Embedded System Programming [ECE 558]
PROJECT 3

GoogleMap Design Report

(Version 1.0)

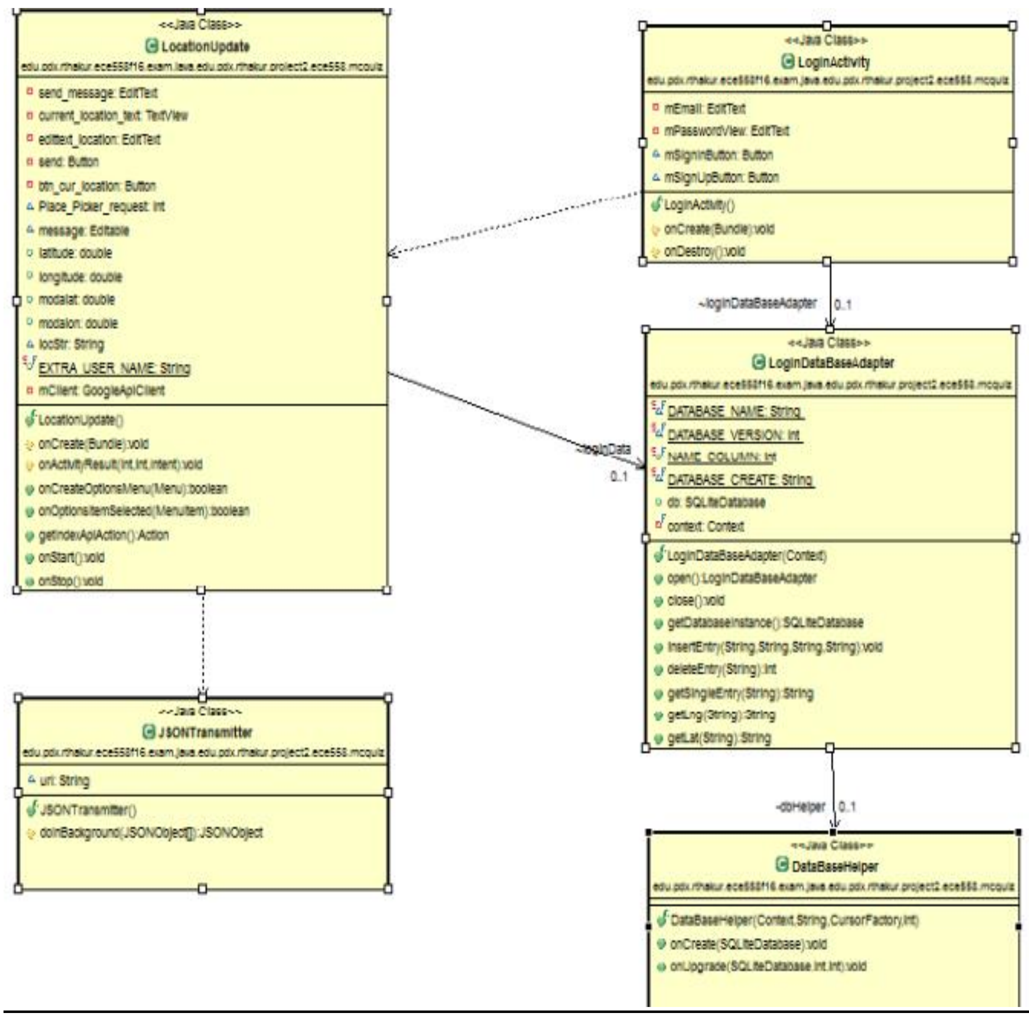
BY:
Surendra Maddula(surendra@pdx.edu)

INTRODUCTION:

This project implements a Google map app where the user can sign up by creating an account with a username and password and setting up the home location facility has also been provided and then with a valid username and password the user can sign in to the map where the place picker can be used to hover over the places and select the location which will be updated as the current location. Also an additional feature of computing time to reach home for a coffee is computed. So by calculating the time to reach home will be displayed showing when your coffee will be ready.

UML CLASS DIAGRAM:

The UML class diagram is shown here. A complete .png file of the UML class diagram is submitted along with the project as well.


