HAAGA-HELIA
University of Applied Sciences

Android Application Development

Yevheniy Dzezhyts

Thesis

Business Information Technology

2013

HAAGA-HELIA
University of Applied Sciences

Author Yevheniy Dzezhyts	**Year of entry** 2007
Title of thesis Android application development	**Number of report pages and attachment pages** 35
Thesis advisor Juhani Välimäki	

The purpose of this thesis was to investigate the design and implementation process of an Android application which uses a web service. The idea of the application was to join an Android user and an Android developer for creating better applications. In addition, the objective of this thesis was to develop a prototype which operates with the Parse web service.

The theoretical part of the thesis discussed the Android platform, its history, the compatible web services and the methods used in this thesis process.

The empirical part of the thesis describes the whole process of developing an Android application which uses a web service. The emphasis was set on finding out how a program would work with the Parse web service. The tools that were used during the development phase were Eclipse with the ADT plug-in, the Android SDK and the Parse SDK.

The result of this thesis project was a working Android application which can connect to the Parse web service. The application can input and retrieve data from it and display information to the user.

The thesis concludes that the study at hand can be extended with further development of the application by improving the design and adding more functionality. Moreover, it is possible to improve the user experience by creating a website for the application which would use the same web service.

Keywords
Android application, web service, list view

Table of Contents

Abbreviations and terms

Android	An open source platform designed for mobile devices(Gargena, 2011)
OS	operation system, system used for controlling device
Android SDK	developing kit for Android platform
JRE	- The Java Runtime Environment (JRE) provides the libraries, the Java Virtual Machine, and other components to run applets and applications written in the Java programming language (CNet, 2013)
IDE tool	Integrated development environment
Eclipse	An IDE tool which consists of workspace and plug-ins. It can be used to develop applications in Java and other programming languages.
Open source software	Computer software developers of which have made source code of the program available and licensed it under license which gives to software provides the rights to study, change and distribute the software to anyone and for any purpose. (St. Laurent, Andrew M, 2008, p 4)
Ubuntu	Operational system which is running on top of Linux Kernel
Linux Kernel	Unix like operational kernel, sofware which is used to run device.
ADT	Android development tool, package of tools which are used to develop Android application.
GUI	Graphical user interface
Terminal session	Gnome-terminal. It is one of the administrative tools in Ubuntu
Android Emulator	Container for running Android OS along with applications
XML	Extensible Markup Language is a specification for storing information (Golberg 2009, p. xii)

1 Introduction

This project is about the developing an android application. Nowadays, the gadgets are rolling the world. Many people cannot imagine even one day without their favorite mobile device. We use them for everything: find information, stay connected with our friends and families, find the way around, decide what to do, and many other things. But very often we come to the point when we would like to have an application for particular situation or for certain need, but there is no such one.

Developing of an application usually takes lots of time and needs professional knowledge of software. And then as people do not find the application they tend to wait until somebody is developing one, or they have to go to the web and ask people to implement their ideas. On different forums there are tons of brilliant ideas but they will wait until developers will see them. On the other hand there are lots of enthusiastic developers who are looking for ideas to implement them.

Need for the application

In my opinion it is a shame that there is no such an application that will keep the user and developers connected that is why good ideas tend to be lost. For this reason we need to develop application that will satisfy both groups and help them to share their ideas. Of course there are several sources on the web which might offer nearly similar services. Good example of it is the Best Ideas web site http://www.bestappideas.com/. But unique of our idea is to unite the end user and developer; on the other hand goal of Best Ideas is to join designer and investor though they can design an application. Such an approach might be more commercially effective but not so creative. No doubts that around whole ideas some can be silly but on the other hand can be brilliant. There are also many separate forms where people can share their ideas. But there are no dedicated ones which can serve as one source for the projects. And of course there is not a single application for Android where users can share their ideas. Based on this we can conclude that our application is needed for today's developing environment and it will be well demanded around android users.

Goals

The goals of this project are to develop application which will team up android developers and user. The application must have simple interface, server connectivity, group all applications into categories, inform any group of users about certain event and etc.

Also in this thesis project we aim to learn Android development process on example of this application development. In this thesis we about to learn: how to create an Android application, how to set up application layout, how to program in Java language for Android framework and also how to run the application on Emulator and any physical Android device.

Scope of the thesis

Due to the time limits of this project we must narrow the scope. The scope of this project is to develop a prototype of application which will connect to the Parse server and manipulate with information. The prototype must have input interface which will post to the server any entered information and listing interface where certain information will be retrieve from server and displayed to the user. Also inside the scope of this project to represent basics of Android development and one of the web services such as Parse.

Out of the scope is the further developing of application and creating web page for this application.

2 Research plan

Research plan contains development steps which we are about to go through and also methods which are going to be used during this thesis project.

