list of items, which when touched, lead to a
* {@link ArriveTimesDetailActivity} representing item details. On tablets, the
* activity presents the list of items and item details side-by-side using two
* vertical panes.
* <p>
* The activity makes heavy use of fragments. The list of items is a
* {@link ArriveTimesListFragment} and the item details (if present) is a
* {@link ArriveTimesDetailFragment}.
* <p>
* This activity also implements the required {@link
ArriveTimesListFragment.Callbacks}
* interface to listen for item selections.
*/

```

```

public class ArriveTimesListActivity extends FragmentActivity implements

    ArriveTimesListFragment.Callbacks
{

    /**
     * Whether or not the activity is in two-pane mode, i.e. running on a tablet
     * device.
     */
    private boolean mTwoPane;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_arrive_times_list);

        if (findViewById(R.id.item_detail_container) != null)
        {
            // The detail container view will be present only in the
            // large-screen layouts (res/values-large and
            // res/values-sw600dp). If this view is present, then the
            // activity should be in two-pane mode.
            mTwoPane = true;

            // In two-pane mode, list items should be given the
            // 'activated' state when touched.

```



```

        ((ArriveTimesListFragment)
getSupportFragmentManager().findFragmentById(R.id.item_list)).setActivateOnItemClick(true);
    }
}

```

```

@Override
public boolean onOptionsItemSelected(MenuItem item)
{

```

```

    int id = item.getItemId();
    if (id == android.R.id.home)
    {

```

```

        // This ID represents the Home or Up button. In the case of this
        // activity, the Up button is shown. Use NavUtils to allow users
        // to navigate up one level in the application structure. For
        // more details, see the Navigation pattern on Android Design:
        //
        // http://developer.android.com/design/patterns/navigation.html#up-

```

vs-back

```

        //
        NavUtils.navigateUpFromSameTask(this);
        return true;
    }

```

```

    return super.onOptionsItemSelected(item);
}

```

```

/**

```

that

```

    * Callback method from {@link ArriveTimesListFragment.Callbacks} indicating

```

```

    * the item with the given ID was selected.

```

```

    */

```

```

@Override
public void onItemClick(String id)
{

```

```

    if (mTwoPane)
    {

```

```

        // In two-pane mode, show the detail view in this activity by
        // adding or replacing the detail fragment using a
        // fragment transaction.

```

```

        Bundle arguments = new Bundle();
        arguments.putString(ArriveTimesDetailFragment.ARG_ITEM_ID,

```

id);

```

        ArriveTimesDetailFragment fragment = new
ArriveTimesDetailFragment();
        fragment.setArguments(arguments);
        getSupportFragmentManager().beginTransaction()
                                                    .replace(R.id.item_detail
_container, fragment)
                                                    .commit();
    }
    else
    {
        // In single-pane mode, simply start the detail activity
        // for the selected item ID.
        Intent detailIntent = new Intent(this,
ArriveTimesDetailActivity.class);
        detailIntent.putExtra(ArriveTimesDetailFragment.ARG_ITEM_ID,
id);
        startActivity(detailIntent);
    }
}
}

```

#### 4 & 5. BuslineDetailActivity &

ArriveTimesDetailActivity - Εδώ γίνεται η εμφάνιση των δεδομένων (για τις στάσεις της γραμμής ή τις ώρες άφιξης). Αυτές οι δύο κλασεις περνάνε τα δεδομένα προς απεικόνιση στην αντίστοιχη κλάση Fragment.

```

package edu.uth.gr.ktelvolou;

import edu.uth.gr.ktelvolou.data.StaticData;
import edu.uth.gr.ktelvolou.pojo.BusLine;
import android.content.Intent;

```

```

import android.os.Bundle;
import android.support.v4.app.NavUtils;
import android.support.v7.app.ActionBarActivity;
import android.view.MenuItem;

/**
 * An activity representing a single Bus line detail screen. This activity is
 * only used on handset devices. On tablet-size devices, item details are
 * presented side-by-side with a list of items in a {@link BuslineListActivity}.
 * <p>
 * This activity is mostly just a 'shell' activity containing nothing more than
 * a {@link BuslineDetailFragment}.
 */
public class BuslineDetailActivity extends ActionBarActivity
{
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_busline_detail);