<<Java Class>>
LocationUpdate
edu.pdx.rthakur.ece558f16.exam.java.edu.pdx.rthakur.project2.ece558.mcquiz
send_message: EditText
current_location_text: TextView
edittext_location: EditText
send: Button
btn_cur_location: Button
Place_Picker_request: int
message: Editable
latitude: double
longitude: double
modalat: double
modalon: double
locStr: String
EXTRA_USER_NAME: String
mClient: GoogleApiClient
LocationUpdate()
onCreate(Bundle):void
onActivityResult(int,int,Intent):void
onCreateOptionsMenu(Menu):boolean
onOptionsItemSelected(MenuItem):boolean
getIndexApiAction():Action
onStart():void
onStop():void
<<Java Class>>
LoginActivity
edu.pdx.rthakur.ece558f16.exam.java.edu.pdx.rthakur.project2.ece558.mcquiz
mEmail: EditText
mPasswordView: EditText
mSignInButton: Button
mSignUpButton: Button
LoginActivity()
onCreate(Bundle):void
onDestroy():void
~loginDataBaseAdapter
0..1
~loginData
0..1
<<Java Class>>
LoginDataBaseAdapter
edu.pdx.rthakur.ece558f16.exam.java.edu.pdx.rthakur.project2.ece558.mcquiz
DATABASE_NAME: String
DATABASE_VERSION: int
NAME_COLUMN: int
DATABASE_CREATE: String
db: SQLiteDatabase
context: Context
LoginDataBaseAdapter(Context)
open():LoginDataBaseAdapter
close():void
getDatabaseInstance():SQLiteDatabase
insertEntry(String,String,String,String):void
deleteEntry(String):int
getSingleEntry(String):String
getLng(String):String
getLat(String):String
-dbHelper
0..1
<<Java Class>>
JSONTransmitter
edu.pdx.rthakur.ece558f16.exam.java.edu.pdx.rthakur.project2.ece558.mcquiz
url: String
JSONTransmitter()
doInBackground(JSONObject[]):JSONObject
<<Java Class>>
DataBaseHelper
edu.pdx.rthakur.ece558f16.exam.java.edu.pdx.rthakur.project2.ece558.mcquiz
DataBaseHelper(Context,String,CursorFactory,int)
onCreate(SQLiteDatabase):void
onUpgrade(SQLiteDatabase,int,int):void

TECHNICAL DESIGN:

The design implementation includes multiple activities:

1) LoginActivity.java – Contains the starter code for the Login activity. This is the launcher activity for the app.
2) SignUpActivity.java – Contains the starter code for registering new users and adding them to the user information database.
3) LocationUpdate.java – Contains the code for getting a new location and for sending/receiving a text string to the remote webserver.
4) LoginDataBaseAdapter.java – Contains the code for accessing the Login (user information) database.
5) JSONTransmitter.java – Contains the code for handling the JSON transactions between the app and the web server.
6) DataBaseHelper.java – Contains the Helper class for creating and upgrading the user information (Login) database.

HIGHLIGHTED FEATURES:

The highlighted features implemented in this app includes the following:

1) This app uses SQLite database in which all the informations provided by the user is saved and is fetched next time the user logs in. Altering the table to add two new columns of latitude and longitude respectively are done in DataBaseHelper.java.

```
//changed the version of the database so that we can alter the table and
add latitude and longitude feilds.
 if(_oldVersion==1){
     _db.execSQL("ALTER TABLE LOGIN ADD COLUMN LONGITUDE text");
     _db.execSQL("ALTER TABLE LOGIN ADD COLUMN LATITUDE text");
 }
```

2) The user can click on the sign up button where the user needs to sign up by entering username and password, confirming the password and setting up the home location (longitude and latitude). These values will be saved in the database so that the user can log in with the same username and password next time he opens up the app.

```
      // check if any of the fields are vacant
if(userName.equals("")||password.equals("")||confirmPassword.equals("")||lat.eq
uals("")||lng.equals(""))
        { Toast.makeText(getApplicationContext(), "Field Vaccant",
Toast.LENGTH_LONG).show();
            return;
        }
        // check if both password matches
        if(!password.equals(confirmPassword))
        {   Toast.makeText(getApplicationContext(), "Password does not match",
Toast.LENGTH_LONG).show();
            return;
        }
        else
        {// Save the Data in Database
            loginDataBaseAdapter.insertEntry(userName, password,lat,lng);
            Toast.makeText(getApplicationContext(), "Account Successfully
Created ", Toast.LENGTH_LONG).show();
        }

    }
});}
```

