



Diploma in Software Development

DSE 730 Mobile Software Development

Assessment: Project Report

Total marks: 100

Course Weighting: 100%

Due Date: Wednesday, 14 February, 2017

Student Name(s): Gabriel Mauricio Melgarejo Sestrem

EXECUTIVE SUMMARY

The Diploma of Software Development course requires a huge potential for the students to improve their skills in computing and also to push themselves to share their knowledges and experiences.

In the Mobile Software Development, we have to be able to identify an interesting mobile software application which can be realistically completed in the time available.

In the first Stage, I have planned my project, building a Gantt Chart, considering technical implementations in Android Studio, predicting future risks and the project and how to mitigate them. After that, I have done a research into existing mobile software applications and talked to some friends and relatives who are already working in the IT market, to find the best Application Idea.

In the Architectural Design I have discussed about which alternative designs for the implementation of a prototype, architectural patterns, technologies to be used, etc. The User Interface (UI) started after building the first prototype using a Linear Layout, and then changed to Relative Layout to become easier for the user interaction.

“MyTripp App”, in the first version (1.0), developed in Android, using the Android Studios as the IDE, SQLite DataBase as a data storage. The application is still under implementations, tests and reviews. All stages of the application development will be detailed in the next topics.



Introduction

Aim of the application

The necessity to build an Application to record your trips and record photos, comments, ratings, locations, etc emerged last year, after make some trips around South and North America and realized my records was only disorganized photos in my phone's album.

The purpose of MyTripp App is to keep the organized form a detailed reminder of the places you have visited.

Target audience and likely market size

Initially the target audience will be my closer friends and some relatives. I'll use them to help me giving me feedbacks to keep improving the application for the following months.

I intend to turn it into a solid application to be able to advertise in the market and be able to help people who like to travel, and to have a consistent record of the places visited.

Technologies used



Android is a mobile operating system developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets.

The Official IDE for Android: Android Studios provides the fastest tools for building apps on every type of Android device.

Android provides many ways to store data, SQLite Database is one of them that is already include in android OS.

Availability of the same or similar applications on other platforms

There are current some trip applications available in the market, one of them is “TripIt”, focused in a pocket travel agent to helps you to organize flights, hotels, rental cars, etc.

MyTripp App is similar, but we have a different approach. MyTripp is focused in record memories of your trip as places you have been, photos, comments, ratings and dates.



TripIT App

Proposed pricing strategy

Making money with a mobile App It's isn't the easiest thing.

“Less Than 0.01 Percent of Consumer Mobile Apps Will Be Considered a Financial Success by Their Developers”. (Gartnercom, 2014).

There are four major pricing strategies: free, freemium, paid and paidmium. The Pricing strategy for MyTripp app is FREEMIUM Strategy.

Under freemium (today's most common model), the app itself is free, but there are in-app purchase opportunities.

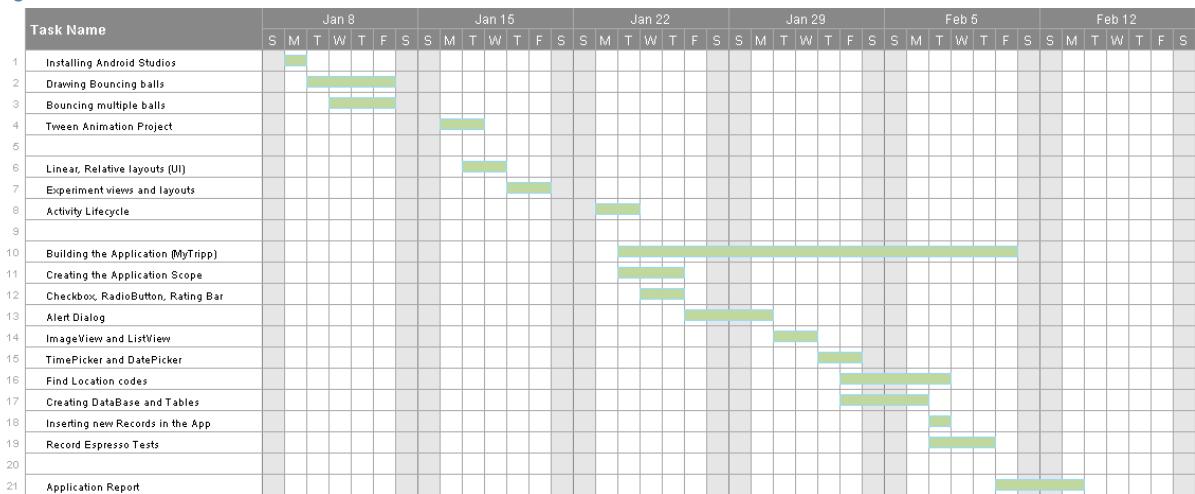
My plan is to make the App 90% free, and implement some extra features options to the Paid accounts.