2.1 Phases

It is very important to follow up phases as the pre requirements for every next phase it is the completion of the previous one. By following these steps we will be able to create well structured project along with thesis paper.

Create a product backlog.

The product backlog can be seen from Attachment 1. It defines desired features of the application. In the end of the thesis project we will indicate which features were implemented into the prototype and which will be postponed for further development.

Form the theory background and familiarize with android development.

There are many information sources which can be used as the base for development of android application: books, articles, tutorials, recommendations. We need to process this theoretical background in order to proceed to next step of development. During this step we need to analyze importance of each and every source, as for development an application can be more important practical knowledge rather than theoretical background.

Create eligible environment and procure all the necessary tools.

During this step we need to install and implement required environment for the further development of an android project such as: get the clean installation of OS such as Ubuntu. It is much better and recommends using fresh environment of OS. As it has less of unneeded software installed. In this way we can get rid of many coming up errors. Install Java runtime environment. Download and install Android SDK tool with Eclipse. As the application will use cloud infrastructure we need to choose the web service.

Design a model of future application.

During this step we need to design the future layout of the android application based on the recommendation of Android community and product backlog.

Implement concept into prototype application.

Implement the actual application using the environment. Implement layouts and classes. Also connect our prototype application to one of the web services. Describe principals and technologies used during creating an Android application.

Test the prototype

During this thesis we are going to perform functional testing base on the interface of our application with the help of Android Emulator tool. We will also design several use cases to test functionality of the prototype.

2.2 Methods

In our thesis we are going to use several methods. The methods can be theoretical and development methods. In this thesis work we need both groups of methods.

Theoretical method

Literature review. This method is going to be used due to need of gathering and processing theoretical base. We are going to use different books and articles also other sources like tutorials and forums. In this way by using different group of sources we can get more detail idea about Android, Android development, structure, principals and technologies which are used along development an Android application.

Development method

Rapid application development method aims to produce high quality system in low time cost schedule. (CMS, 2003)

3 Theory Background

Background theory in this thesis work serves as prophase for developing an application. That allows us to understand more compatibly the principals and technologies of Android development and can give us idea about further structure of prototype project.

3.1 Android framework

Android is one of an Open source platforms. It is created by Google and owned by Open Handset Alliance. It is designed with goal "accelerate innovation in mobile" As such android has taken over a field of mobile innovation. It is definitely free and open platform that differs hardware from software that runs on it. It results for much more devices be running the same application. Also it gives possibility of friendlier environment for developers and consumers. Android it is complete software package for a mobile device. Since the beginning android team offers the developing kit (tool and frameworks) for creating mobile applications quick and easy as possible. In some cases you do not specially need an android phone but you are very welcome to have one. It can work right out of the box, but of course users can customize it for their particular needs. For manufactures it is ready and free solution for their devices. Except specific drivers android community provides everything else to create their devices. (Gargenta 2011, 1-6)

3.2 **Android history**

The actual history of Android starts when Google has had purchased and Android inc. in 2005. But the development did not start immediately. The actual progress on android platform starts when 2007 Open Handsets Alliance has announced the Android as Open Source platform and year later the Android SDK 1.0. In the same 2008 the G1 phone was produced by HTC and was retailed within the T-Mobile carrier. In the next two years came out 4 versions of Android. In 2010 there were at least 60 devices running android and it becomes second after Blackberry the best spread mobile platform. (Gargenta 2011, 3-6).

3.3 Development methods used

There are several methods which can support developing process of any project such as:

- Waterfall
- Prototyping
- Incremental development
- Spiral development
- Rapid application development
- Agile software development
- Object oriented
- Top-down programming
- Unified process (CMS 2008)
- System testing.

But for our needs, due to limited time of this project, only two are the most suitable: rapid application developing and like part of it prototyping and also system testing.

3.4 Rapid application development method

This method is directed by producing high quality system in most efficient time costs. Also it allows to divide project into smaller parts and develop them separately. It's basic idea to produce high quality system quickly by use of interactive Prototyping. It emphasizes on satisfaction the business requirements. It sets strict deadlines and if project slips the schedule, the emphasis is set to reducing requirements and not on changing deadlines. The prototypes are giving to the user for evaluation. (CMS 2008)

3.5 System testing

System testing in this thesis work will be represented by Specification-based testing. Testing efforts to rich certain level of perfection by detecting all the faults which are in implementation The best way to perform any testing is to select test case that satisfy certain criteria. These criteria are depending on the nature of application and the scope. (Laycock, 1993)

3.6 Literature review

The literature review method is an examination of information on specific subject. It is reviewing what is known and not what is assumed. It aims to create the final, precise representation of the knowledge and research-based theory available topic. (Dawidowicz 2010, 5-17)