        // Show the Up button in the action bar.
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        // savedInstanceState is non-null when there is fragment state
        // saved from previous configurations of this activity
        // (e.g. when rotating the screen from portrait to landscape).
        // In this case, the fragment will automatically be re-added
        // to its container so we don't need to manually add it.
        // For more information, see the Fragments API guide at:
        //
        // http://developer.android.com/guide/components/fragments.html
        //
        if (savedInstanceState == null)
        {
            BusLine busLine = StaticData.BUS_LINES.get(
                Integer.parseInt(getIntent().getStringExtra(
                    BuslineDetailFragment.ARG_ITEM_ID)));
            setTitle(busLine.toString());

            // Create the detail fragment and add it to the activity

```

```

        // using a fragment transaction.
        Bundle arguments = new Bundle();
        arguments.putString(BuslineDetailFragment.ARG_ITEM_ID,

getIntent().getStringExtra(BuslineDetailFragment.ARG_ITEM_ID));
        BuslineDetailFragment fragment = new BuslineDetailFragment();
        fragment.setArguments(arguments);
        getSupportFragmentManager().beginTransaction()
                                                    .add(R.id.busline_detail_
container, fragment)
                                                    .commit();
    }
}

@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    int id = item.getItemId();
    if (id == android.R.id.home)
    {
        // This ID represents the Home or Up button. In the case of this
        // activity, the Up button is shown. Use NavUtils to allow users
        // to navigate up one level in the application structure. For
        // more details, see the Navigation pattern on Android Design:
        //
        // http://developer.android.com/design/patterns/navigation.html#up-
vs-back
        //
        NavUtils.navigateUpTo(this, new Intent(this,
BuslineListActivity.class));
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}
}

```

```
package edu.uth.gr.ktelvolou;
```

```
import edu.uth.gr.ktelvolou.data.StaticData;
```

```

import edu.uth.gr.ktelvolou.pojo.BusLine;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.support.v4.app.NavUtils;
import android.view.MenuItem;

/**
 * An activity representing a single Item detail screen. This activity is only
 * used on handset devices. On tablet-size devices, item details are presented
 * side-by-side with a list of items in a {@link ArriveTimesListActivity}.
 * <p>
 * This activity is mostly just a 'shell' activity containing nothing more than
 * a {@link ArriveTimesDetailFragment}.
 */
public class ArriveTimesDetailActivity extends ActionBarActivity
{

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_arrive_times_detail);

        // Show the Up button in the action bar.
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        // savedInstanceState is non-null when there is fragment state
        // saved from previous configurations of this activity
        // (e.g. when rotating the screen from portrait to landscape).
        // In this case, the fragment will automatically be re-added
        // to its container so we don't need to manually add it.
        // For more information, see the Fragments API guide at:
        //
        // http://developer.android.com/guide/components/fragments.html
        //
        if (savedInstanceState == null)
        {
            BusLine busLine = StaticData.BUS_LINES.get(
                Integer.parseInt(getIntent().getStringExtra(
                    BuslineDetailFragment.ARG_ITEM_ID)));

```

```

        setTitle(getString(R.string.title_arrive_times_detail,
busLine.toString()));

        // Create the detail fragment and add it to the activity
        // using a fragment transaction.
        Bundle arguments = new Bundle();
        arguments.putString(ArriveTimesDetailFragment.ARG_ITEM_ID,

        getIntent().getStringExtra(ArriveTimesDetailFragment.ARG_ITEM_ID));
        ArriveTimesDetailFragment fragment = new
ArriveTimesDetailFragment();
        fragment.setArguments(arguments);
        getSupportFragmentManager().beginTransaction()
                                                    .add(R.id.item_detail_co
ntainer, fragment)
                                                    .commit();
    }
}

@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    int id = item.getItemId();
    if (id == android.R.id.home)
    {
        // This ID represents the Home or Up button. In the case of this
        // activity, the Up button is shown. Use NavUtils to allow users
        // to navigate up one level in the application structure. For
        // more details, see the Navigation pattern on Android Design:
        //
        // http://developer.android.com/design/patterns/navigation.html#up-
vs-back
        //
        NavUtils.navigateUpTo(this, new Intent(this,
ArriveTimesListActivity.class));
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}

```

6. BusstopActivity - Εδώ γίνεται η παρουσίαση των γραμμών που περνάνε από συγκεκριμένη στάση. Αξίζει να σημειωθεί η χρήση της κλάσης ArriveTimesArrayAdapter στη γραμμή 58.