MyTripp App Free version, will be some limited features (Add records, listing your records, update and delete than) and will come with Ads.

MyTripp App Paid version, will have all the features (Free features plus searching options, and google maps interaction) and without Ads.

Project Planning and Execution

Project Plan and Gantt Chart



Risk Management

According to the previous Assessment (Desktop Development), I have learned about how to list your risks in the beginning of the project and keeping controlling them (mitigate) to an acceptable levels.

As an inexperienced developer, the highest risk I can ever find is the difficult to solve problems during the phases of a project. Despite of that, the security of the application and how to meet the functional requirements on time was the principal risks.

Risks founded:

- Learning some new technologies (I was still learning how to use new technologies as Android Studios, SQLite);

- User and Functional requirements (The functional requirements have changed week by week);
- Application and System architecture (I was focused in learn new languages and the architecture wasn't the priority).

After cataloging all of the risks according to type, I crafted a risk management plan to monitor and mitigate them.

Monitoring tasks I've made to mitigate risks:

- Publishing project status presentations to class and include risk management issues
- Revising risk plans according to any major changes in project schedule
- Reviewing and reprioritize risks, eliminating those with lowest probability

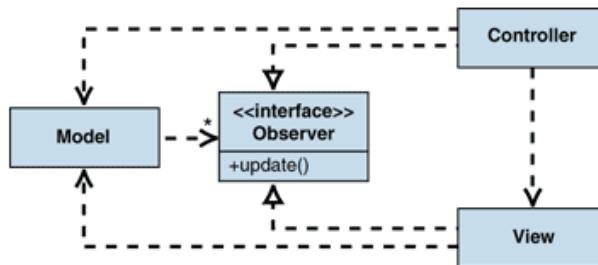
Every Mondays I've been checking the risks of my project and I've been mitigating them until to acceptable levels.

Architecture and Design

Alternative architectural designs considered

In my researches, I found different kinds of Architectural designs to Android Applications:

- The Model View Controller (MVC) pattern is a very popular approach for the development of a mobile application. The aim of this pattern is to separate the components of user interface (View); core functionality and data (Model) and the response to user inputs (Controller).



- The Layered Abstraction pattern consists of a hierarchy of layers. At each layer, we have components that work together within the same level of abstraction, with the layer below providing functionality for the layer above.

The Android architecture follows this Layered Abstraction pattern. It consists of four layers of abstraction with direct communication only between a layer and the layer directly above or below it: Applications, Applications Framework, Libraries, Linux Kernel.