3) The user can sign in by entering the username and password he used while creating the account and can go to the map activity where the user can select the current location (LocationUpdate Activity) on successfully logging in.

```
if(password.equals(database_password)){
    Toast.makeText(getApplicationContext(), "Successfully Logged In ",
Toast.LENGTH_LONG).show();
    Intent i_Locationupdate = new
android.content.Intent(getApplicationContext(), LocationUpdate.class);
    i_Locationupdate.putExtra(LocationUpdate.EXTRA_USER_NAME,userName);
    startActivity(i_Locationupdate);
}
```

4) The Place picker widget used to get the current location is started by calling the PlacePicker's IntentBuilder.

```
//object of PlacePicker's Intentbuilder to add the PlacePicker
functionality
        PlacePicker.IntentBuilder builder = new
PlacePicker.IntentBuilder();
        try {
            Intent intent = builder.build(LocationUpdate.this);
            startActivityForResult(intent,Place_Picker_request);
        } catch (GooglePlayServicesRepairableException e) {
            e.printStackTrace();
        } catch (GooglePlayServicesNotAvailableException e) {
            e.printStackTrace();
        }

    }
});
```

5) The username, password, latitude and longitude from the database are fetched by using getSIngleEntry() for username and password and getLan() and getLng() methods for latitude and longitude implemented on the LoginDataBaseAdapter.java.

```
//fetching longitude from database using getLng()
public String getLng(String userName)
{
    Cursor cursor=db.query("LOGIN", null, " USERNAME=?", new
String[]{userName}, null, null, null);
    if(cursor.getCount()<1) // UserName Not Exist
    {
        cursor.close();
        return "NOT EXIST";
    }
    cursor.moveToFirst();
    String longitude=
cursor.getString(cursor.getColumnIndex("LONGITUDE"));
    cursor.close();
```

```
           return longitude;
       }

       //fetching Latitude from database using getLat()
public String getLat(String userName)
{
    Cursor cursor=db.query("LOGIN", null, " USERNAME=?", new
String[]{userName}, null, null, null);
    if(cursor.getCount()<1) // UserName Not Exist
    {
        cursor.close();
        return "NOT EXIST";
    }
    cursor.moveToFirst();
    String latitude= cursor.getString(cursor.getColumnIndex("LATITUDE"));
    cursor.close();
    return latitude;
}
```

6) The places nearby to the current location can also be used as the current location by selecting on that desired place.
7) This current location is used to calculate the time to make coffee if the user sends I need Coffee message to the server.
8) The time is basically computed using the current and home location.

ISSUES & FIXES:

1) ISSUE: The app was not running as required on the mobile phones having different versions.
   FIXES: Reading different blogs and StackOverflow; it was the problem of google maps PlacePicker and hence it behaves differently.
2) ISSUE: The app was not updating the home location on clicking the set home button.
   FIX: This was solved by turning on the location services settings on the mobile devices.
3) ISSUE: Widgets positioning in the GUI.
   FIX: By using the margin attributes in the xml file the widgets positions were set based on trial method.
4) Many times I faced the difficulty running my app as it would mark the R as red everywhere in java files which led me to restart the android studio again and again and proceed.

REFRENCES:

1) https://developer.android.com
2) http://tools.android.com/tech-docs/new-build-system/user-guide
3) http://stackoverflow.com
4) http://www.androidwarriors.com/2016/02/android-sqlite-database-tutorial-sqlite.html
5) http://www.androidhive.info/2011/11/android-sqlite-database-tutorial/

6) Project 3 document and release notes by Professor Roy Kravitz , Hiral Barot and Dheeraj Chand V.
7) ECE 558 fall 2016 class notes and in class exercises by Professor Roy Kravitz.