PROJECT INCREMENT I

PROJECT TITLE: TOUR LOGGER

BY

KAPA RAJESH (CLASS ID: 25)

GUMMADI RAGHU VITAL(CLASS ID: 20)

Existing Services/API:

For the Increment 1, the API's used are Google api and Facebook api. The Sign in process in the application is designed in such a way that the end user can sign in using his google account, Facebook account, own log in or register for the application. So in order to implement those facilities, some existing services are required. Their functionality is discussed below.

Google API: The api provided by google allows the user to log in through his google account. It authenticates the user and allows him to log in directly through his account.

Facebook API: This Facebook api allows users to log in with their Facebook account. It allows the application to use the details of the user from his Facebook account and allows him to log in directly.

Widgets Used: The widgets used in this increment are mainly the button widget and the text field widget. These are helpful for a better user interface. The Facebook button widget is designed from the Facebook SDK.

Detail Design of Services:

User Stories /Use Case :

For iteration#1 below are the stories

1. Initial setup for project: Project setup in android studio and database integration into the project. Analysis of all scenarios.

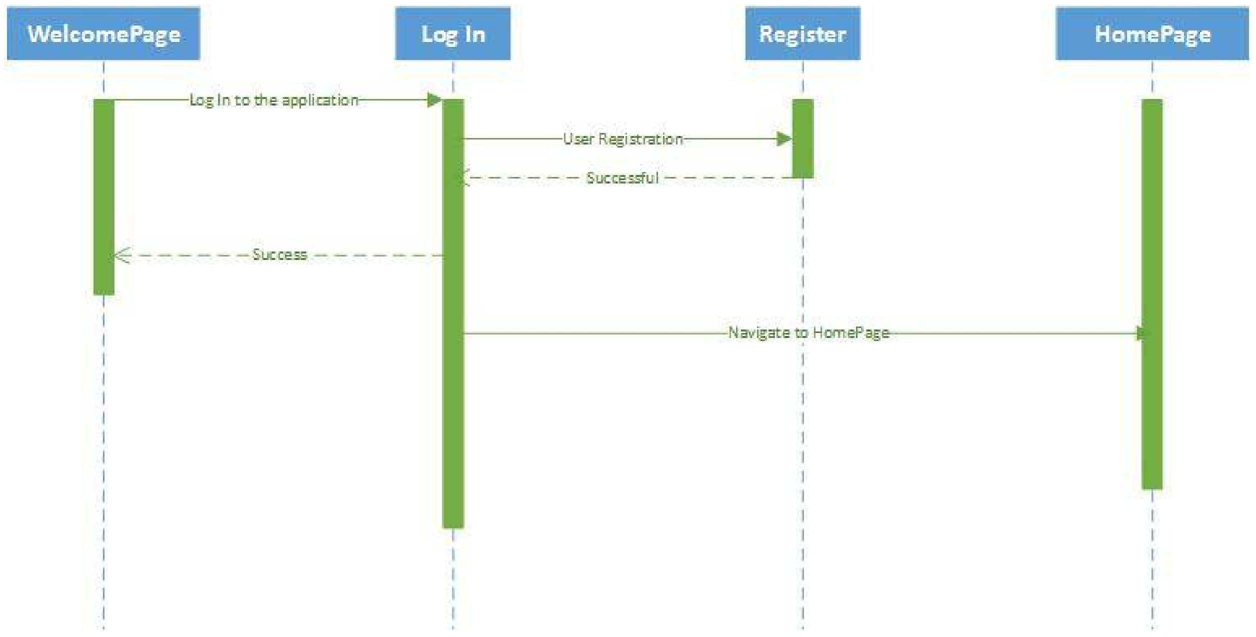
2. Login, Registration: Login using gmail and facebook or creating (registration) using e-mail and storing data in database.

3. Developer testing: Testing all scenarios from frontend UI

Service description:

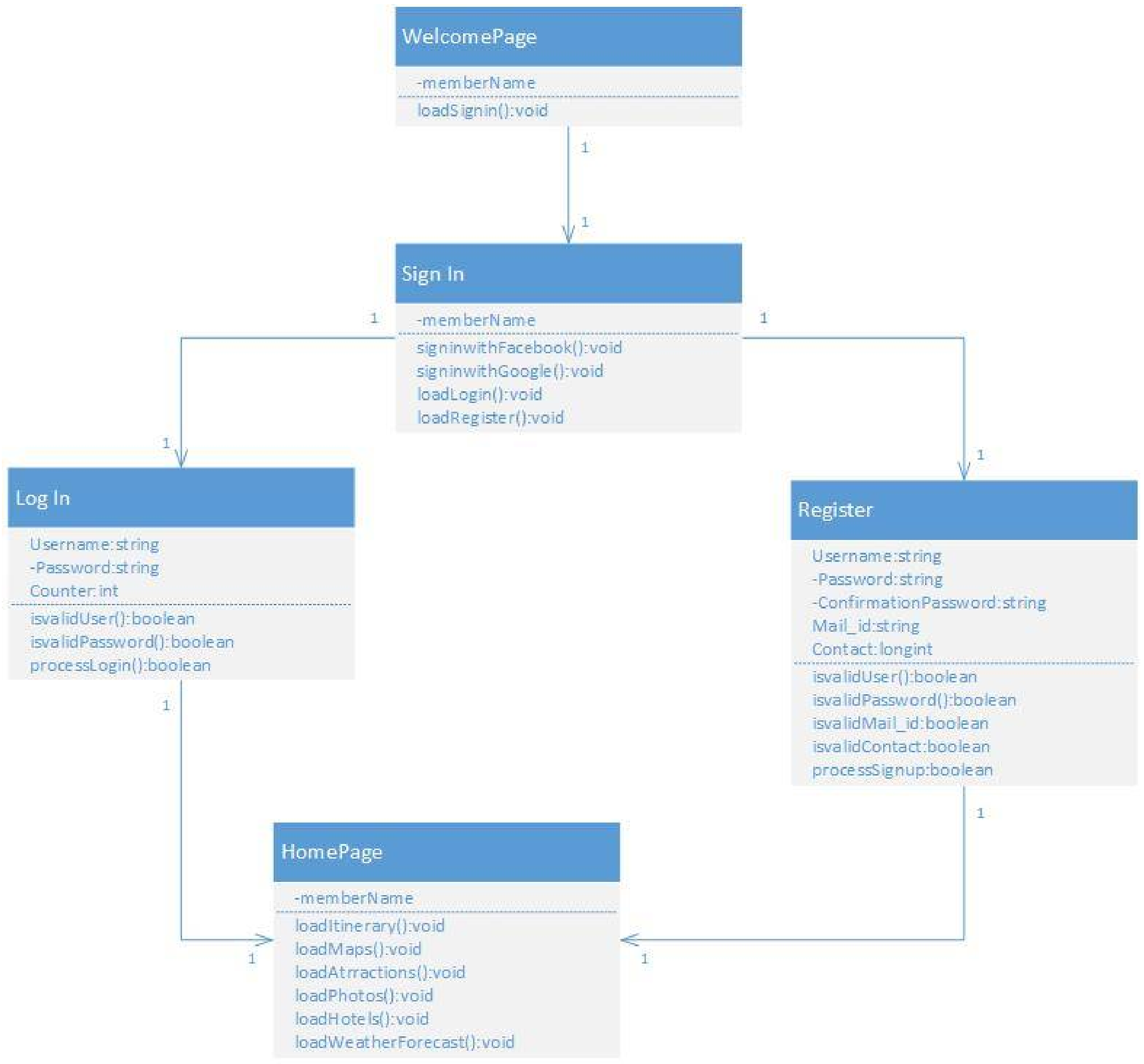
Login service is done using gmail REST API and Facebook API, these API return response in JSON Format for data exchange between application and service. Database server stores the information of the user like, firstname, lastname, e-mail and password.

Sequence diagram:



The above shown Diagram represents the sequence diagram for logging into the application. The Welcome page is the main page. From this the user logins and then the user registers if he is not an existing user and from login page if user logins, the application navigates to the homepage. This scenario is for the first increment only.

Class diagram:



The above shown class diagram is the design for this increment. It includes Classes such as Welcome Page, Sign In, Log In, Register and Home Page. Welcome Page is the main page for this project and the operation of this class is to go the Sign in page. This Sign in class operations are Facebook login, google login, own login and registration. The attributes of login page are username and password and its operation is to go the welcome page and before doing that validations for the details entered are done. The Register class is similar to that of log in class except its attributes. Home Page class has several operations such as loading the main features of the application and this class can be discussed in the next increment.

Design of Mobile Client Interface:

The Mobile Client Interface consists of different screens for the first increment of the project. They are described below

1. Welcome Page – The user opens the application and interacts with the welcome page which consists of the sign in option.
2. Sign in page- The user will have different options in this page that is different ways to sign in.
3. Log in page- the user logins giving his information like username and password.
4. Register page- user registers through the application by giving his information asked.