3.7 Cloud computing and Web services

In the past years, due to the rapid development of mobile frameworks, the need for storing and sharing information throughout different devices has grown. All the process from installing to running most of applications is maintained by using different web sources and services. Insofar as there was need for such product, many companies have established cloud and other mobile services. Companies like Amazon, Google and Parse would offer their web services for modest payment or in some case for free. (Parse, 2013)

Cloud computing is based on the idea that large information sources can provide data to any application which will call for it. The clients of cloud can be smartphones, tablets, and mobile Internet devices. In the case of cloud application design we do not need any more to use storage capabilities of device. Only what is mater are network bandwidth and display capabilities.(Sosinsky 2010, p. 431)

We will present one kind of web services on example of the source “Parse”. Basically it is the same hosting service, but there are major differences which are making this kind unique and most convenient for developers use.

Parse would offer their version of Parse SDK. This is an empty project that has needed classes to process cloud objects and push them to the web service. On that level of integration it will be very convenient for programmer to use them. That will help to save time cost which would take normal database integration.

Of course Parse is one of the commercial services but it has also free option which is limited by the number of requests, pushes per month and also mobile traffic per second.

4 Development

In this chapter we will design and implement layout with help of XML programming language and Eclipse programming environment. Also we will add Java classes and method to make program match requirements.

4.1 Reviewing the requirements

As was stated in the product backlog (Attachment 1) there are several needs for the application prototype. During this thesis process we will create prototype which will host only part of the stated requirements. For the prototype functionality are most important input, output and connection to the web service.

4.2 Structure of an Android project

The structure of android project is mostly the same, but also may differ depending on the project needs and IDE tool. We will describe basic structure while using the ADT. When programmer uses ADT the project structure is generated automatically. Even further, ADT is also generating the ready- made application "Hello word". The GUI version of ADT is the easiest way to create an Android project but the advance programmer can be also using the set of tools which can be run in terminal session. The terminal tool called "ant" can debug the Android project and create sample structure even if developer uses any other programming tool than Eclipse. (Jordan and Greyling 2011, p. 60)

Project folder structure

Basic Android project would have six directories such as: assets, bin, gen, libs, res, src. Also there are some files in project root directory such as: AndroidManifest.xml, licenses, project.properties and other files. (Android Developers, 2013)

The most important for the developers are “res” and “src” directories.

“res” directory contains all the current project resources such as: images, layouts, custom strings and other values. Images are stored in different directories depending on their size that application can automatically choose right image depending on the device specifications. Layouts are store in the “layout” folder. Basically layout file example would be an XML file which would specify elements and their position in current view. Also it is possible to code custom strings and colors so the parser can display them in application. It is recommended approach to store them in values directory rather than hard code to the actual code or XML file. It would make easy further development at translating the application to other languages.

The other important directory is “src”. This directory would usually consist of Java files which are adding functionality to the application. Then developer would create classes as separated Java files. If the class is created in GUI ADT environment the tool would generate automatically the statement in “AndroidManifest” file. If other programming environment is used, user must specify any new class activity by hard coding the “AndroidManifest” file. (Lee, 2012)

Android manifest file usually would be placed to the root directory of the project and state required version of android, needed permissions and all activities which are run within the application.

4.3 Designing of the layouts

On the requirements base will be designed four different views. The idea of layout was brought from one of the discussing forums and tested in our project.

First view

This view will consist of greeting for the user and two large buttons. The buttons will give user choice to proceed to the new view depending on his choice. First option will be – to see all posted ideas and their status. This option will be handled by List Ideas view. The second option will be – to enter an idea. This function will handle Add Idea view.

List ideas view

This view will list all the ideas. The single idea representation will consist of an image and tree text views. Text views will display information about idea such as name, short description and state of application.

Add Ideas view

This view will offer to the user to enter the name and description of the application.

4.4 Implementing the layouts

As was stated before, android application uses XML layout for displaying its content. XML document would consist of several tags with given properties. The parent tag would state type of view which is the main for the document. Also it is allowed to use several views inside of one main view.

Like any other XML tag, this tag is given several properties which will define its identification, style, onclick action, etc. Identification is one of the important part of the tag. By defining the "id" programmer can use it in the Java code. Styles can be hard coded in the statement or linked to separate file which will specify style for this element. Action properties can call the event action which is defined in the code part of the project.

Main view

The main view of prototype application will have simple look. For the organizing with the help of LinearLayout statement. This statement is parent layout statement for main view XML document. That is why it has to be linked to the http://schemas.android.com/apk/res/android.