Reasons for your choice

I will now explain some part of my code from MyTripp Application that follows this software pattern (Layered Abstraction). When a user uses the application to determine their current location, a background running task is executed from the Service component which sends a location request from Applications to the Application Framework layer. The location manager sees the request and provides periodic updates of the device's geographical location (latitude, longitude coordinates) with the help of the core system libraries in the Libraries layer. The system libraries communicate with the WiFi and GPS device drivers in the Linux Kernel which interact with the device's hardware to receive GPS signals. Once the signals are received the notifications are propagated back up the layers to the Application layer which updates the user with their location information.

```
}

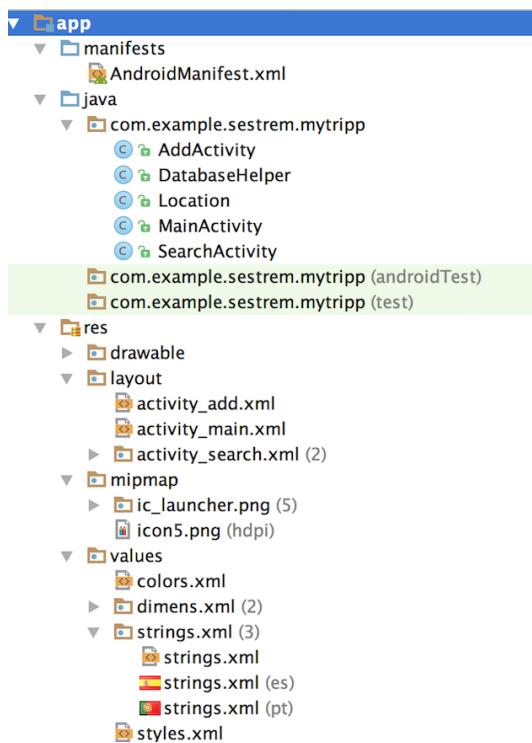
public String hereLocation(double lat, double lon){
    String curCity = "";
    Geocoder geocoder = new Geocoder(AddActivity.this, Locale.getDefault());
    List<Address> addressList;
    try {
        addressList = geocoder.getFromLocation(lat, lon, 1);
        if(addressList.size() > 0){
            curCity = addressList.get(0).getLocality();
        }
    } catch (Exception e){
        e.printStackTrace();
    }
    return curCity;
}

@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
    switch (requestCode) {
        case MY_PERMISSION_REQUEST_LOCATION: {
            if(grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                if(ContextCompat.checkSelfPermission(AddActivity.this,
                    android.Manifest.permission.ACCESS_COARSE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
                    LocationManager locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
                    Location location = locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                    try {
                        t_location.setText(hereLocation(location.getLatitude(), location.getLongitude()));
                    } catch (Exception e){
                        e.printStackTrace();
                        Toast.makeText(AddActivity.this, notFound, Toast.LENGTH_SHORT).show();
                    }
                } else {
                    Toast.makeText(this, "Not Permission Granted!", Toast.LENGTH_SHORT).show();
                }
            }
        }
    }
}
```

HereLocation Method

OnRequestPermissionsResult Method

Overall architecture of the final design. i.e., activities, services, intents



I have been using Java files and XML files to develop my Application.

In the folder java/com.example.sestrem.mytripp you can see the Java files. DatabaseHelper.java is responsible to create, insert, delete and connect to the Data Base.

The XML files in the Layout folder are responsible for the user interface layout. XML is commonly used as a data format on the Internet. If you want to access data from the Internet, chances are that the data will be in the form of XML. If you want to send data to a Web service, you might also need to send XML. In short, if your Android application will leverage the Internet, then you will probably need to work with XML. Luckily, you have a lot of options available for working with XML on Android.

All the application messages and texts were put in the Strings.xml to translate them in any languages (English, Portuguese and Spanish in this case).

Discussion of technologies used

The Android programming codes I researched in many videos of youtube. I had to find how to use DatePickerDialog and TimePickerDialog, Find current location by GPS and Wi-fi resources and how to open the camera device, take a picture and save into the DataBase.

ProgrammingKnowledge. (2017). YouTube. Retrieved 14 February, 2017, from <https://www.youtube.com/watch?v=czKLAX750N0>

Netsoul. (2017). How to open camera to take and save picture in android studio. Retrieved 14 February, 2017, from <https://www.youtube.com/watch?v=vd3QmsSLVil>

Filip vujovic. (2017). How To Get GPS Location In Android. Retrieved 14 February, 2017, from https://www.youtube.com/watch?v=QNb_3QKSmMk

Discussion of data organisation

The Data organisation is very simple at the moment. I have already implemented one table, called "trip4.db" and six fields: ID, Title, Date and Time, Photo, Comments and Rate.

The next task is to create a user login, and match the current records in the table "trip4.db" with the right user.

