

Log In Application

Homework 1

1. [Overview](#)
2. [Specifications](#)
3. [Perquisites](#)
4. [Documentation](#)
 - a. [Creating an account](#)
 - b. [Sign in](#)
 - c. [Manage Profile](#)
 - d. [Known Bugs](#)
5. [Tests](#)
 - a. [Test 1-Sign Up](#)
 - b. [Test 2- Sign In](#)
 - c. [Test 3 - Profile](#)
6. [Functions](#)
 - a. [Sign In](#)
 - b. [Sign Up](#)
 - c. [Profile](#)
7. [Appendix 1](#)

1. Overview

The purpose of this assignment was to build a simple *Log in* application for mobile devices. The two major platforms on the market, nowadays, are Android and iOS.

Android, is the world's most popular mobile platform, powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast—every day another million users power up their Android devices for the first time and start looking for apps, games, and other digital content. That's why we chose **Android SDK** to develop our application.

The application was built using the **Eclipse IDE**, with the **ADT plug-in** for **Android SDK**.

2. Specifications

- a. The application will display a Log In interface
- b. The user will be able to log in, or create an account
- c. As an authenticated user, one can create and modify the profile, with the data being saved in a database
- d. An authenticated user can change the password

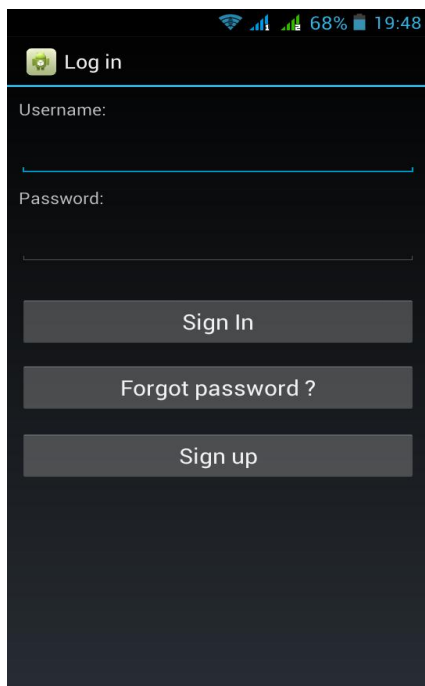
3. Perquisites

- a. A device with at least Android 2.3.1 (Froyo)
- b. The device must have an internet connection
- c. The application is target for the Android 4.0.4, API Level 15
- d. 1 MB of free space

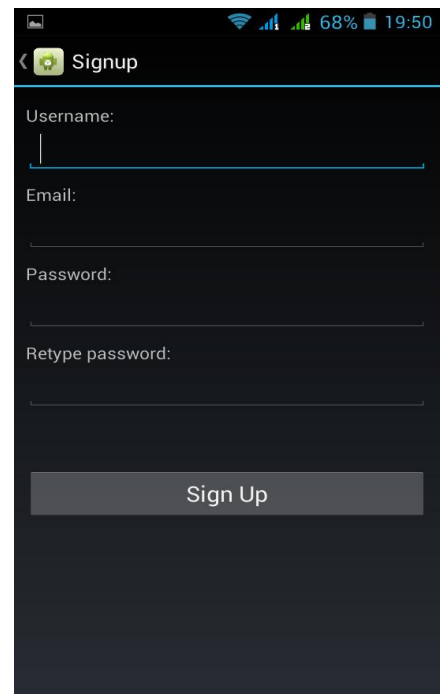
4. Documentation

a. Creating an account

Launch the application on your phone and choose **Sign Up** from the menu (Fig1). A new activity will appear where you can fill your information (Fig2). Press **Sign Up** again to create the account. If everything was right, a message will be prompted and the application will return to the **Log In** activity (Fig1).



(Fig 1)



(Fig 2)

b. Sign In

If you already have an account, fill in the username and the password in the **Log In** activity (Fig 1).