The example of parent statement for XML document would look like this

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
</LinearLayout>
```

As we can see this statement hosts some properties which are important for displaying of this document. Xmlns:android could define namespace information but in this case it just defines of unique of the parent tag as "http://schemas.android.com/apk/res/android" does not exist.

Layout_width tells to the mobile device how to display this view. In this case option "fill_parent" is used so the program will use all the space on horizontal scale.

Layout_height is the same as width but it uses vertical scale to display the view.

Orientation is to display view depending on device orientation. It can be vertical or horizontal. In this case it is horizontal so it means that device's screen is used in portrait mode.

All the child entries will be entered between the end of definition and closing statement </LinearLayout> .

The child of linear layout can be any view: button, text view, etc. It can also host the other linear layout.

ListView

List view has a bit more complex structure than main view. In our case list view is created with the help of two separate XML documents. The main document will have List view statement and will define its position in the document. The adapter view will define position of the element inside one item of the list. It is possible to use standard list adapter which is provided by android framework, but in this case custom made adapter is more suitable. Later each item will be appended in the list with the help of Java code.

The example of list statement is normal XML tag which is using ListView definition.

```
<ListView
    android:id="@+id/lvList"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
/>
```

In this case layout_height is set to “wrap_content”. This means that the high of this element will be set depending on it's entries and their size.

The example of adapter for the list view can be created using any layout in this case is used relative layout. It means that all the elements will be situated depending on their position compare to the previous element. This is allows to design better layout and use less device resources compare to the linear layout.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="horizontal"
>
<ImageView
    android:id="@+id/icon"

    android:paddingTop="10dp"
    android:paddingRight="10dp"
    android:paddingBottom="10dp"
    android:layout_width="70dp"
    android:layout_height="70dp"
/>


    <TextView
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="15dp"
        android:text="TextView1"
        android:gravity="top"
        android:layout_toRightOf="@id/icon"

    />

    <TextView
        android:id="@+id/description"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="10dp"
        android:text="TextView2"
        android:layout_toRightOf="@id/icon"
        android:layout_below="@id/name"
    />
</RelativeLayout>
```

As we can see from the application code relative layout hosts three elements: image view and two text views. Image view is used to displays image and text views display text. The statement android:id=*"@+id/icon"* in android development will allow not only to create element but also define it in the xml tag. This statement is not compulsory and can be used when programmer wants to use it in code of the program.

The position of the element is set with help of android:layout_toRightOf=*"@id/icon"* .

This statement sets this element right of the element flag. Like lenear layout relative will start using screen of the device starting from top left corner. Then programmer needs to set following element by considering this fact. Also statement android:layout_below=*"@id/name"* will position element below of the element txt. Then statement layout_above will have opposed meaning and will place this element above some other element.

The idea of the list_layout was brought from one of the discussing forums and tested in this application. (Stackoverflow, 2009)

Add idea view

Add idea view is the view where ideas are inserted. The parent of this view will be linear layout oriented vertically. It will consist of two edit text elements and one button. In first edit text element user will enter the name of idea for application, in the second edit text element user will enter description of idea and its need.

The button will be located in the end of document and when pressed will save entered data to the server.

The implementation of this layout will look like this.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingBottom="@dimen/activity_vertical_margin"
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   tools:context=".AddIdea"
   android:orientation="vertical">

   <TextView
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Name Of App" />

   <EditText
       android:id="@+id/etNameOfApplication"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:ems="5"
       android:inputType="text" >

       <requestFocus />
   </EditText>

    <TextView
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Decription of Application" />

   <EditText
       android:id="@+id/etContentOfApplication"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:layout_weight="1"
       android:ems="10"
       android:inputType="textMultiLine" >

       <requestFocus />
   </EditText>

   <Button
       android:id="@+id/btPost"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Post To Server"
       android:onClick="pushToServer"
       />

</LinearLayout>
```

Also as we can see from example there are several text views which will give user the idea how to use application.

Detailed view

Detail view consists of linear layout and two text view elements. The layout has vertical orientation so the elements are located according to the vertical scale. The example detailed layout looks like this

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/tvDetailedName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:layout_margin="5dp"
        />


    <TextView
        android:id="@+id/tvDetailedDescription"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="15dp"
        />

</LinearLayout>
```

The name of text views are tvDetailedName and tvDetailedDescription and their will be referenced in the program code. Both text views have android:textSize parameter which would set any text pasted to this view to the certain text size.

4.5 Designing and implementing the classes and functions

Java programming language is used in android development. All the activities are organized in classes and triggered by certain events.

Every android application has its starting activity. In the default project it would be main activity class but since we are using SDK from Parse.com our starting activity differs from the default one.

Main Activity

By Parse developers are created two classes: ParseApplication and

ParseStarterProjectActivity. In ParseApplication class we will initialize our application and register it in Parse.com. It is made with the help of initializing statement

```
Parse.initialize(this, "EjEB6DbRkBSUDG8eBo5UJxUvhZ3SghAPXjOcJzzK",
"TXhlSWvGKSjxhMf3g5Nc8AVC4HU5z8HPLeO5GWPI");
```

In this statement we will send to Parse id of our application and key which is required for initializing of application.