Basically, I will create a new table: "user.db" and create a Foreign Key to Match the user ID with the Records of that user.

Reflection

As a simple reflection, I can conclude how basic was my knowledge about architectural patterns. Either MVC or the Layered Abstraction have their utilities, depends of the technologies used and how big will be your application or how deep will be your approach. Nowadays is getting more common to see the MVC approach, to be easier to separate the View (user interface), from the model (schema) and the controller (back-end codes).

UI Design

Initial screen designs showing layout and user interaction



First Design in 22 January 2017

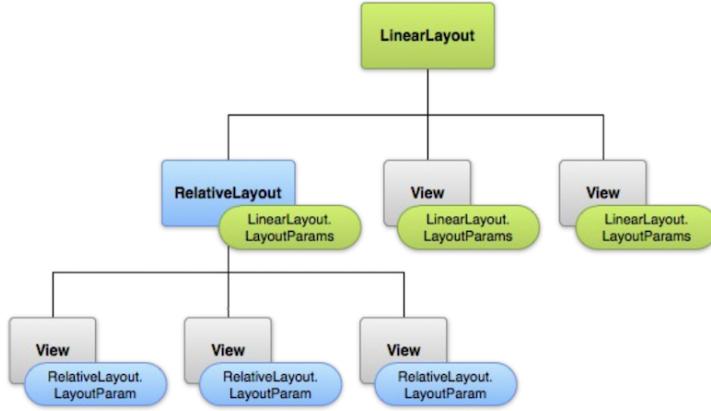


Second version in 25 January 2017

Final designs



Reasons for your final choices



I have used the IDE's User Interface creator. It allowed me to drag-and-drop views on to the User Interface. The result is saved in an .xml file in the layouts folder. I did the customization changing the values in the .xml file.

To be more interactive with the final user, this User Interface Design was chosen because I followed some best practices as:

- Design Multiple Screens (supporting different screen sizes);
- Showing Pop-up Messages;
- Creating customs views (interactive and smooth);

Reflection

After a deeply research, I realized some best practices for User Interfaces I haven't the opportunity to implement, for example:

- Build a Responsive UI with ConstraintLayout (build a layout using ConstraintLayout and the Android Studio Layout Editor), I have used the RelativeLayout;
- Adding the AppBar (Setting Up the App Bar, Adding and Handling Actions, Adding an Up Action, Action Views and Action Providers);
- Implement Accessibility (How to make your app accessible to users with vision impairment or other physical disabilities);

For the followings implementations, The focus will be to have a deeply approach in the User Interfaces Layout thinking about different devices and also different kind of users.

Implementation

Database schema

Columns (7)

Column ID	Name	Type	Not Null	Default Value	Primary Key	
0	ID	INTEGER	1	null	1	
1	TITLE	TEXT	0	null	0	
2	DATE_TIME	TEXT	0	null	0	
3	PLACE	TEXT	0	null	0	
4	IMAGE	BLOB	0	null	0	
5	COMMENT	TEXT	0	null	0	
6	RATE	FLOAT	0	null	0	

Tables (4)

- ▶ config
- ▶ sqlite_sequence
- ▼ trip4
 - ID
 - TITLE
 - DATE_TIME
 - PLACE
 - IMAGE
 - COMMENT
 - RATE
- ▶ users
- ▶ Views (0)
- ▶ Indexes (0)

Enter SQL

```
SELECT * FROM TRIP4
```

Run SQL Actions Last Error: not an error

ID	TITLE	DATE_TIME	PLACE	IMAGE	COMMENT	RATE
1	EDENZ COL...	05/02/2017 ...	AUCKLAND	BLOB (Size:...)	Nice to study	5
2	TAKAPUNA ...	01/02/2017 ...	AUCKLAND		Cool!	4
3	DEVONPO...	27/01/2017 ...	AUCKLAND	BLOB (Size:...)	Nice View	5
4	QUEEN ST	12/02/2017 ...	AUCKLAND	BLOB (Size:...)	Busy Street	5

Android technologies

The screenshot shows the Android Studio interface with the following components:

- Project Structure:** Shows the project tree with packages like app, manifests, java, and res.
- Code Editor:** Displays the MainActivity.java code, which includes imports for DatabaseHelper and ImageView, and defines a Main Activity extending AppCompatActivity.
- Emulator:** A Nexus 5 API 25 emulator is running, displaying a travel-themed app interface with icons for add, location, search, and a couple on a mountain.
- SQLite Manager:** A window showing the structure of Test1.sqlite database, listing tables such as Master Table, config, trip4, users, and sqlite_sequence.

