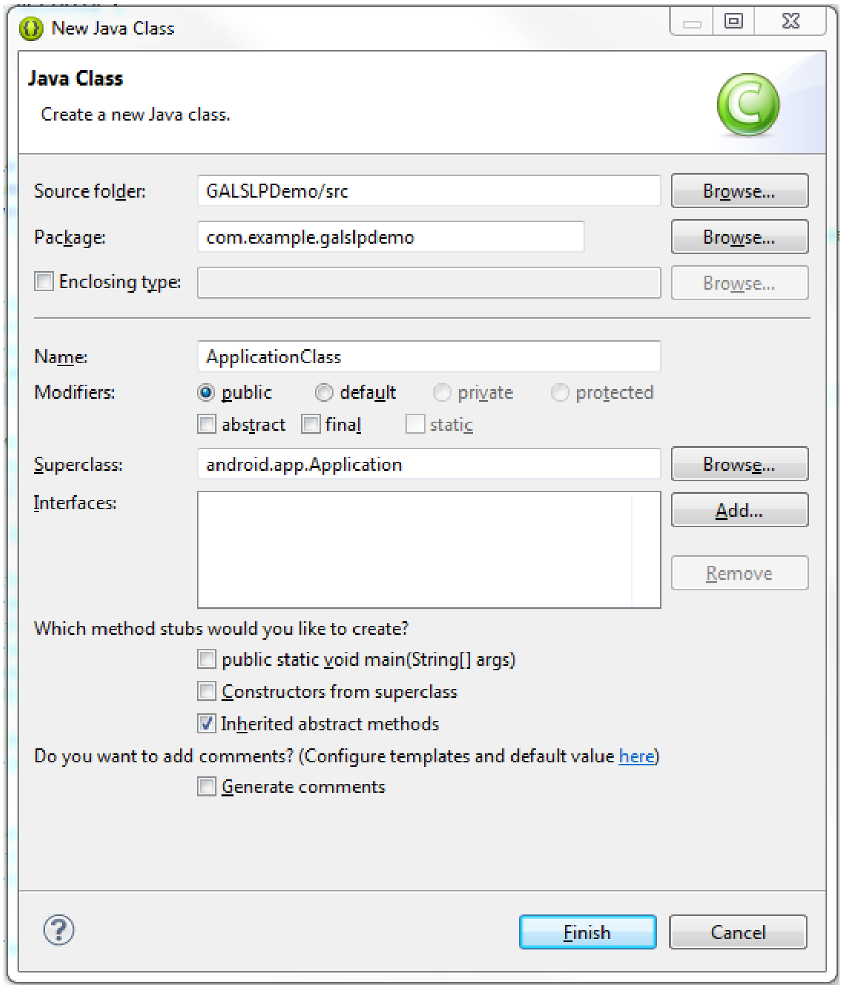
This document assumes you know the basics of Android. All of the code will posted online after the project is published and a link will be added to the blog post.

STEP 1: Making an Application Class

Like any good programmer out there, I don’t like repeating code. So to start off we’re making an application class. This class runs before anything else and stays active throughout the entirety of the app. This means that we do not need to initialize parse in every activity. We only need to initialize it in the application class.

If you do not need parse in too many of your classes I recommend to just initializing as you go and you can just skip over to the next step of this tutorial.

1. Add a new java class to your main with the superclass android.app.Application



New Java Class

Java Class

Create a new Java class.

Source folder: GALSLPDemo/src

Package: com.example.galslpdemo

□ Enclosing type:

Name: ApplicationClass

Modifiers: ● public ○ default ○ private ○ protected

□ abstract □ final □ static

Superclass: android.app.Application

Interfaces:

Which method stubs would you like to create?