```
package edu.uth.gr.ktelvolou;

import java.util.ArrayList;
import java.util.List;
import android.content.Intent;
import android.os.Bundle;
import android.support.v4.app.NavUtils;
import android.support.v7.app.ActionBarActivity;
import android.view.MenuItem;
import android.widget.ListView;
import android.widget.TextView;
import edu.uth.gr.ktelvolou.data.StaticData;
import edu.uth.gr.ktelvolou.misc.ArriveTimesArrayAdapter;
import edu.uth.gr.ktelvolou.pojo.BusLine;
import edu.uth.gr.ktelvolou.pojo.BusStop;

public class BusstopActivity extends ActionBarActivity
{
    public static final String ARG_BUSSTOP = "busstop";

    private TextView txvTitle;
    private ListView lsvStops;

    private BusStop mItem;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_busstop);
    }
}
```

```

    if (getIntent().hasExtra(ARG_BUSSTOP))
    {
        // Load the content specified by the arguments.
        mItem = (BusStop)
getIntent().getSerializableExtra(ARG_BUSSTOP);
    }
    if (mItem == null)
    {
        mItem = new BusStop(ARG_BUSSTOP);
    }

    setTitle(getString(R.string.title_activity_busstop, mItem.toString()));

    List<BusLine> busLines = new ArrayList<>();

    for (BusLine busLine : StaticData.BUS_LINES)
    {
        if (busLine.getBusStops().contains(mItem))
        {
            busLines.add(busLine);
        }
    }

    txvTitle = (TextView) findViewById(R.id.busstop_txv_title);
    lsvStops = (ListView) findViewById(R.id.busstop_lsv_buslines);

    txvTitle.setText(getString(R.string.busstop_txv_title, mItem.toString()));
    lsvStops.setAdapter(new ArriveTimesArrayAdapter(this,

android.R.layout.simple_list_item_activated_1,
                                android.R.id.text1,
                                busLines,
                                mItem));
}

@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

```



```

        if (id == android.R.id.home)
        {
            NavUtils.navigateUpTo(this, new Intent(this,
BuslineListActivity.class));
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}

```

## 3.2 Parsing δεδομένων απο το αρχείο Static Data

Επειδή τα δεδομένα στο site του Αστικού ΚΤΕΛ Βόλου δεν υπήρχαν σε κάποια parseable μορφή και επειδή δεν προσφέρεται ούτε κάποιο Webservice, όλη η πληροφορία γίνεται parse αφότου αποθηκεύτηκε σε βοηθητικές κλασεις/δομές. Αυτή η διαδικασία γίνεται στο αρχείο StaticData

```

static
{
    BusLine busLine;
    BusStop busStop;

    Scanner scanner = new Scanner(DATA);

    while (scanner.hasNextLine())
    {
        busLine = new BusLine();
        busLine.setLineNumber(scanner.nextInt());
        String str = scanner.nextLine();
        str = scanner.nextLine();

        String[] s = str.split(" - ");

        busLine.setStartLocation(s[0]);
    }
}

```

```

busLine.setEndLocation(s[1]);

str = scanner.nextLine();

if (str.length() > 1)
{
    s = str.split(" - ");
    for (String street : s)
    {
        STREETS.add(street);
        busLine.addStreet(street);
    }
    do
    {
        str = scanner.nextLine();

        if (str.length() <= 1)
        {
            break;
        }

        busStop = new BusStop(str);

        BUS_STOPS.add(busStop);
        busLine.addBusStop(busStop);
    } while (scanner.hasNext());
}

BUS_LINES.add(busLine);
BUS_LINES_MAP.put(busLine.id(), busLine);
}

scanner.close();

scanner = new Scanner(TIMES_DATA);

while (scanner.hasNext())
{
    busLine = new BusLine();

    String[] s = scanner.nextLine().split(" - ");
    busLine.setLineNumber(Integer.parseInt(s[0]));
}

```

```

busLine.setStartLocation(s[1]);
busLine.setEndLocation(s[2]);

int arriveType = Integer.parseInt(scanner.nextLine());
Log.d("dbg", busLine.id());
busLine = BUS_LINES_MAP.get(busLine.id());

String str;

while (scanner.hasNextLine())
{
    str = scanner.nextLine();
    if (str.length() <= 1)
    {
        break;
    }

    s = str.split(" ");

    int hour = Integer.parseInt(s[0]);

    for (int i = 1; i < s.length; i++)
    {
        busLine.addArriveTimes(arriveType, new
SimpleTime(hour, Integer.parseInt(s[i])));
    }
}

scanner.close();
}
}

```

### 3.3 Κλάση SimpleTime

Για την εύκολη και γρήγορη απεικόνιση των χρόνων

άφιξης/αναχώρησης δημιουργήθηκε η κλάση SimpleTime. Αξίζει να σημειωθεί η υλοποίηση του interface Comparable ώστε να μπορούμε εύκολα να ταξινομήσουμε τους χρόνους και να βρίσκουμε το αμέσως επόμενο λεωφορείο που περνά από τη συγκεκριμένη στάση.