Android Studios - IDE

Android Emulator - Nexus_5_API_25

SQLite - DataBase

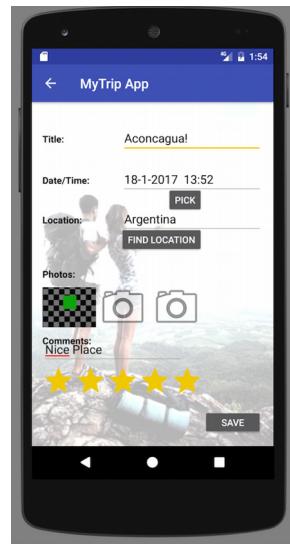
Screen shots



MyTripp App Icon



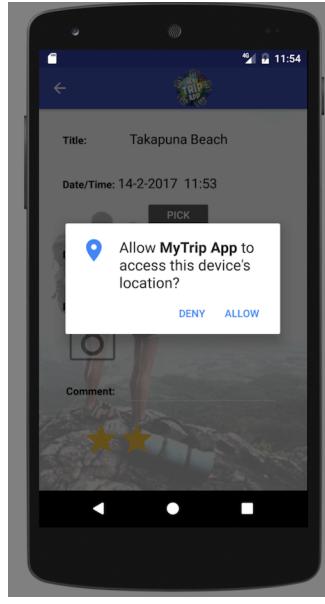
Main Activity



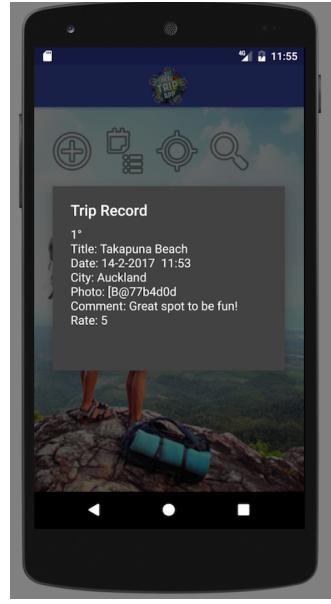
Add New Record Activity



Choose Date and Time using