□ public static void main(String[] args)

□ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

□ Generate comments

2. Within the application class add these lines and of course import parse at the top

```
public void onCreate() {
	Parse.initialize(this, APPLICATION_ID, CLIENT_KEY
}
```

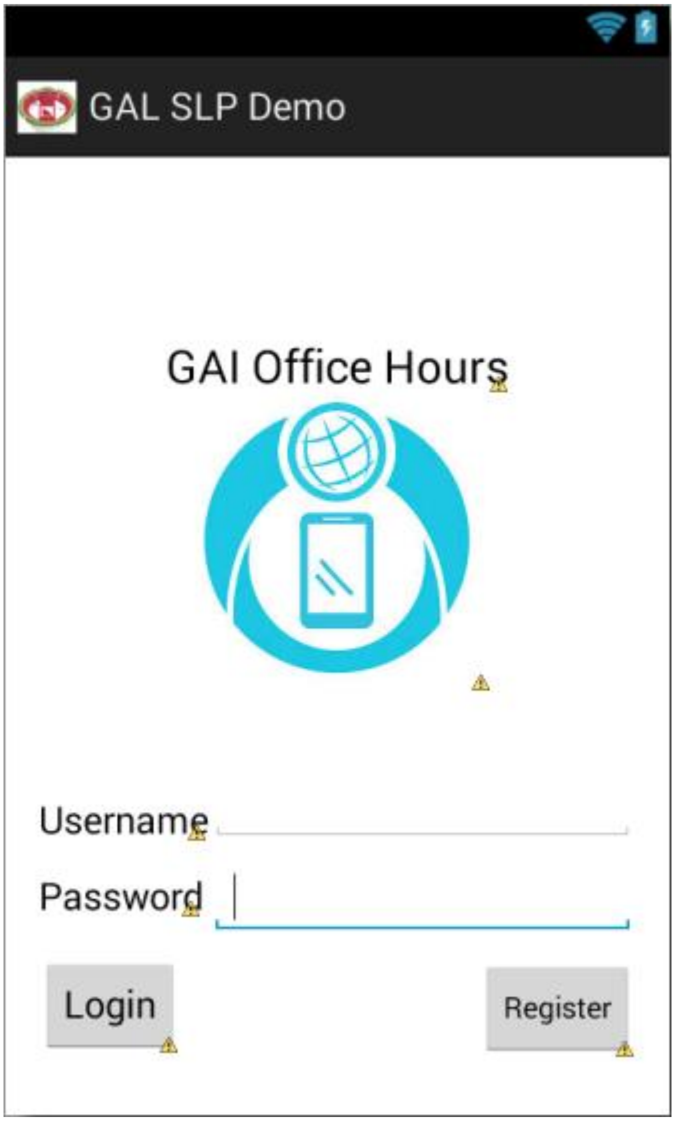
3. You then need to add the application class to your manifest file in the application heading.

```
android:name="com.example.galslpdemo.ApplicationClass"
```

STEP 2: Login Screen

We're making the Login Screen. I'm not too concerned with it looking good but it at least needs 1 name field, 1 password field, a login button, and a register button. This thing also will probably mess up when in a different screen size but whatever.