ParseStarterProjectActivity is the main activity for our application. As we need some event to trigger code executing we have onCreate method.

```
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);}
```

This method is run when application is started. Inside of this method we will set the layout for application. It is made with the help of setCounterView method.

```
setContentView(R.layout.main_1);
```

We can see from example this command sets the layout main_1.

As we have created this layout before and it has two buttons, we must set up methods which are triggered by pressing those buttons. The methods will not return any values and just take user to the next state depending on his choice. As these methods will not return any values they can be set as void method and since we would like to use them inside of whole application we must set them as public.

```
public void listIdeas(View view){

    Intent intent = new Intent(this, ListIdeas.class);
    startActivity(intent);

}
```

This method will take as parameter view which is button itself and it will trigger new intend. Basically intend in android development it is instants which will run code in the desired class. From this example we can see that this intend runs ListIdeas class. All intends are defined in AndroidManifest file.

```
<activity
            android:name="com.parse.starter.ListIdeas"
            android:label="@string/title_list_view" >
</activity>
```

Add Idea

This class will have two methods.

OnCreate method will be called when activity is created. It will set add_idea layout.

```
setContentView(R.layout.add_idea);
```

As was described before add_idea layout has a button. On the button click will be called second method pushToServer. This method will read edit text elements then create parse object add important properties to the object and save it to the server.

```
EditText appName =(EditText) findViewById(R.id.etNameOfApplication);
EditText appDescription = (EditText) findViewById(R.id.etContentOfApplication);

ParseObject appIdea = new ParseObject("Idea");
appIdea.put("name", appName.getText().toString());
appIdea.put("decription", appDescription.getText().toString());
appIdea.saveInBackground();


Toast.makeText(getApplicationContext(),"SAVED", Toast.LENGTH_SHORT).show();
```

For this method important Internet connection but saving can be organized in off line mode and then pushed to the server. In this example we will assume that device has active Internet connection.

This method will give feedback which is important for the user experience. It will generate Toast message "SAVED". Then user can assume that information is saved on the server.

List Ideas

It is the most important class of our application. This class will have three methods. Oncreate method will set list_view layout. Also in this method we will make connection to the parse server and execute the query. In the query we read certain field of the object: name and description. Then read data will be parsed by custom made parse method to the list format and set to every single item of the list. For this reason we create list of hashMap items. For parsing data in the new created list we use loop. In the loop we add name and description field and also image which will be logo of application. After this we need to use adapter which will generate layout and enter data to the generated layout elements. For this reason we use SimpleAdater.

```
SimpleAdapter adapter = new SimpleAdapter(getBaseContext(), aList,
R.layout.listview_layout, from, to);
```

alist element is the name of our list. Also adapter needs to know which values of the list we are using and into what elements we want to insert data. For this are used two arrays “from” and “to”. “from” array will indicate what field we want to use:

```
String[] from = { "icon","name","description" };
```

“to” array will consist of layout element which we are using:

```
int[] to = { R.id.icon,R.id.name,R.id.description};
```

At the end we must to set up adapter to the list by statement

```
listView.setAdapter(adapter);
```

After this all list items will be generated and displayed to the user.

Also inside of onCreate method we need to initialize onItemClickListener.

```
public void onItemClick(AdapterView adapterView, View view, int position, long id) {

      detailView(results.get(position).getObjectId());
         }
```

This method will be executed when user clicks on single list item. As we cannot call new intent from Oncreate method we have to design method which will call new intent and pass id of the chosen item to the new activity.

New method is called detailView. It will take as parameter string. Here we enter object id and send it to the new activity.

```
public void detailView(String id)
    {

Intent intent = new Intent(this, DetailView.class);
intent.putExtra("id", id);
startActivity(intent);

    }
```

Detail view of idea

This class will display detail information of chosen item. For that purpose we pass object's id from previous activity. Parse has special method which can get item depending on its id. Then we would run this method with supplying id from previous activity.

```
ParseQuery<ParseObject> query = ParseQuery.getQuery("Idea");
        query.getInBackground(id, new GetCallback<ParseObject>() {
          public void done(ParseObject object, ParseException e) {
            if (e == null) {

                tvName.setText(object.getString("name"));
                tvDescription.setText(object.getString("decription"));

                                }
             else {
              // something went wrong
            }
          }
        });
```

When we get the object we can extract object properties. In this case we are getting only name and description of the Idea. Then we assign it to the text views which were defined in detail_view layout. After this device would run and display text views with information inside.

4.6 **Functional testing of application prototype**

As was defined in the scope of this project we will perform functional system testing by using Specification-based testing method. For this reason we will design several use cases where user will: first enter his idea and then view it in the list of other ideas.

