420-616-AB Oracle Technologies
Fall 2016

November 7th, 2015
Worth 5%

Assignment V

(Due date is on Léa)

In this assignment, you will add server communication to your Notes app. First, you will create a login page that checks username and password. Second, in `NoteListActivity`, you will load the user's notes from the server. Finally, in `NoteActivity`, you will create/update the note on the server.

1 Prerequisites

1.1 Internet Permission

You're app must have the INTERNET permission. These are granted by the eventual user of your app. To do this, add the following `<uses-permission>` to your `AndroidManifest.xml`:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ca.qc.johnabbott.cs.cs616.notes">

    <uses-permission android:name="android.permission.INTERNET"/>
    ...
</manifest>
```

1.2 Starter files

The '`server`' folder contains the following classes, useful in communicating with the server. For usage instructions, see the Javadoc comments and the lecture demos.

Class	Description
`AsyncHttpRequest`	Make an HTTP request as an asynchronous task, i.e.: perform it in the background.
`HttpProgress`	Represents the progress of the data transmitted during the HTTP request and subsequent response.
`HttpResponse`	Contains the response information from the server, specifically the HTTP status, headers and the body of the response.

1.2.1 Optional: Assignments 1-3

The folder '`asg3`' contains a simplified version of Assignments 1-3 without some of the features (simplified editor, no SQLite db, no snackbar notification). If you had difficulty completing these assignments, you can use this as the starting point for the current assignment.

1.2.2 Optional: Assignment 4

The folder ‘`asg4`’ contains a version of the `User` and `Note` class from Assignment 4. They are implemented using the `json.org` libraries.

1.3 Application class

Add a class `NoteApplication` to your project. This class instantiated when your app launches and can be used to store information used across all activities and fragments. Start with the following:

```
public class NoteApplication extends Application {

    public static String HOST = "10.0.2.2";
    public static int PORT = 9999;
    public static String PREFIX = "http://" + HOST + ":" + String.valueOf(PORT);

    @Override
    public void onCreate() {
        super.onCreate();
    }
}
```

Update your app manifest to use your application class.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ca.qc.johnabbott.cs.cs616.notes">

    <uses-permission android:name="android.permission.INTERNET"/>

    <application android:name=".NoteApplication" ... >

        ... activities

    </application>
</manifest>
```

2 Server Integration

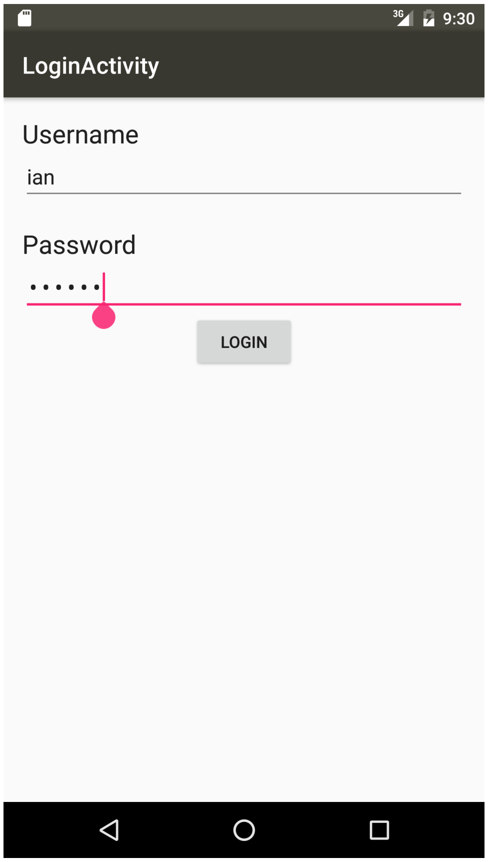
You will modify your app the use the NoteServer provided.

2.1 Login

Add a new Activity `LoginActivity` to your app to allow user login. It is the first activity to load when launching the app.

When a user types their username and password, retrieve their user information on the server and verify that their password is correct. Since each activity should have access to the user information, store it in the `NoteApplication` class. The user information is only kept while the app is running, i.e.: restarting the app logs the user out.

Use the Android "shared preferences" feature to auto-fill the username and password when the app is relaunched. `https://developer.android.com/training/basics/data-storage/shared-preferences.html`



LoginActivity

Username

ian

Password

••••••

2.2 Note List

Instead of reading the notes from the SQLite database, load the list of notes currently stored on the server. Load only those created by the logged in user.

2.3 Note Editor

Instead of creating a new note or editing a note in the SQLite database, perform these operations on the server. A new note must be associated with the logged in user.

3 Requirements

- Your program should be clear and well commented. It must follow the "420-616 Style Guidelines" (on Léa).
- Work from your Assignment 3 and Assignment 4, or the starter files provided.
- Your project includes an app class `NoteApplication` with server information: IP address and port. All server requests must use the information in `NoteApplication`.

- Your app must not perform network requests on the UI thread, i.e.: the lines

  ```
  StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
  StrictMode.setThreadPolicy(policy);
  ```

  must not be anywhere in your project.
- Submit your project using git. Follow the Git Submission instructions on Léa.