1. Make your layout. Do whatever you want. This is what mine looks like



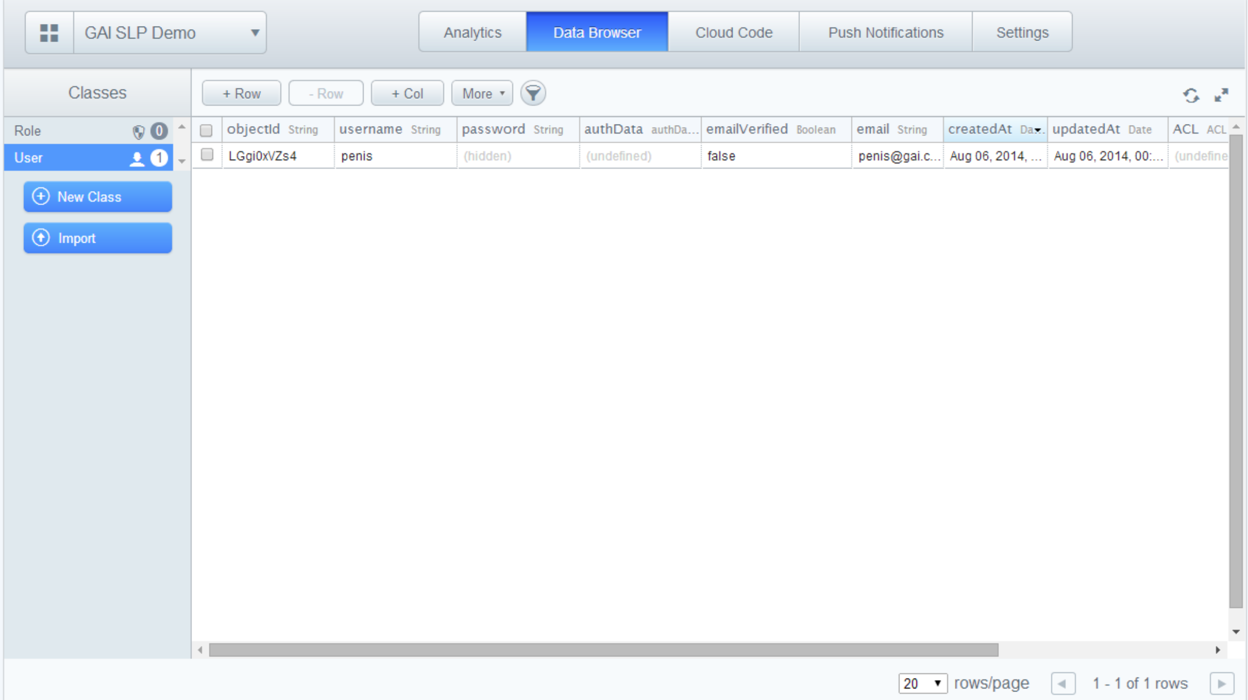
2. On the buttons, add something like this line which allows those buttons to do things when pressed. Change "login" to the name of the method that it should call. This is in the layout file for the app. In this case it is in the fragment that the activity shows.

```
android:onClick="login"
```

3. For the login method. You want something like this. The method is called by the button on click. It then takes the information from the username and password EditText objects in the layout and sets it to local variables declared at the top of the class. It then logs the user in and calls two other methods that tell the program what to do after. This is all in the main metod

```
//logs in user
public void login(View view) {
        //gets username and password and assigns it to the fields at the top
        EditText username = (EditText) findViewById(R.id.Username);
        EditText password = (EditText) findViewById(R.id.Password);
        this.username = username.getText().toString();
        this.password = password.getText().toString();
        //logs in calls different methods depending on success and failure
        ParseUser.logInInBackground(this.username, this.password, new
        LogInCallback() {
                public void done(ParseUser user, ParseException e) {
                        if (user != null) {
                                //The user has logged in
                                loginSuccessful();
                        } else {
                                loginFailed();
                        }
                }
        });
}
```

On the parse side of things, you need to make a new object in your database called user.



□	objectId String	username String	password String	authData authDa...	emailVerified Boolean	email String	createdAt Da...	updatedAt Date	ACL ACL
□	LGgi0xVZs4	penis	(hidden)	(undefined)	false	penis@gai.c...	Aug 06, 2014, ...	Aug 06, 2014, 00:...	(undefine

4. The success and failure methods will either bring the user into a new activity or will say that they fucked up somehow like so.

   The loginSuccessful method just starts a new activity, the landing page that they should go to. In this case, it will go to a part of a master detail flow containing the data that it will pull off of parse. This will be in a tutorial called "Displaying Shit Off of Parse" or something like it.
   The loginFailed method makes a toast that says that they fucked up somehow and shows it.

   ```
   public void loginSuccessful() {
   	Intent newActivity = new Intent(this, RequestListActivity.class);
   	startActivity(newActivity);
   }


   public void loginFailed() {
   	Context context = getApplicationContext();
   	CharSequence text = "You fucked up somehow";
   	int duration = Toast.LENGTH_SHORT;
   	//System.out.println(this.username + " " +  this.password);
   	Toast toast = Toast.makeText(context, text, duration);
   	toast.show();
   }
   ```

5. Congratulations you made a login screen that works. There are ways to make it with more parse things and a special parse login activity but that will be in a later parse tutorial.