Table 1. Use Cases

	Use Case	Function Being Tested	Initial System State	Input	Expected Output
1	Application Startup	Application is started when user presses application item	Application is not active	Press application's icon	Application runs default activity and sets main layout.
2	Application Startup	Application sets add_idea view when user presses Add Idea button	System is on default view is displayed	Press Add Idea button	Application runs AddIdea activity and sets add_idea layout.
3	Posting to the server	Application reads input field and sends information to the server	add_idea layout is set active	Press button Post To Server	Information is being uploaded to the server, system confirms action by Toas message "SAVED"
4	Retrieving information from the server	Retrieve and display information to the user inside of list view	System is on default view is displayed	Press button List Ideas	ListIdeas activity is run and list of items displayed to the user
5	Display detail view of Idea	Display detail view of Idea after certain item in the list was clicked	List of item is displayed	Press any item from the list	DetailView activity is run the more detailed information of chosen idea is displayed to the user

Attachment 2 contains results of test use cases.

Table 2. Result of test use cases.

Use Case Number	Test result	Match Expected Output
1	Picture 1. Default view	Yes
2,3	Picture 2. Add Idea View	Yes
4	Picture 3. Listing Ideas View	Yes
5	Picture 4. Detailed View	Yes

5 Evaluation and Conclusions

Our project has resulted into working prototype which can perform simple activity with posting to the server and retrieving data. This prototype is fully functioning and brings the structure which can be used in further development of application.

The scope of this thesis project was to create user input interface and connection to the one of the cloud services such as Parse. Also inside the scope was to create listing interface, which would display information which is stored on the server.

The prototype which was mentioned before can handle input and push information to the server. The layout add_idea has two fields where user can enter name of possible application (idea) and short description of his/her needs. The posting to the servers is handled by the AddIdea class. Add idea class would retrieve information from the server and post it to the web service.

Retrieving of information is done with the help of the main activity class and DetailView class. It will retrieve information and generate the list view where all the items will be displayed. The user has option to click on any item from the list then more detailed information will be displayed.

We also have described used technologies and way of integrating cloud service into application.

5.1 Evaluation of results

Due to the tight schedule we did not have enough time to develop application that would contain full features which were stated in the goals chapter but the prototype of the application is representing the scope of the thesis project and host properties which were defined in the scope.

5.2 Source evaluation

The android platform which is chosen for the application has lots of information and tutorials which can help us while developing the application.

Such source as http://developer.android.com was one of the bases for further development. This web source is very informative and guides how to build application from scratch. Same source http://developer.android.com/sdk/ can help us to build the important environment.

What we have seen during creating the prototype application that the information which is needed the most and described in most clear way can be found from the forums. The web forum like StackOverflow can supply more useful information than the particular book about Android development. Besides this, information which is available in the forums can be accessed with no limits when the book must be purchased in advance. Based on this we can conclude that during development of the application, forums were more efficient than books.

5.3 Evaluation of methods

The methods which were used in this thesis work were very useful. The main theoretical method was literature review. Literature review method allows us to review different sources like books and articles and present our knowledge based on ideas described by other scientists.

The development methods were represented by rapid application development method and software testing method. As rapid application development method was used on any stage of development during this project we always had working prototype. Periodically we add more and more functionality to our prototype until it could satisfy the requirements, which were defined in the scope of this project.

System testing methodology was represented by Specification-based testing. With this method we could test the prototype on the base of requirements analyzes.

5.4 Evaluation of process

The process of this project can be defined as rapid development with getting actual result. As normal thesis project would last for six month this project was done for one and half month. The completion of this work was due to hard work of thesis advisor and student. This process was divided into phases and was followed by one time a week reporting and later two times a week reporting.

Tools which were used in this project were acquired free of charge. This project does not contain any private code and can be review as detailed as needed.

5.5 Evaluation of learning

As during this project different technologies were used we have learned a lot about Android framework and Android development. We also have learned how to create the Android application, how to design and implement layout, how to add functionality to the project and how to use web service. Also we investigate how to set up and use virtual Emulator of Android OS and real Android device.

6 Summary

In this thesis prototype of application was developed. The application can post to server and retrieve information from the server. It can be installed and function on the several devices at the same time. The prototype is working on android platform and made on the base of Java framework. It uses one of the web services to store and to receive information. The information is stored on-line and can be accessed in any time.

During this thesis project we have gone through several phases: creating product backlog, forming theory background, creating environment for developing application, designing model of application, implementing prototype, performing testing.

Methods which were used during the thesis process are literature review and rapid application development.

We have investigated history of android and technologies which are used during the Android application development. Also we have done research about web services and have chosen one of the best which is Parse.

On the base of requirement analysis we have created layouts for prototype. There are as many as four major layouts which are: main layout, add idea layout, list idea layout and detailed view layout.

