

COP290

MOODLE PLUS ANDROID APP

Mayank Rajoria (2014CS10233)

Prakhar Gupta (2014CS10290)

Prakhar Agrawal (2014CS10207)

Course Coordinator
Vinay Joseph Ribeiro

Abstract

This assignment pertains to the development of the front of the moodle app, a learning management system, with the help of a locally deployed server created using a set of provided APIs to interact with it using GET requests. The app contains functionalities of logging in, logging out, view courses, assignments, grades, notifications and view and create threads and add comments to it. The application has been made in order to follow the design practices by Google and made as user friendly as possible.

1 Introduction

This android app has been made using the open source Android SDK provided by Google Inc. The application first provides user with a login page which asks the user for login credentials to access the account.

The app presents a compact easy to use interface for the user by means of tabviews wherever possible thus making the design minimalistic and saving the user from the hassle of clicking a series of links. Presence of Floating Action Buttons for creating new Threads and adding comments to threads again provides the user quick ways to reach ways to post TODOTODO data.

2 User Interface

The application interface is divided into five screens. The home screen that provides the login utility while others help the user to interact with the database. The Action Bar Menu of the activities (other than the login activity) has option to logout which also bring him back to the login page.

2.1 Registration Screen

Figure 1: Login Screen

- This screen presents the two text boxes to the user to enter his login id and password.

- The screen is also illustrated with images representing to above parameters.
- Finally there is the "Log In" button to enter the app. TODOTODO
- The team name can be anything apart from an empty string. TODOTODO PARA
- The name of a student can contain only English alphabet characters and space characters.
- The entry number is checked for its correct format using the language of Regular Expression.
- All inputs are first trimmed of trailing white spaces before performing the above checks.

2.2 Dashboard Screen

Figure 2: Dashboard Screen

- The screen presents a tabview.
- The user can see the list of all notifications, all courses and all grades in various tabs.
- Individual items can be accessed by touching them. TODOTODO
- Upon clicking the share button an Intent is created which contains a message and a link to be shared. TODOTODO
- This intent is passed on to the system to create a chooser of various apps to allow the user to choose where to share the message.

2.3 Course Screen

Figure 3: Course Screen

- This screen again presents a tabview.
- The tabs available are Course Threads, Course Assignments and Course Grades

- Similar to the previous screen here also individual options can be touched to access further details.
- The tab with Course Threads also has a Floating Action Button to add more threads. TODOTODO
- Clicking this button creates a Dialog Box asking for the title and the detailed description of thread.

2.4 Assignment Screen

Figure 4: Assignment Screen

- This screen provides the details of a particular assignment.
- The UI is pretty much stationary.
- It contains the Name of the assignment with Details about it and the Created Date and Deadline.

2.5 Thread Screen

Figure 5: Thread Screen

- This screen contains a list of all comments of a particular thread.
- There is a Floating Action Button to add a new comment to this thread.
- Clicking this button pops up a Dialog Box asking for the Comment.

3 Implementation Details

3.1 Managing Request

- The application is based on exchange of data with a local database using locally deployed server
- The server provided is a "Web2Py" server.

- The requests to the server have been managed using a "Volley" which is a networking library for android applications and has a huge upcoming community support.
- Volley was included to the project and requests to server were managed via a Volley Request Queue of a singleton class for the entire app.
- The response String, which is in JSON format, is then parsed to obtain various returned parameters depending on the request.

3.2 Validation TODOTODO

- Validating data before posting it to the server is an important part of the application as it does not allow fake/invalid users to register for the course via the application interface.
- This is implemented via the checkData() function in the same module. It checks for names being only sets of strings separated by white spaces.
- checkEntryNumber() is another function that handles the validation for the Entry number boxes of the application.
- We check for entry number via the pattern matching in form of Regular expressions.
- **Error Cases Handled:**
 - In TeamName: All Strings except the empty string are accepted. Handled in the method checkData().
 - In Names: Characters other than [A-Za-z" "] in the entry number are not allowed in the names. Handled in the method checkData().
 - In EntryNumbers: Check for all errors of incorrect number of characters or incorrect position of letters or invalid entry year. Handled in the method checkEntryNumber(). All probable error cases are extensively handled using simple regular expression.

3.3 Autocomplete & IntelliSense TODOTODO

- On writing first few characters of name, the application starts suggesting people it thinks you may be naming. Since we had access to only data of the department people, suggestions are based on this data only.
- Similar suggestions appear on typing entry numbers using the same `AutocompleteTextView`.
- On completing the name field the application tends to complete the entry number field correctly by automatically determining the person's identity based on the name. This is the IntelliSense feature which is unique to our application.
- This feature is based on our access to the department data from the CSE website. We store entries of student names vs entry numbers which we later use to predict entry numbers based on names.

3.4 Miscellaneous TODO

- Keyboard hiding feature has been added, which hides the keyboard whenever the user clicks outside any of the `EditText` text box.

4 Project repository and guildlines for Contribution

The code for the project is being maintained in this repository: <https://github.com/Prakhar0409/MoodlePlus.git>. Read the contribution guidelines given on the github project page to contribute. Don't forget to document changes.

Voila! you have made it till the end.

5 Citations

- developer.android.com/training/index.html
- www.javatpoint.com/android-tutorial

- developer.android.com/training/volley/index.html
- [http://stackoverflow.com/questions/33573803/
how-to-send-a-post-request-using-volley-with-string-body](http://stackoverflow.com/questions/33573803/how-to-send-a-post-request-using-volley-with-string-body)