TimePicker and DatePicker



Find Current Location By GPS
Position



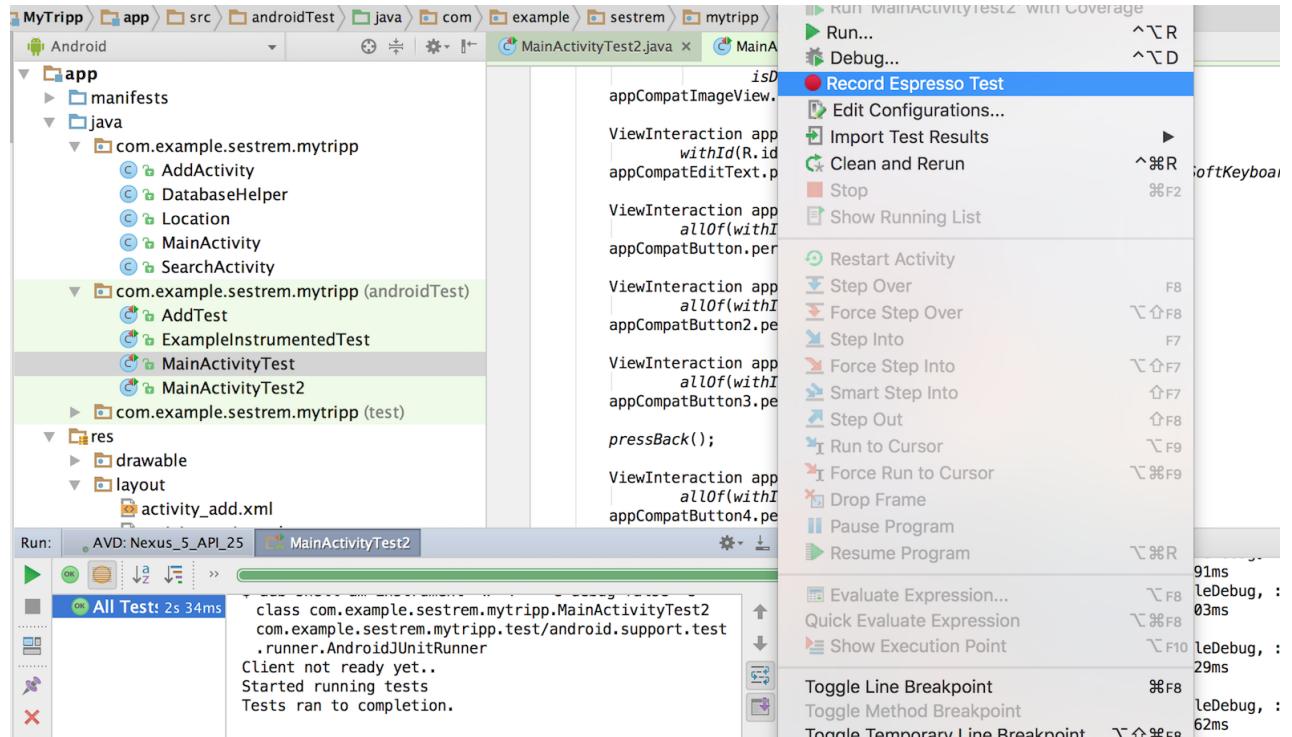
Listing your Records using
StringBuffer

Reflection

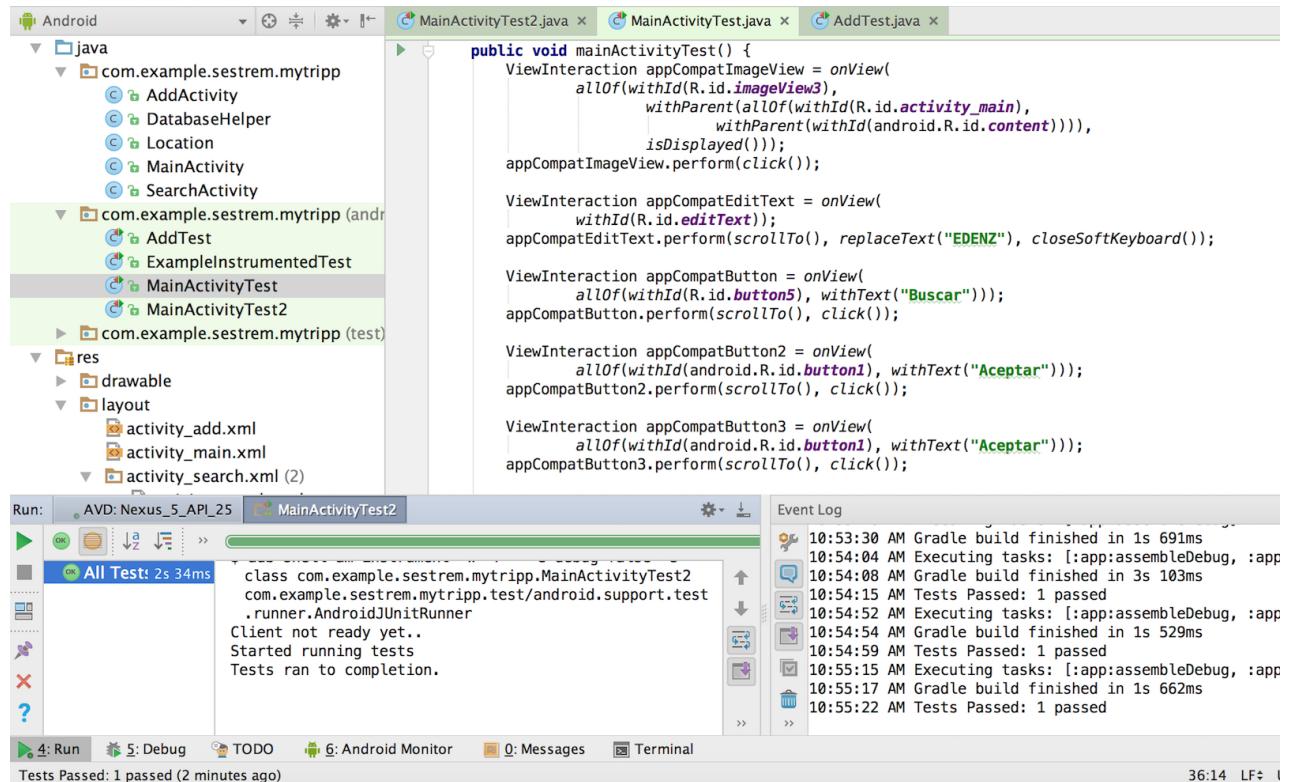
Discuss what you learned during this project about implementing new technologies.