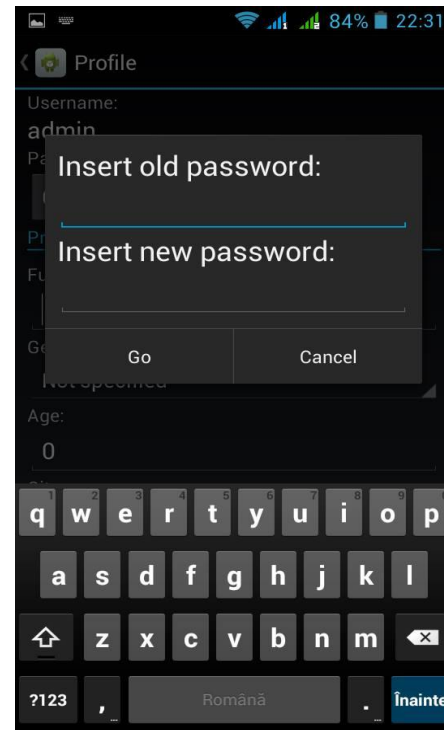
c. Manage Profile

After successfully signing into the application you will be redirected to your **Profile** (Fig 3). If you have a new account, most of the fields will be empty. You can complete your profile by filling into the text views for each field, or by choosing from the list boxes. Then, hit the **Save** button at the end in order to

save your data. **Caution:** The updates will not be saved after you close the application without pressing **Save** button.



(Fig 3)



(Fig 4)

In order to change, your password hit the **Change Password** button. A dialog will appear, as in (Fig 4). Introduce your old password and your new password and hit OK. It is not required to hit **Save** after this operation.

5. Tests

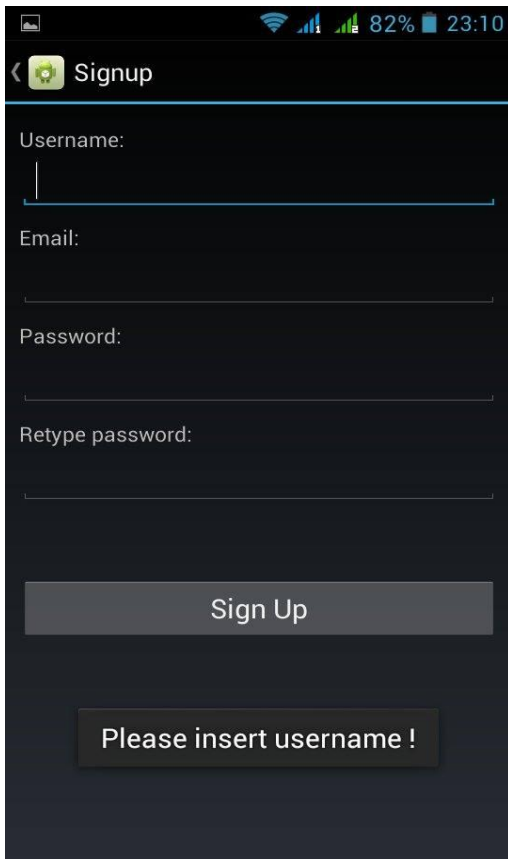
a. Test 1 – Sign up

A number of constraints have been applied to ensure the data validation.

If the user tries to register with an empty field user name, password or email, a message will be prompted and the account won't be created (Fig 5).

To avoid any mistyping error in the password field, it has to be introduced twice. If there is any difference, the account won't be created and a message will be prompted (Fig 6).

To ensure, the singularity of the username, the database is checked before. If there is another account with the same username, the account won't be created (Fig 7).



Signup

Username:

Email:

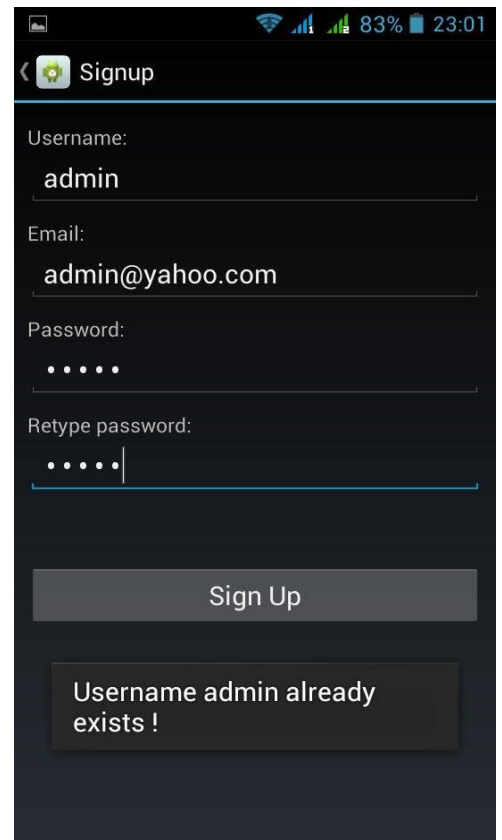
Password:

Retype password:

Sign Up

Please insert username !

(Fig 5)



Signup

Username:

admin

Email:

admin@yahoo.com

Password:

.....

Retype password:

.....

Sign Up

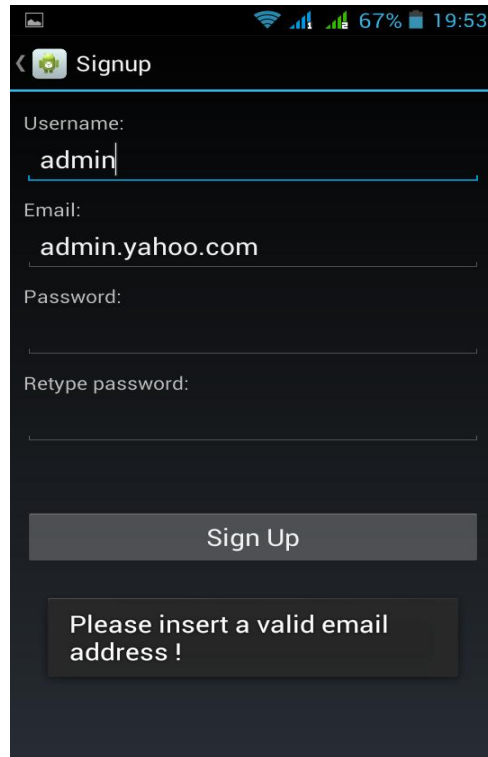
Username admin already exists !

(Fig 6)

b. Test 2 – Sign in

If you try to log in with an empty field, password or username, a message will be displayed.

If the username or the password doesn't match any recording in the database, then a message like in (Fig 8) will be displayed.



(Fig 8)

c. Test 3- Profile

The user can insert the into the Profile view, and save by hitting the save button. No email validation has been implemented yet. The password can be changed from the Dialog in (Fig 4). The old password is checked to correspond to the one in the database.

d. Known Bugs

- i. No data will be saved if suddenly the internet connection fails, no data saved locally
- ii. No email validation in the Profile activity(not implemented yet)
- iii. Forget Password button is not implemented yet

6. Functions

a. Sign in Activity(LoginActivity.java) - Functions

void onCreate() method :

- is called when the application is launched. The user interface is built here based on the corresponding XML file
- the connection with the parse.com database is created here
- controls are connected to the widgets in the interface

void signin() method:

- called by clicking on the **Sign In** button
- checks the data inserted and looks for the user in the database
- if the data is valid launches the Profile Activity sending the username via an Intent

void signup() method:

- called by clicking on the **Sign Up** button
- opens the Sign Up Activity

See full code with comments in Appendix 1.

b. Sign Up Activity(SignupActivity.java) - Functions

void onCreate() method :

- is called when the application is launched. The user interface is built here based on the corresponding XML file
- the connection with the parse.com database is created here
- controls are connected to the widgets in the interface

void signup() method:

- called by clicking on the **Sign Up** button
- validates the data inserted into the fields
- performs regular expression validation on the email
- checks for the existence of the introduced username
- creates a new account and saves it in the database

See full code with comments in Appendix 1.

c. Profile Activity(ProfileActivity.java) - Functions

void onCreate() method :

- is called when the application is launched. The user interface is built here based on the corresponding XML file
- the connection with the parse.com database is created here
- gets the username sent by the **Sign In** activity via intent
- queries the database for the given username'
- controls are connected to the widgets in the interface
- updates the fields with the information from the database

void popDialog() method:

- called by clicking on the **Change Password** button
- opens an AlertDialog with two fields for old an new password
- queries the table for the old password to compare it to the inserted one
- updates the new password(no empty field allowed)

void saveData() method:

- called by clicking on the **Save** button
- updates the database for each field with the values inserted by the user
- return to Sign In acitivity

See full code with comments in Appendix 1.