```
package edu.uth.gr.ktelvolou.misc;
```

```
public class SimpleTime implements Comparable<SimpleTime>
{
    private int hours;
    private int minutes;

    public SimpleTime(int hours, int minutes)
    {
        if (hours < 0 || hours > 23
            || minutes < 0 || minutes > 59)
        {
            throw new IllegalArgumentException("Hours must be in range 0 to
23 and minutes 0 to 59.");
        }

        this.hours = hours;
        this.minutes = minutes;
    }

    public SimpleTime()
    {
        this(0, 0);
    }

    public int getHours()
    {
        return hours;
    }

    public int getMinutes()
```

```

    {
        return minutes;
    }

    public void addMinutes(int minutes)
    {
        if (minutes < 0 || minutes > 59)
        {
            throw new IllegalArgumentException("Minutes must be in range 0 to
59.");
        }

        this.minutes += minutes;

        if (this.minutes > 59)
        {
            ++hours;

            hours %= 24;
            this.minutes -= 60;
        }
    }

    @Override
    public String toString()
    {
        StringBuilder builder = new StringBuilder();

        builder.append(String.format("%02d", hours))
            .append(":")
            .append(String.format("%02d", minutes));

        return builder.toString();
    }

    @Override
    public int hashCode()
    {
        final int prime = 31;
        int result = 1;
        result = prime * result + hours;
    }

```

```
        result = prime * result + minutes;
        return result;
    }
```

```
@Override
public boolean equals(Object obj)
{
    if (this == obj)
    {
        return true;
    }
    if (obj == null)
    {
        return false;
    }
    if (!(obj instanceof SimpleTime))
    {
        return false;
    }
    SimpleTime other = (SimpleTime) obj;
    if (hours != other.hours)
    {
        return false;
    }
    if (minutes != other.minutes)
    {
        return false;
    }
    return true;
}
```

```
@Override
public int compareTo(SimpleTime another)
{
    if (this.hours != another.hours)
    {
        return (int) Math.signum(this.hours - another.hours);
    }
    if (this.minutes != another.minutes)
    {
        return (int) Math.signum(this.minutes - another.minutes);
    }
}
```

```
        return 0;
    }
}
```

### 3.4 Κλάσεις BusLine και BusStop

Για την μοντέλοποίηση των οντοτήτων γραμμής και στάσης δημιουργήθηκαν οι κλάσεις BusLine και BusStop. Η κλάση busline έχει από οδούς, στάσεις και ώρες άφιξης για την προσομοίωση του "ένα-προς-πολλά". Το άξιο προσοχής στην κλάση BusStop είναι η υλοποίηση του Serializable ώστε να περνάμε εύκολα τις μεταβλητές ανάμεσα από δύο Activity κλάσεις.

```
package edu.uth.gr.ktelvolou.pojo;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import edu.uth.gr.ktelvolou.misc.SimpleTime;

public class BusLine
{
    private int lineNumber;
    private String startLocation;
    private String endLocation;
    private List<String> streets;
    private List<BusStop> stops;
    private Map<Integer, List<SimpleTime>> arriveTimes;

    public BusLine()
    {
        streets = new ArrayList<>();
        stops = new ArrayList<>();
    }
}
```

```

        arriveTimes = new HashMap<>();
    }

    public String id()
    {
        return lineNumber + startLocation + endLocation;
    }

    public String streets()
    {
        if (streets.isEmpty())
        {
            return "";
        }

