

COP290: Application for Moodle IIT Delhi

Aayan Kumar (2014CS10201)
Shreyan Gupta (2014CS10485)
Vaibhav Bhagee (2014CS50297)

February 24, 2016

The application provides an easy-to-use user interface to access the IIT Delhi Moodle on android platform. The application interacts with a locally deployed Web2Py through API calls and provides the relevant functionality to the user.

1 User Interface

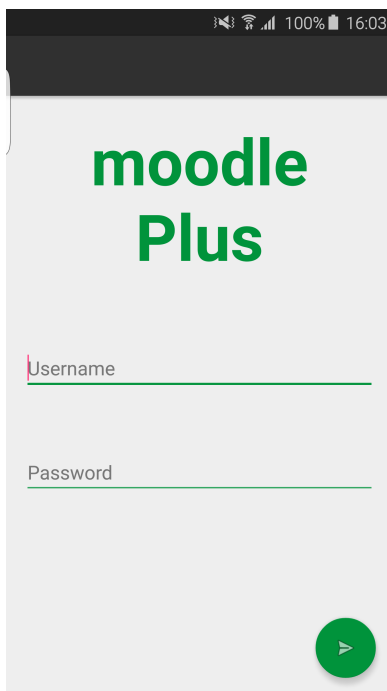


Figure 1: Login Page

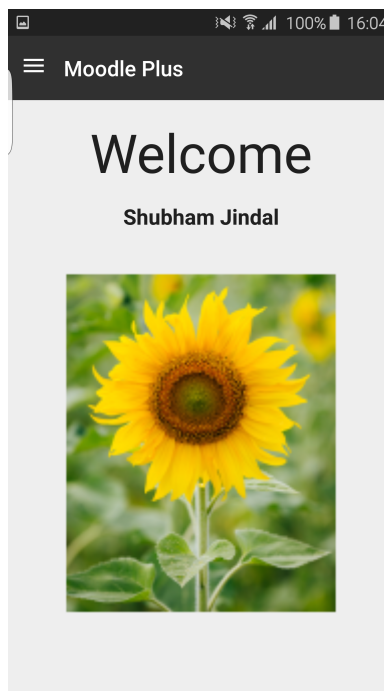


Figure 2: Home Page

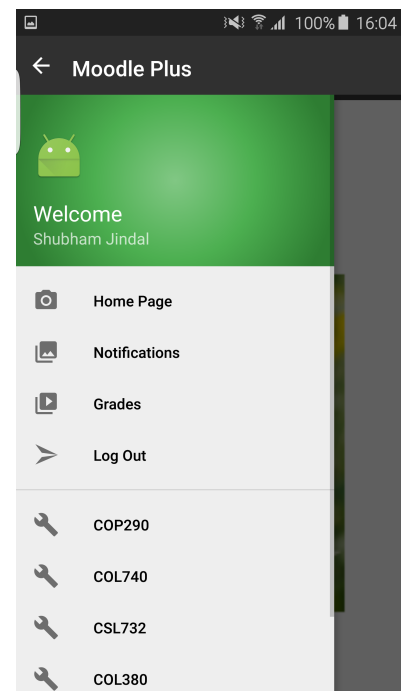


Figure 3: Navigation Drawer

- The user application contains the following screens for the functionality mentioned alongside:
 - * A **Login Screen** is encountered as the user opens the application prompting for the login credentials. The user proceeds by clicking the floating action button present on the screen
 - * On a successful login, the **Home Page Screen** appears. This home page screen contains a hidden **Navigation Drawer** which is accessible from the button on

the top left. Apart from the navigation drawer, the Home Page also contains the Profile information of the user who has logged in

- * The **Navigation Drawer** which shows up on clicking the button on the top left of the screen contains the pages to which the logged in user can navigate. It contains the links to the **Notification and Grades** page, the **list of courses** in which the user is registered and the **Link to Logout** off Moodle.