The functionality was added to the project with the help of four major Java classes, which are AddIdea class, ListIdeas class, DetailView class and main class which is called ParseSrarterProjectActivity. Also we have one more class which is supplied by the Parse for identification of our client.

6.1 Further research questions

It is possible to add more functionality for the application. Also create a list of top developers, rate projects and developers activity according to their input to development. Copyright policy must be defined in advance. Organize private groups of developers and idea makers. Add other social network support. Create one of the web sources for example a web page which can be accessed from other devices with different OS.

Registering application with Google service and uploading it into Google Play Store.

References

Android Developers http://developer.android.com accessed on 12.2.2013

Cnet *JRE* http://download.cnet.com/Java-Runtime-Environment-JRE/3000-2378_4-10009607.html#ixzz2mLmJ28Kg accessed on 12.2.2013

CMS http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf, accessed November 3, 2013.

Dawidowicz, Paula 2010 *Literature Reviews Made Easy: A Quick Guide to Success.* IAP.

Gargenta, Marko 2011 *Learning Android.* O'Reilly Media, Inc.

Goldberg, Kevin 2009 *Visual QuikStart Guide.* Berkeley, CA 94710
How to Display List of Images in ListView in Android? http://stackoverflow.com/questions/459729/how-to-display-list-of-images-in-listview-in-android, accessed November 4, 2013.

Jordan, Lucas, and Pieter Greyling 2011 *Practical Android Projects. Apress.*

Laycock, G. T. (1993). *The Theory and Practice of Specification Based Software Testing.* Dept of Computer Science, Sheffield University, UK.

Lee, B. W.-M. (2012). Android Application Development Cookbook. In *Android Application Development Cookbook: 93 Recipes for Building Winning Apps.* Indiana: John Wiley & Sons, Inc.

Parse. (n.d.). *Parse documentation*. Retrieved 2013, from Parse: https://parse.com/docs/android_guide

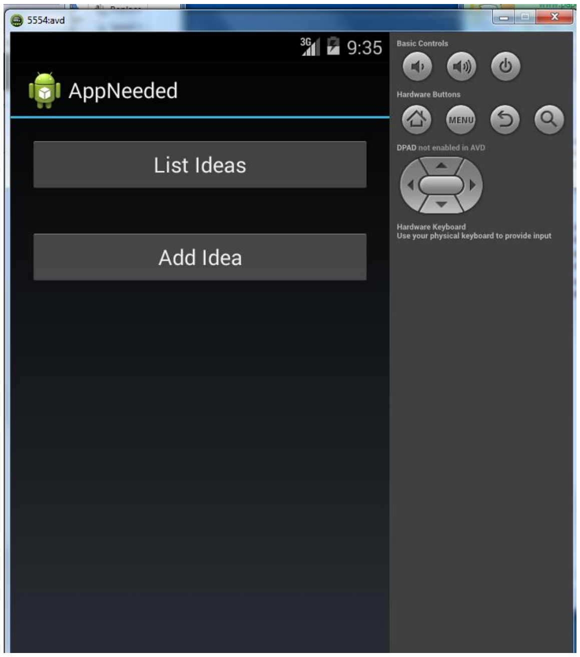
Sosinsky, Barrie, 2010 *Cloud Computing Bible*. John Wiley & Sons.

St. Laurent, Andrew M. (2008). *Understanding Open Source and Free Software Licensing*. O'Reilly Media.