7. Appendix 1

LoginActivity.java

```
package com.example.md_hw1;

import java.util.List;

import com.parse.FindCallback;
import com.parse.Parse;
import com.parse.ParseAnalytics;
import com.parse.ParseObject;
import com.parse.ParseQuery;
import com.parse.PushService;
import com.parse.ParseException;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class LoginActivity extends Activity {
    private EditText usernameEdit;
    private EditText passwordEdit;
    private final String USER_TABLE = "Profiles";
    public final static String EXTRA_MESSAGE =
"com.example.myfirstapp.MESSAGE";
    private String message;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        //The arguments are the APP_ID, and the CLient_ID from parse.com
        Parse.initialize(this, "udumMDhauwhzIXHbWK1Bm3twUIP63F4y2Eq1BGVO",
"FXKkoOICrczFFrYLMMvn6qRg3bB4rLCkzKdkS1qOU");
        ParseAnalytics.trackAppOpened(getIntent());
        //Get username and password editText from layout
        usernameEdit=(EditText)findViewById(R.id.usernameEdit);
        passwordEdit=(EditText)findViewById(R.id.passwordEdit);
    }
    public void signin(View view){
        //Fetch the username and password from the edittext
        String username=usernameEdit.getText().toString();
        String password=passwordEdit.getText().toString();
        if(username.length()==0)
            Toast.makeText(getApplicationContext(), "Please insert username
!", Toast.LENGTH_SHORT).show();
        else
            if(password.length()==0)
```

```

        Toast.makeText(getApplicationContext(), "Please insert password
!", Toast.LENGTH_SHORT).show();
    else
    {
        //Initiate a parse query
        ParseQuery<ParseObject> query = ParseQuery.getQuery(USER_TABLE);

        //Select where username is @username and password is @password
        query.whereEqualTo("username", username);
        query.whereEqualTo("password", password);

        //make the query
        try{
            List<ParseObject> userList=query.find();
            //if the username was found open Profile activity
            if(userList.size()>0)
            {

                message=username;
                Toast.makeText(getApplicationContext(), "Redirecting...",
Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(this, Profile.class);
                //send the user name via intent to be used in profile
acitivity

                intent.putExtra(EXTRA_MESSAGE, message);
                startActivity(intent);
            }
            else
            Toast.makeText(getApplicationContext(), "Invalid
username/password !", Toast.LENGTH_SHORT).show();

        }
        catch(ParseException e){
            Toast.makeText(getApplicationContext(), e.getMessage(),
Toast.LENGTH_LONG).show();
        }
    }
    passwordEdit.setText("");
}
public void signup(View view){
    //start the signup activity
    Intent intent = new Intent(this, Signup.class);
    this.startActivity(intent);
}
}

```

Signup.java

```

package com.example.md_hw1;

import java.util.ArrayList;
import java.util.List;