Implementation :

Implementation of REST services:

1. Google REST API:

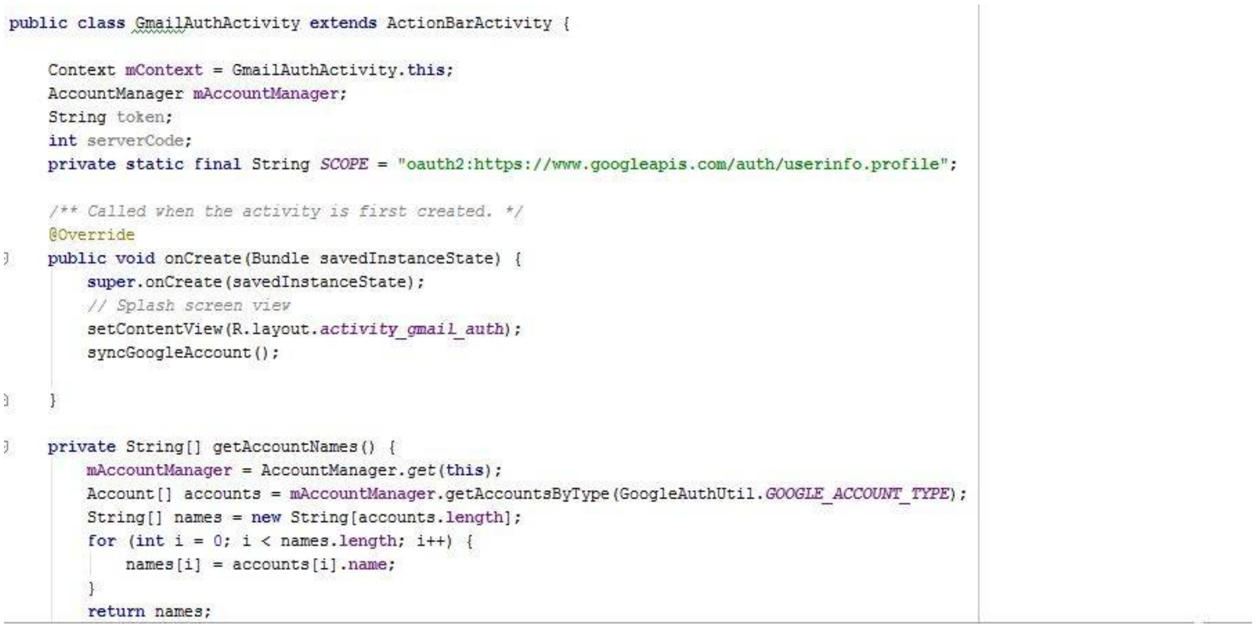
Implemented Google REST APIs service to authenticate the user to login into application using Gmail. Below is the code used for integrating this service.

```
"oauth2:https://www.googleapis.com/auth/userinfo.profile"

"https://www.googleapis.com/oauth2/v1/userinfo?access_token="+ token
```

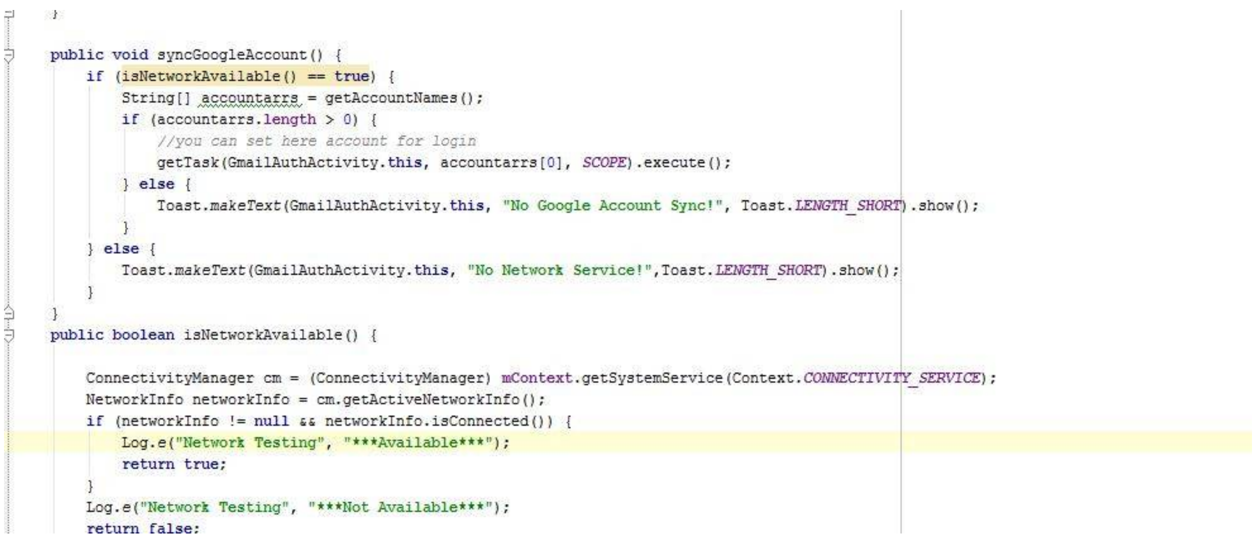
Getting the token from mobile gmail account and passing it to Google REST API service to get below Json object.

```
{
 "id": "xx",
 "name": "xx",
 "given_name": "xx",
 "family_name": "xx",
 "link": "xx",
 "picture": "xx",
 "gender": "xx",
 "locale": "xx"
}
```