Testing

Test planning



Espresso Test Recorder Tools.



Testing

I have used The Espresso Test Recorder tool. These tool lets you create UI tests for your app without writing any test code. By recording a test scenario, you can record your interactions with a device and add assertions to verify UI elements in particular snapshots of your app. Espresso Test Recorder then takes the saved recording and automatically generates a corresponding UI test that you can run to test your app.

Espresso Test Recorder is the newest UI Tests in Android Studio and is still a Beta Tool.

Remaining defects

During my tests, I realized many bugs and necessities in the application as:

- lack of validation fields;
- threads treatment while “onresume”, “onstop”, “onpause” activities lifecycle;
- Enable options of Delete and Update previous records;
- Keep a safety way to login in the application as an authentic user.

Reflection

The Espresso API encouraged me to create concise and reliable UI tests based on user actions. By stating expectations, interactions, and assertions without directly accessing the underlying app's activities and views, this structure prevents test flakiness and optimizes test run speed.

Conclusion

Reflection

Thinking about my learnings in the Mobile Assessment, I can say how useful It was be for me. I have learned many concepts I have not idea what was like Design Patterns and Design Architect.

It encouraged me to learn too much in a short period of time (5 weeks), studying how to improve my application, how to plan a new and useful application and how to test it.

Now I can say I am very confident in meet requirements, follow instructions and find how to use best practices of programming.

Summary

The Diploma of Software Development course requires a huge potential for the students to improve their skills in computing and also to push themselves to share their knowledges and experiences.

In the Mobile Software Development, we have to be able to identify an interesting mobile software application which can be realistically completed in the time available.

“MyTripp App”, in the first version (1.0), developed in Android, using the Android Studios as the IDE, SQLite DataBase as a data storage. The application is still under implementations, tests and reviews.

References

Gartnercom. (2017). Gartnercom. Retrieved 14 February, 2017, from
<http://www.gartner.com/newsroom/id/2648515>

Tripitcom. (2017). TripIt. Retrieved 14 February, 2017, from <https://www.tripit.com>

Ibmcom. (2017). Ibmcom. Retrieved 14 February, 2017, from
<http://www.ibm.com/developerworks/library/x-android/>

Androidcom. (2017). Androidcom. Retrieved 14 February, 2017, from
<https://developer.android.com/studio/test/espresso-test-recorder.html>

ProgrammingKnowledge. (2017). YouTube. Retrieved 14 February, 2017, from
<https://www.youtube.com/watch?v=czKLAx750N0>

Netsoul. (2017). How to open camera to take and save picture in android studio. Retrieved 14 February, 2017, from <https://www.youtube.com/watch?v=vd3QmsSLViI>

Filip vujovic. (2017). How To Get GPS Location In Android. Retrieved 14 February, 2017, from
https://www.youtube.com/watch?v=QNb_3QKSmMk