        String delim = " - ";
        StringBuilder builder = new StringBuilder();

        for (String s : streets)
        {
            builder.append(s);
            builder.append(delim);
        }

        builder.delete(builder.length() - delim.length(), builder.length() - 1);

        return builder.toString();
    }

    public int getLineNumber()
    {
        return lineNumber;
    }

    public String getStartLocation()
    {
        return startLocation;
    }

    public String getEndLocation()
    {
        return endLocation;
    }

```



```

}

public List<String> getStreets()
{
    return streets;
}

public List<BusStop> getBusStops()
{
    return stops;
}

public void addStreet(String street)
{
    streets.add(street);
}

public void addBusStop(BusStop busStop)
{
    stops.add(busStop);
}

public Map<Integer, List<SimpleTime>> getArriveTimes()
{
    return arriveTimes;
}

public void addArriveTimes(int type, SimpleTime time)
{
    if (!arriveTimes.containsKey(type))
    {
        arriveTimes.put(type, new ArrayList<SimpleTime>());
    }

    arriveTimes.get(type).add(time);
}

public void setLineNumber(int lineNumber)
{
    this.lineNumber = lineNumber;
}

```

```

public void setStartLocation(String startLocation)
{
    this.startLocation = startLocation;
}

public void setEndLocation(String endLocation)
{
    this.endLocation = endLocation;
}

@Override
public String toString()
{
    StringBuilder builder = new StringBuilder();
    builder.append("No");
    builder.append(lineNumber);
    builder.append("\n");
    builder.append(startLocation);
    builder.append(" - ");
    builder.append(endLocation);
    return builder.toString();
}
}

```

```
package edu.uth.gr.ktelvolou.pojo;
```

```
import java.io.Serializable;
```

```
public class BusStop implements Serializable
{
```

```
    private static final long serialVersionUID = -6099107724487724003L;
```

```
    private String name;
```

```
    public BusStop(String name)
```

```
    {
```

```
        super();
```

```
        this.name = name;
```

```
    }
```

```

@Override
public String toString()
{
    StringBuilder builder = new StringBuilder();

    builder.append(name);

    return builder.toString();
}

```

```

@Override
public int hashCode()
{
    final int prime = 31;
    int result = 1;
    result = prime * result + ((name == null) ? 0 : name.hashCode());
    return result;
}

```

```

@Override
public boolean equals(Object obj)
{
    if (this == obj)
    {
        return true;
    }
    if (obj == null)
    {
        return false;
    }
    if (!(obj instanceof BusStop))
    {
        return false;
    }
    BusStop other = (BusStop) obj;
    if (name == null)
    {
        if (other.name != null)
        {
            return false;
        }
    }
}

```

```

    }
    else if (!name.equals(other.name))
    {
        return false;
    }
    return true;
}
}

```

### 3.5 Κλάση ArriveTimesArrayAdapter

Επίσης υπάρχει η κλάση ArriveTimesArrayAdapter για την εύκολη απεικόνιση των ωρών άφιξης του επόμενου λεωφορείου σε συγκεκριμένη στάση. Όλη η αναζήτηση/εύρεση της κατάλληλης ώρας γίνεται στις γραμμές 67-80

```

package edu.uth.gr.ktelvolou.misc;

import java.util.Calendar;
import java.util.List;
import java.util.TreeSet;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;
import edu.uth.gr.ktelvolou.R;
import edu.uth.gr.ktelvolou.pojo.BusLine;
import edu.uth.gr.ktelvolou.pojo.BusStop;

public class ArriveTimesArrayAdapter extends ArrayAdapter<BusLine>
{

```

```

private int mResource;
private int mTextViewResourceId;
private BusStop mBusStop;

public ArriveTimesArrayAdapter(Context context, int resource, int
textViewResourceId,
                                List<BusLine> objects,
BusStop busStop)
{
    super(context, resource, textViewResourceId, objects);

    mResource = resource;
    mTextViewResourceId = textViewResourceId;
    mBusStop = busStop;
}

@Override
public View getView(int position, View convertView, ViewGroup parent)
{
    TextView txv;

    if (convertView == null)
    {
        convertView = LayoutInflater.from(this.getContext())
                                    .inflate(mResource,
                                            parent,
                                            false);

        txv = (TextView)
convertView.findViewById(mTextViewResourceId);

        convertView.setTag(txv);
    }
    else
    {
        txv = (TextView) convertView.getTag();
    }

    BusLine item = getItem(position);
    if (item != null)
    {
        String[] titles =

```

```

getContext().getResources().getStringArray(R.array.arrive_times_titles);
    StringBuilder builder = new StringBuilder(item.toString());

    for (int i = 0; i < 3; i++)
    {
        if (item.getArriveTimes().containsKey(i))
        {
            builder.append("\n");
            builder.append(titles[i]);
            builder.append(": ");