From there displaying the details in UserDetailsActivity page.



```java
public class GmailAuthActivity extends ActionBarActivity {

    Context mContext = GmailAuthActivity.this;
    AccountManager mAccountManager;
    String token;
    int serverCode;
    private static final String SCOPE = "oauth2:https://www.googleapis.com/auth/userinfo.profile";

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Splash screen view
        setContentView(R.layout.activity_gmail_auth);
        syncGoogleAccount();

    }

    private String[] getAccountNames() {
        mAccountManager = AccountManager.get(this);
        Account[] accounts = mAccountManager.getAccountsByType(GoogleAuthUtil.GOOGLE_ACCOUNT_TYPE);
        String[] names = new String[accounts.length];
        for (int i = 0; i < names.length; i++) {
            names[i] = accounts[i].name;
        }
        return names;
```

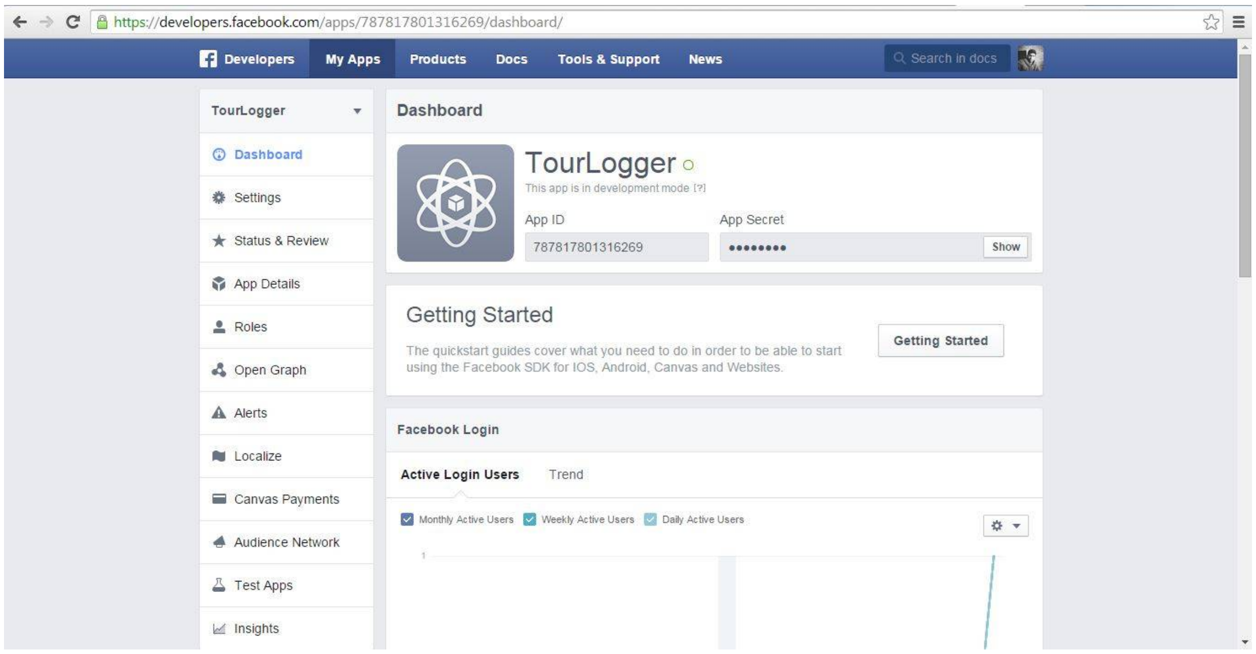


```java
    }

    public void syncGoogleAccount() {
        if (isNetworkAvailable() == true) {
            String[] accountarrs = getAccountNames();
            if (accountarrs.length > 0) {
                //you can set here account for login
                getTask(GmailAuthActivity.this, accountarrs[0], SCOPE).execute();
            } else {
                Toast.makeText(GmailAuthActivity.this, "No Google Account Sync!", Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(GmailAuthActivity.this, "No Network Service!",Toast.LENGTH_SHORT).show();
        }
    }
    public boolean isNetworkAvailable() {

        ConnectivityManager cm = (ConnectivityManager) mContext.getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = cm.getActiveNetworkInfo();
        if (networkInfo != null && networkInfo.isConnected()) {
            Log.e("Network Testing", "***Available***");
            return true;
        }
        Log.e("Network Testing", "***Not Available***");
        return false;
```

2. Facebook API:

Created facebook app and installed facebook dependencies in android studio. Getting facebook login button from facebook api.

Below screenshots contain app created for this project.



https://developers.facebook.com/apps/787817801316269/dashboard/

Dashboard

TourLogger

This app is in development mode [?]

App ID: 787817801316269

App Secret: ••••••••

Getting Started

The quickstart guides cover what you need to do in order to be able to start using the Facebook SDK for IOS, Android, Canvas and Websites.