```

```
import com.parse.FindCallback;
import com.parse.Parse;
import com.parse.ParseAnalytics;
import com.parse.ParseException;
import com.parse.ParseObject;
import com.parse.ParseQuery;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Signup extends Activity {
    private EditText usernameEdit;
    private EditText passwordEdit;
    private EditText repasswordEdit;
    private EditText emailEdit;

    private final String USER_TABLE = "Profiles";
    private String username,password,email;
    //regular expression pattern for email validation
    private static final String EMAIL_PATTERN =
        "^[_A-Za-z0-9-\\+]+(\\.\\[_A-Za-z0-9-\\+]*@"
        + "[A-Za-z0-9-]+(\\.\\[_A-Za-z0-9-\\+]*"
        + "(\\.\\[_A-Za-z0-9-\\+]*){2,})$";

    private ParseObject users,contacts,profiles; //the object handles the
table users

    private Pattern pattern;
    private Matcher matcher;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);
        //Initialize PARSE
        //The arguments are the APP_ID, and the CLient_ID from parse.com
        Parse.initialize(this, "udumMDhauwhzIXHbWK1Bm3twUIP63F4y2Eq1BGVO",
"FXK0iOCrczFFrYLMMvn6qRg3bB4rLCkzKdkS1qOU");
        ParseAnalytics.trackAppOpened(getIntent());

        //Create a new ParseObject that will handle the users table
        users = new ParseObject(USER_TABLE);
        //Get username and password editText from layout
        usernameEdit=(EditText)findViewById(R.id.username);
        passwordEdit=(EditText)findViewById(R.id.password);
        repasswordEdit=(EditText)findViewById(R.id.repassword);
        emailEdit=(EditText)findViewById(R.id.email);

        //Regular expression email validator
        pattern = Pattern.compile(EMAIL_PATTERN);
```

```

    }

    public void signup(View view){
        //Fetch the username and password from the edittext
        username=usernameEdit.getText().toString();
        password=passwordEdit.getText().toString();
        email=emailEdit.getText().toString();
        if(username.length()==0)
            Toast.makeText(getApplicationContext(), "Please insert username
!", Toast.LENGTH_LONG).show();
        //if it DOES NOT validates the reg expression for email
        if(pattern.matcher(email).matches()==false)
            Toast.makeText(getApplicationContext(), "Please insert a valid
email address !", Toast.LENGTH_LONG).show();
        else
            if(password.length()==0)
                Toast.makeText(getApplicationContext(), "Please insert
password !", Toast.LENGTH_LONG).show();
            else
                if(!password.equals(repasswordEdit.getText().toString()))
                    Toast.makeText(getApplicationContext(), "Passwords are not
the same", Toast.LENGTH_LONG).show();
                else
                {
                    //Make a query to check wether the user already exists
                    ParseQuery<ParseObject> query = ParseQuery.getQuery(USER_TABLE);
                    //Select where username is @username
                    query.whereEqualTo("username",username);
                    query.findInBackground(new FindCallback<ParseObject>() {
                        public void done(List<ParseObject> userList, ParseException
e) {
                            if (e == null) {
                                if(userList.size()>0)
                                    Toast.makeText(getApplicationContext(), "Username
"+username+" already exists !", Toast.LENGTH_LONG).show();
                                else
                                    {
                                        Toast.makeText(getApplicationContext(), "Account
created !", Toast.LENGTH_SHORT).show();
                                        users.put("username", username);
                                        users.put("password", password);
                                        users.put("username",username);
                                        users.put("Email",email);
                                        users.saveInBackground();
                                        finish();
                                    }
                                } else {
                                    Toast.makeText(getApplicationContext(),
e.getMessage(), Toast.LENGTH_SHORT).show();
                                }
                            }
                        });
                }
            }
        }
    }
}