            List<SimpleTime> times = item.getArriveTimes().get(i);
            int stopIndex = item.getBusStops().indexOf(mBusStop);

            Calendar c = Calendar.getInstance();
            SimpleTime now = new
SimpleTime(c.get(Calendar.HOUR_OF_DAY), c.get(Calendar.MINUTE));
            now.addMinutes(stopIndex);

            TreeSet<SimpleTime> sortedTimes = new
TreeSet<>(times);

            SimpleTime next = sortedTimes.higher(now);

            if (next == null)
            {
                next = times.get(0);
            }

            builder.append(next.toString());
        }
    }

    txv.setText(builder.toString());
}

return convertView;
}
}

```

## 3.6 Strings

Τέλος όλα τα strings (εκτός τετριμμένων περιπτώσεων) είναι γραμμένα στο αρχείο `res/values/strings.xml` για την εύκολη διαχείριση και αλλαγή τους.

## 4 Πιθανές Επεκτάσεις

### 4.1 Πιθανές Επεκτάσεις(Updates) για το “Αστικό Λεωφορείο Βόλου”

Η εφαρμογή επιδέχεται πολλές προσθήκες και βελτιώσεις που μπορεί να γίνουν για να την μετατρέψουν σε πιο πλήρη και πιο χρήσιμη. Σε μερικές περιπτώσεις μπορεί να χρειάζεται σύνδεση στο internet. Ενδεικτικά μερικές επεκτάσεις:

- Η δυνατότητα χρήσης της εφαρμογής και σε άλλες γλώσσες με την προσθήκη μίας άλλης οθόνης πριν την υπάρχουσα αρχική, όπου ο χρήστης θα επιλέγει ανάμεσα σε μία σειρά απο γλώσσες πατώντας πάνω σε ένα αντίστοιχο πλήκτρο με εικόνα την σημαία της χώρας.



- Η απεικόνιση με χρήση χαρτών των στάσεων, καθώς και των διαδρομών που θα ακολουθήσει το δρομολόγιο που επιλέγει ο χρήστης.
- Η δυνατότητα για την “βέλτιστη διαδρομή”, χρησιμοποιώντας αλγόριθμο που να υπολογίζει την πιο σύντομη διαδρομή ανάμεσα σε 2 στάσεις(αφετηρία-τέρμα) που θα επιλέγει ο χρήστης, αναλόγως τον συνολικό χρόνο και το σύνολο των μετεπιβιβάσεων.
- Προσθήκη κουμπιού “Νέα” στην αρχική οθόνη που να ενημερώνει για αλλαγές ή τροποποιήσεις σε δρομολόγια, εισητήρια ή οποιοδήποτε άλλο νέο που να αφορά το αστικό λεωφορείο Βόλου.
- Καλύτερη μορφοποίηση με προσθήκη εικόνων και πιο ελκυστικού μενού για τον χρήστη.
- Προσθήκη διαφημίσεων σχετικών με την εφαρμογή που να μπορεί να ενδιαφέρουν τον χρήστη.

# Βιβλιογραφία

[1] Το βιβλίο *Android Programming: The Big Nerd Ranch Guide* (Big Nerd Ranch Guides)

[2] Επίσημος οδηγός προγραμματισμού για συσκευές Android:

<http://developer.android.com/training/index.html>

[3] Επίσημος οδηγός χρήσης του Android Studio:

<http://developer.android.com/tools/studio/index.html>

[4] Εγχειρίδιο χρήσης του Genymotion: [https://www.genymotion.com/?utm\\_source=dlvr.it&utm\\_medium=twitter#!/developers/user-guide](https://www.genymotion.com/?utm_source=dlvr.it&utm_medium=twitter#!/developers/user-guide)

[5] Η περιήγηση σε διάφορα προγραμματιστικά forums/blogs για συγκεκριμένα τεχνικά ζητήματα ή και προβλήματα που παρουσιάστηκαν κατά τη συγγραφή του κώδικα. Τα κυριότερα είναι:

[5i] <https://stackoverflow.com/>

[5ii] <http://www.vogella.com/tutorials/Android/article.html>

[5iii] <http://www.insomnia.gr/forum/15-%CF%80%CF%81%CE%BF%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B1%CF%84%CE%B9%CF%83%CE%BC%CF%8C%CF%82/>

[5iv] <http://www.javacodegeeks.com/category/android/>

[5v] <http://android-developers.blogspot.gr/>