In TourLogger project it was implemented by adding below code in .

```
<meta-data android:name="com.facebook.sdk.ApplicationId" android:value="@string/facebook_app_id" />
```

The facebook_app_id needs to be added in values/strings.xml

```
<string name="facebook_app_id">787817801316269</string>
```

To get facebook button need to add below code snippet in activity.xml

```
<com.facebook.widget.LoginButton
    android:id="@+id/authButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="43dp" />
```

Testing:

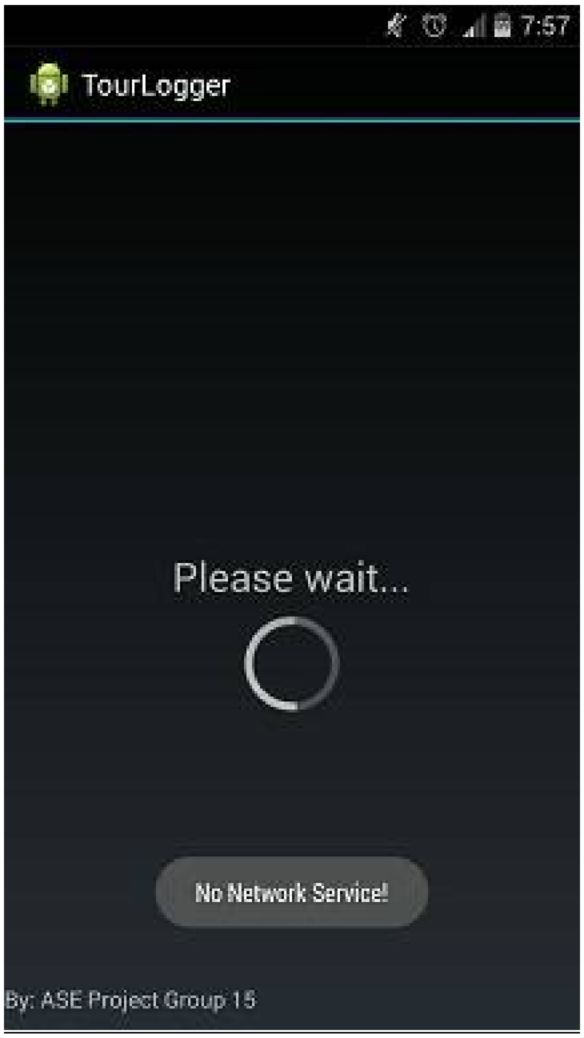
Perform Unit testing :

Manual Test Case : If there is no network connection then while authenticating login with gmail it should show "No Network Service".

Expected: Screen with no network service.

Result: **Passed**

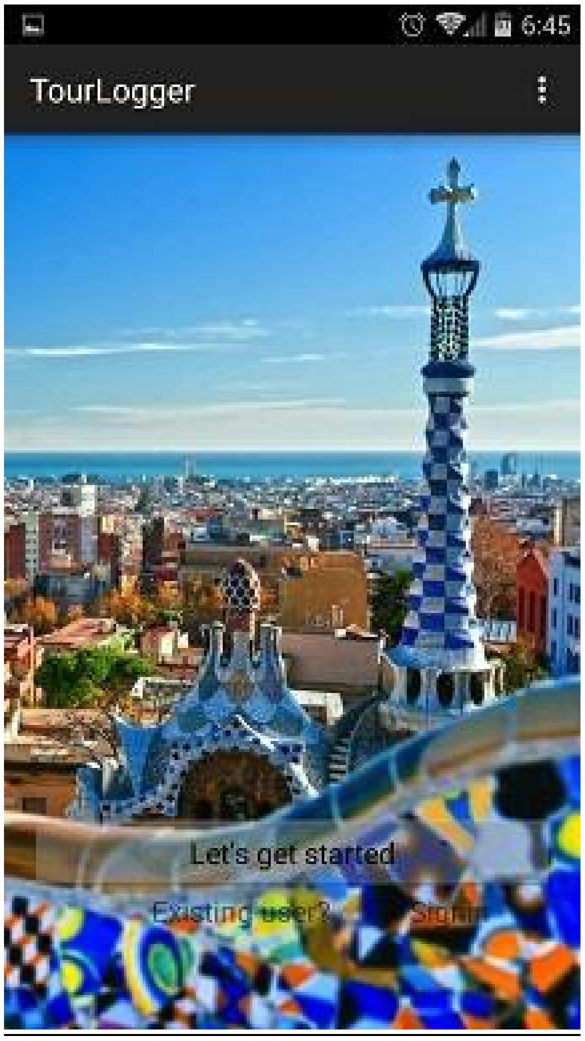
Screenshot:


7:57
TourLogger
Please wait...
No Network Service!
By: ASE Project Group 15

Report:

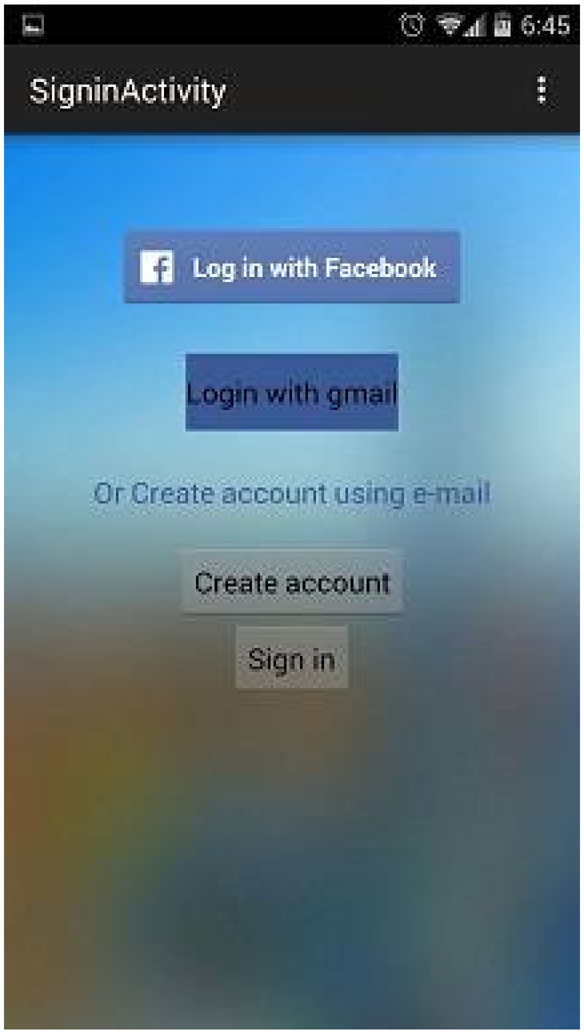
Screenshots and explanation on the design, implementation and testing.

1.



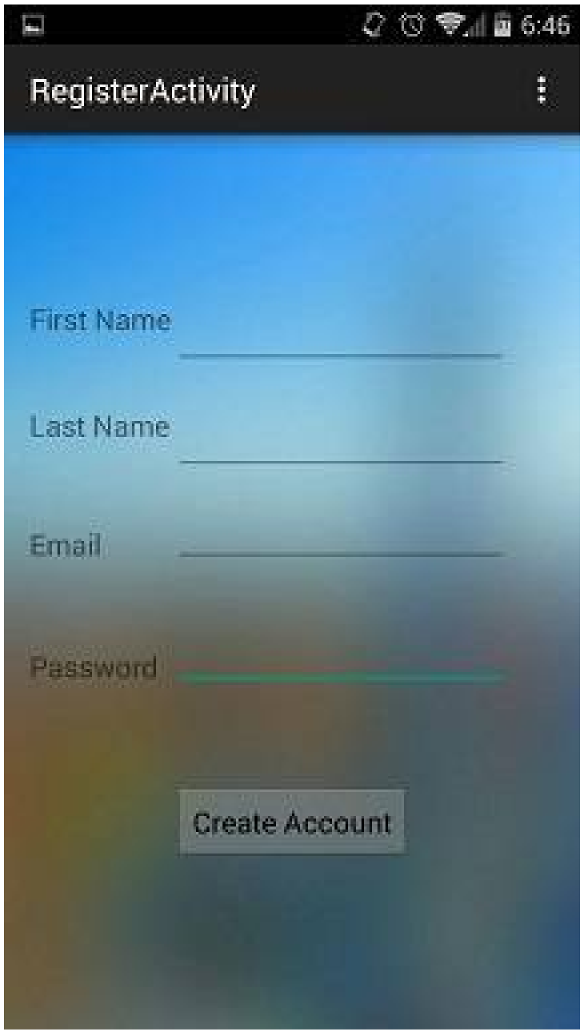
This is the design of the Welcome Page which is the first page of this application. It has one button Let’s get started and below that, it has some text Existing user? Which means is the user is already registered, he can go to Sign in page to sign in. The background image is the main part because it plays a crucial role in the design. This implementation is done by the help of android widgets such as the button and the text view. Testing of this page is done manually by deploying the code into the android phone.

2.