```

Profile.java

```
package com.example.md_hwl;

import java.util.List;

import com.parse.Parse;
import com.parse.ParseAnalytics;
import com.parse.ParseException;
import com.parse.ParseObject;
import com.parse.ParseQuery;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

public class Profile extends Activity {
    private TextView
usernameView,fullNameView,ageView,cityView,emailView,phoneView,passwordView;
    private Spinner genderView,languageView;
    private final String USERS_TABLE ="Profiles";
    private String ID,password;
    private ParseObject user;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_profile);
        //The arguments are the APP_ID, and the CLient_ID from parse.com
        Parse.initialize(this, "udumMDhauwhzIXHbWK1Bm3twUIP63F4y2Eq1BGVO",
"FXKoOICrczFFrYLMvn6qRg3bB4rLCkzKdkS1qOU");
        //Get textviews and spinners
        genderView = (Spinner) findViewById(R.id.gender);
        // Create an ArrayAdapter using the string array and a default
spinner layout
        ArrayAdapter<CharSequence> adapter =
ArrayAdapter.createFromResource(this,
            R.array.genders, android.R.layout.simple_spinner_item);
        // Specify the layout to use when the list of choices appears
```

```

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item
);
    // Apply the adapter to the spinner
    genderView.setAdapter(adapter);

    //Init language spinner
    languageView=(Spinner)findViewById(R.id.language);
    ArrayAdapter<CharSequence>
ladapter=ArrayAdapter.createFromResource(this,
        R.array.languages, android.R.layout.simple_spinner_item);
    // Specify the layout to use when the list of choices appears

ladapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_ite
m);
    // Apply the adapter to the spinner
    languageView.setAdapter(ladapter);

    usernameView=(TextView)findViewById(R.id.username);
    fullnameView=(TextView)findViewById(R.id.fullname);
    passwordView=(TextView)findViewById(R.id.password);
    ageView=(TextView)findViewById(R.id.age);
    cityView=(TextView)findViewById(R.id.city);
    emailView=(TextView)findViewById(R.id.email);
    phoneView=(TextView)findViewById(R.id.phone);
    // Get the message from the intent
    Intent intent = getIntent();
    String username = intent.getStringExtra(LoginActivity.EXTRA_MESSAGE);
    //Query the database for the rest of the information
    ParseQuery<ParseObject> query = ParseQuery.getQuery(USERS_TABLE);
    //Select where username is @username and password is @password
    query.whereEqualTo("username", username);
    try{
        List<ParseObject> userList=query.find();
        //Fetch the data from the database into the fields on the
acitivty
        ParseObject user=userList.get(0);
        ID=user.getObjectId();
        usernameView.setText(user.getString("username"));
        password=user.getString("password");
        fullnameView.setText(user.getString("fullname"));
        genderView.setSelection(user.getInt("Gender"));
        ageView.setText(Integer.toString(user.getInt("Age")));
        cityView.setText(user.getString("City"));
        languageView.setSelection(user.getInt("Language"));
        emailView.setText(user.getString("Email"));
        phoneView.setText(user.getString("Phone"));
    }
    catch(ParseException e){
        Toast.makeText(getApplicationContext(), e.getMessage(),
Toast.LENGTH_LONG).show();
    }

}

public void popDialog(View view){
    // get prompts.xml view

```

```

        user = ParseObject.createWithoutData(USERS_TABLE, ID);
        LayoutInflater li = LayoutInflater.from(this);
        View promptsView = li.inflate(R.layout.password, null);
        final AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(this);
        alertDialogBuilder.setView(promptsView);

        final EditText oldpass = (EditText) promptsView
            .findViewById(R.id.oldpass);
        final EditText newpass = (EditText) promptsView
            .findViewById(R.id.newpass);

        // set dialog message
        //add button and action listeners
        alertDialogBuilder
            .setCancelable(false)
            .setNegativeButton("Go",
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog,int id) {
                        //Check the old password to correspond to the inserted
one
                        //Make sure new password is not null
                        if(oldpass.getText().toString().equals(password) &&
newpass.getText().length()>0){
                            user.put("password",newpass.getText().toString());
                            user.saveInBackground();}
                        else
                            Toast.makeText(getApplicationContext(), "Invalid
old/new password", Toast.LENGTH_SHORT).show();

                    })

            .setPositiveButton("Cancel",
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog,int id) {
                        dialog.dismiss();
                    }
                }
            );

        // create alert dialog
        AlertDialog alertDialog = alertDialogBuilder.create();

        // show it
        alertDialog.show();
    }
    public void saveData(View view){
        user = ParseObject.createWithoutData(USERS_TABLE, ID);

        // Update database
        user.put("fullname", fullnameView.getText().toString());
        user.put("Gender", genderView.getSelectedItemPosition());
        user.put("Age", Integer.parseInt(ageView.getText().toString()));
        user.put("City", cityView.getText().toString());
        user.put("Language", languageView.getSelectedItemPosition());
    }

```

```
        user.put("Email",emailView.getText().toString());
        user.put("Phone",phoneView.getText().toString());
        // Save
        Toast.makeText(getApplicationContext(),"Saving...",
Toast.LENGTH_SHORT).show();
        try {
            user.save();
        } catch (ParseException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        finish();
    }
}
```