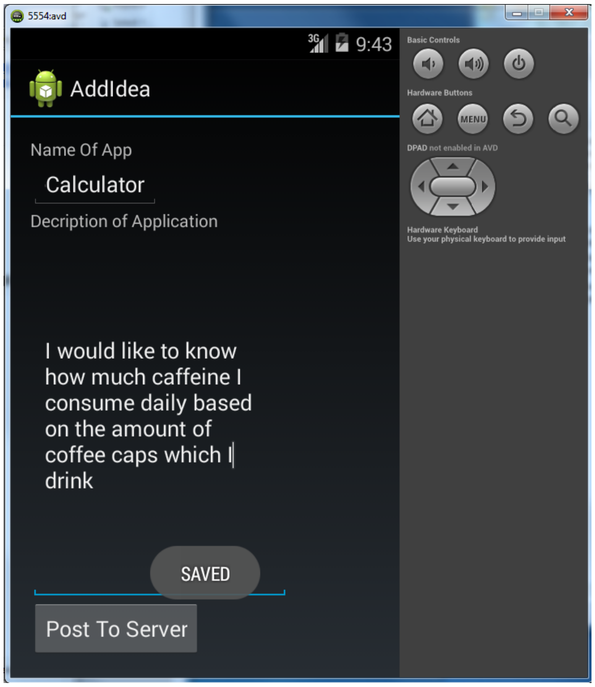
Attachment 1. Product Backlog

1	**Application must have an input option**. User should be able to insert certain information which later will be available through other devices	Done
2	**Application must be able to submit information to the server.** After user would click a button information would be send to the server and stored there.	Done
3	**Application must be able to read information from the server.** Access certain item depending on the user choice	Done
4	**Information which was retrieved from the server must be displayed in the list view.**	Done
5	**Group the ideas depending on certain criteria.**	Postponed
6	**Display information if idea is processed by some developer.**	Postponed
7	**Notify that idea or problem was resolved and solution is ready.**	Postponed
8	**If the mentioned idea is already implemented users must be able to point out ready solution.**	Postponed
9	**App must have option to display top/common solutions and also rate them.**	Postponed
10	**Application must list developers and rate them if needed.**	Postponed
11	**There must be some internet source which can be accessed from other platform for example web page.**	Postponed

Attachment 2. Test case validation screenshots



Picture 1. Default view



Name Of App

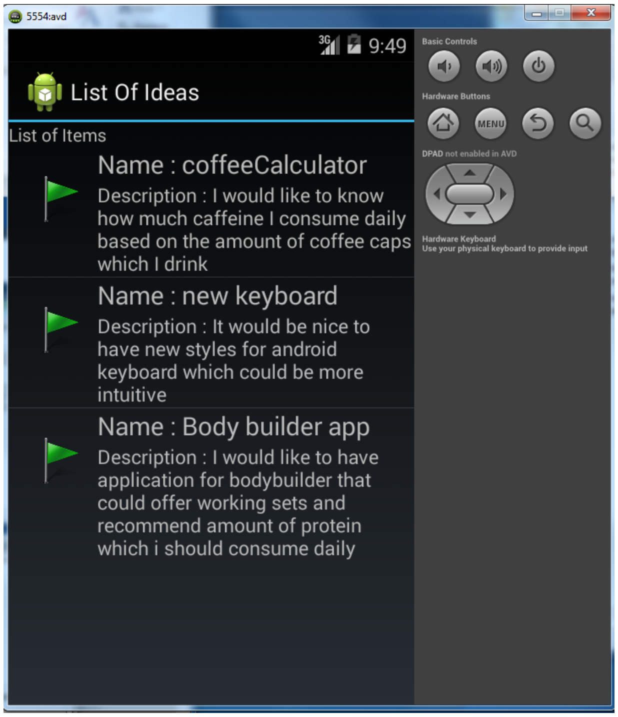
Calculator

Decription of Application

I would like to know how much caffeine I consume daily based on the amount of coffee caps which I drink

SAVED

Picture 2. Add Idea View



List Of Ideas

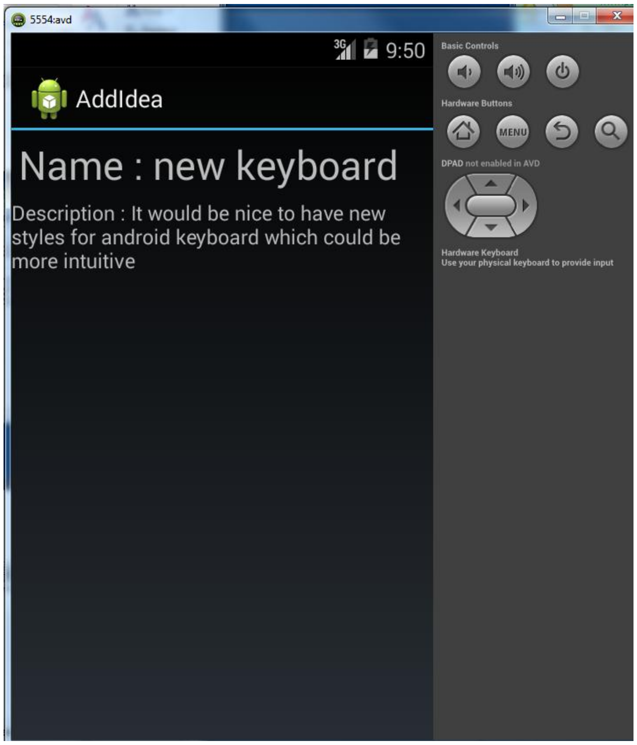
List of Items

Name : coffeeCalculator
Description : I would like to know how much caffeine I consume daily based on the amount of coffee caps which I drink

Name : new keyboard
Description : It would be nice to have new styles for android keyboard which could be more intuitive

Name : Body builder app
Description : I would like to have application for bodybuilder that could offer working sets and recommend amount of protein which i should consume daily

Picture 3. Listing Ideas View



AddIdea

Name : new keyboard

Description : It would be nice to have new styles for android keyboard which could be more intuitive

Picture 4. Detailed View