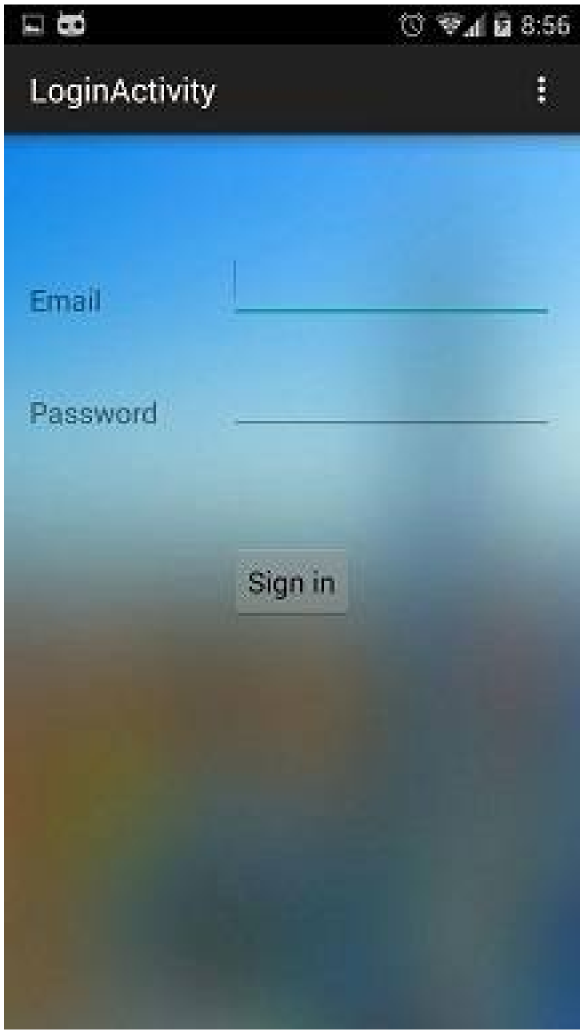
This is the second page which is a sign in page. It provides various features like Sign in button, create account button, log in with facebook button and log in with gmail button and some text in between like Or Create account using e-mail. The implementation of log in with facebook is done using the facebook api and facebook sdk. Integration of facebook login is done in this application and also the gmail log in integration is done. Testing is done whether the integration process working properly or not manually.

3.



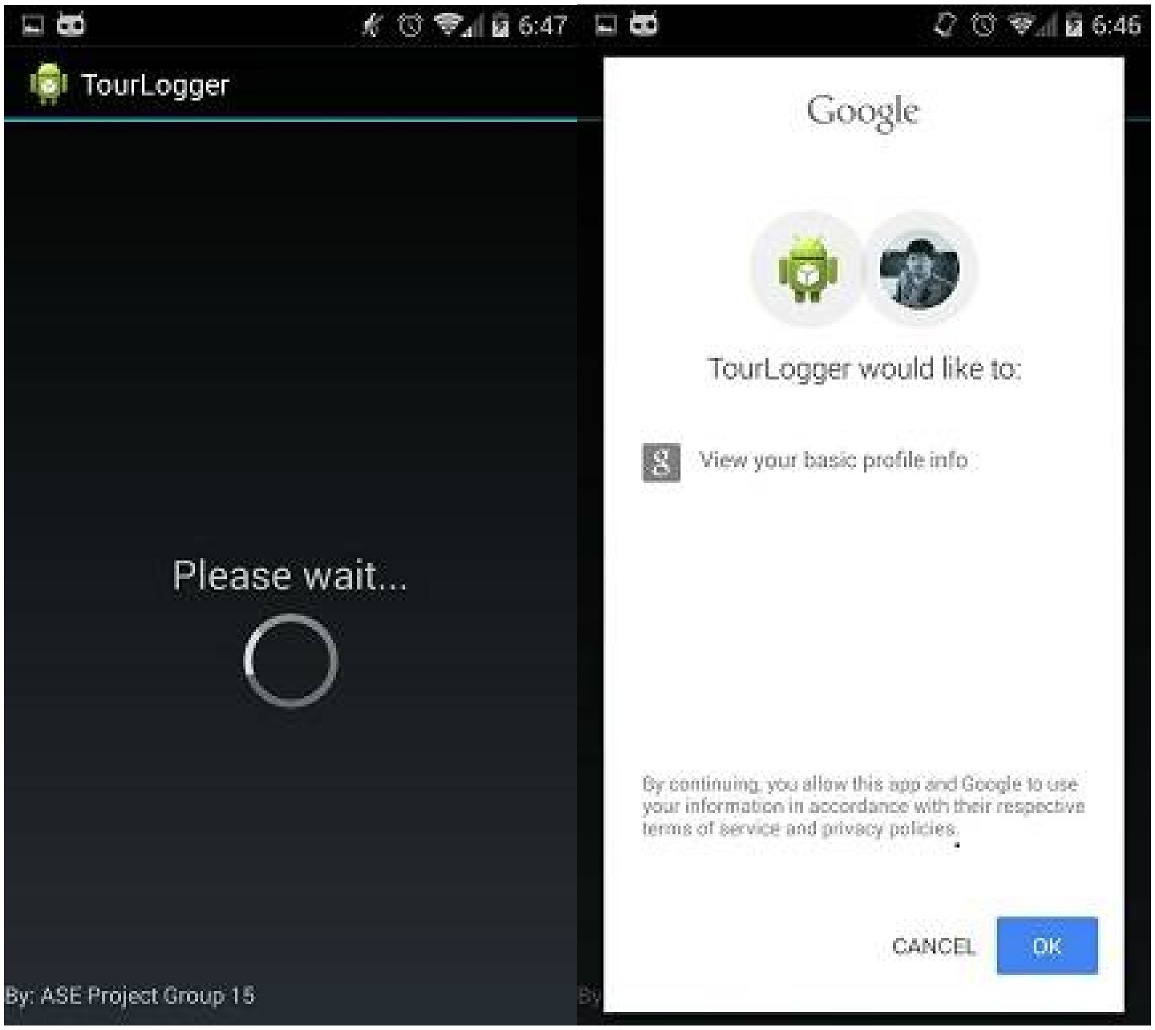
This page is the Register activity page or create account page, when users click on the create account button in the previous page, the application navigates to this page. This page contains several fields such as first name, last name, email, password etc. which are text fields and when a user fills the data and click on the create account page, the registration will be successful. This is implemented by using the android widgets.

4.



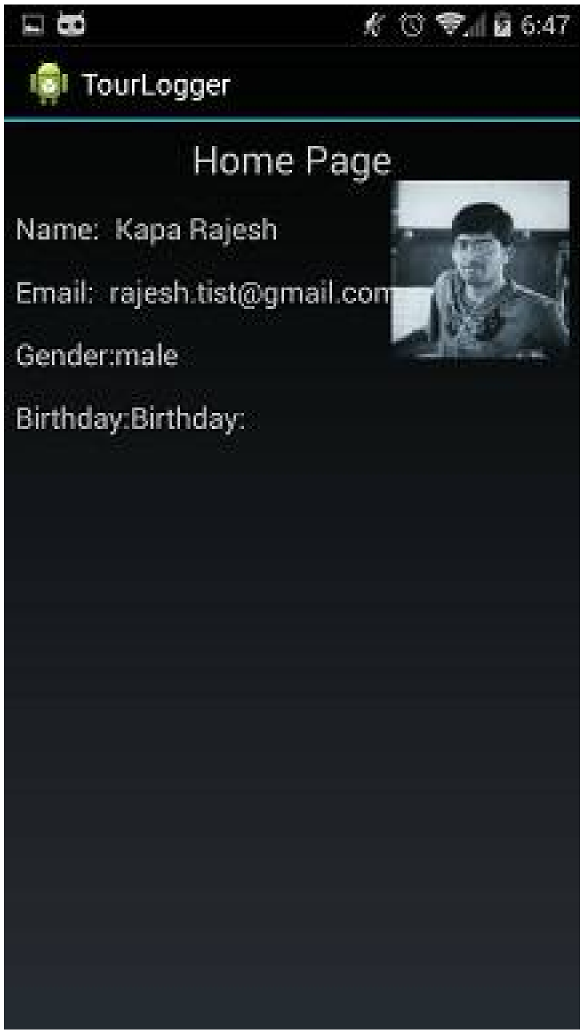
This page is the Login activity page or Sign in page, when users click on the sign in button in the previous page 2, the application navigates to this page. This page contains several fields such as email, password etc. which are text fields and when a user fills the data and click on the sign in page, and the application navigates to the home page. This is implemented by using the android widgets such as buttons and text fields.

5.


6:47
TourLogger
Please wait...
By: ASE Project Group 15
6:46
Google
TourLogger would like to:
View your basic profile info
By continuing, you allow this app and Google to use your information in accordance with their respective terms of service and privacy policies.
CANCEL
OK

So, when the user clicks on the log in with gmail button in the page 2, he will be asked to accept the permissions for the application in order to login with his gmail account. The implementation of this is done using the google api and the source code for that is shown above. If the user clicks ok, then the details of the user can be shown and these details can be used in the application in the next increments. Presently the information is displayed in the application as shown in the below figure.

6.


6:47

TourLogger

Home Page

Name: Kapa Rajesh

Email: rajesh.tist@gmail.com

Gender:male

Birthday:Birthday:

This page shows the information of the user which is present in his gmail account.

Deployment:

ScrumDo link: https://www.scrumdo.com/projects/project/asetourlogger/iteration/121729/board

GitHub link: https://github.com/rajeshkapa/ASETourLogger

Project Management:

Implementation status report:

Work completed:

Description – The work assigned for the increment 1 is completed. That is understanding the user requirements, designing the User interface, Login page, Register page, integration of the gmail login and facebook login using the api's and displaying the user information from his gmail account, a user's table is created in the database with the attributes like first name, last name, email and password and these values are stored in the database and finally testing the pages described above is done.

Responsibility (Task, Person) - Kapa Rajesh (login and registration pages, database and testing), Gummadi Raghu Vital (user requirements, login and registration pages and designing user interface).

Time taken (#hours) – Time taken for this work is approximately 20 hours.

Contributions (members/percentage) – Kapa Rajesh (50%), Gummadi Raghu Vital (50%)

Work to be completed:

None for the first increment.

Issues/Concerns:

- Build issues while integrating google play service.
- Integration of facebook app id issue in application.
- Deployment of code in cloud server issue.