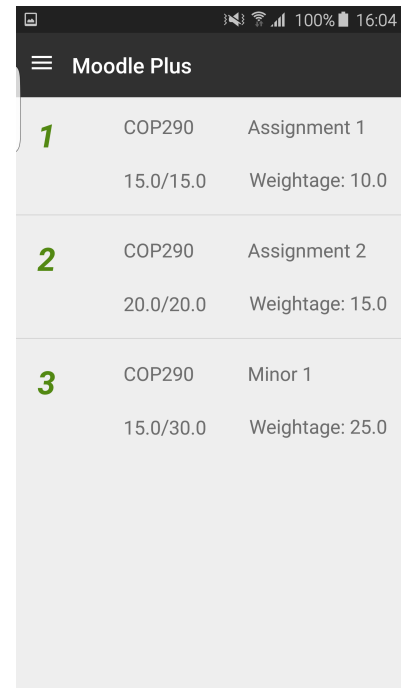
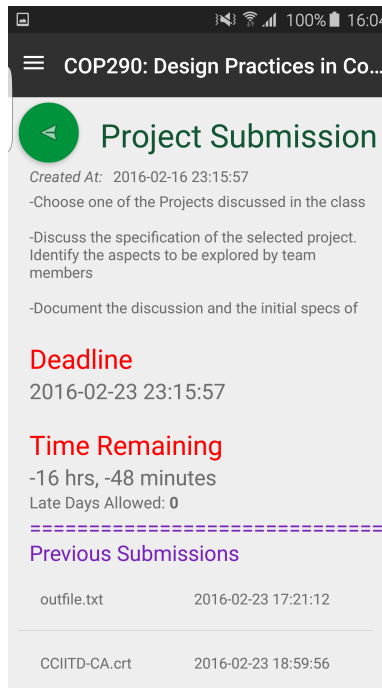
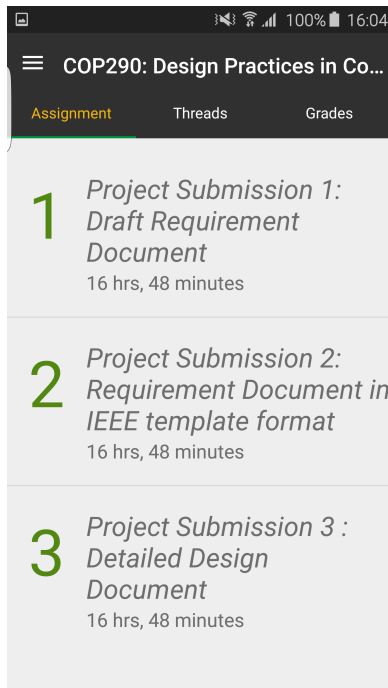


Figure 4: Assignment Page Figure 5: Assignment Detail Figure 6: Grades Page

- – * The **Course Home Page** which shows up on clicking the Course code on the navigation drawer is the Home page of a given course. It contains three fragments **Assignments, Threads and Grades** which can be accessed by swiping towards left or right on the course Home Page.
- * The **Assignments Fragment** of the Course home page contains the list of assignments along with their titles and times remaining for deadlines. The details of a **Particular Assignment** can be accessed by clicking on the title of the relevant assignment whose details you want to access.
- * The **Threads Fragment** of the Course home page contains the list of course threads along with their titles and descriptions. The details of a **Particular Thread** can be accessed by clicking on the title of the relevant thread whose details you want to access.
- * The **Grades Fragment** of the Course home page contains the list of course grades of the student.
- * The **Notifications Page** which shows up on clicking the Notifications tab on the navigation drawer is the page which shows the list of notifications.
- * The **All Grades Page** which shows up on clicking the Grades tab on the navigation drawer is the page which shows the list of grades of the user in all the courses he has registered for.

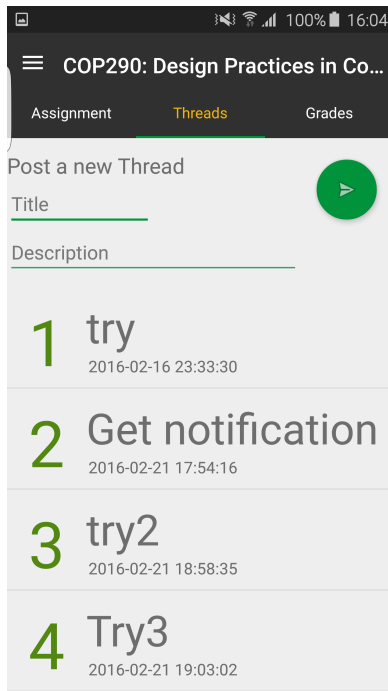


Figure 7: Threads Page



Figure 8: Thread Detail

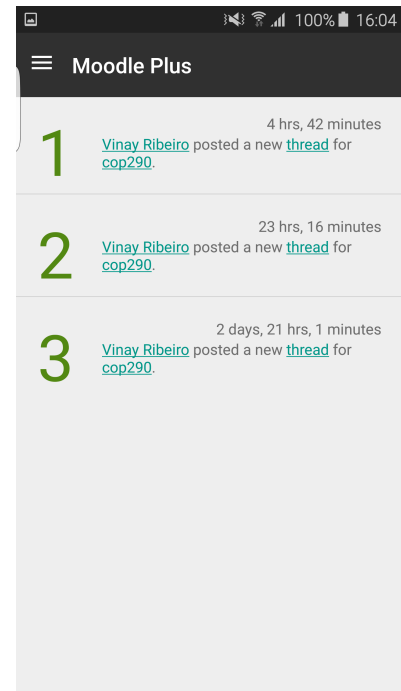


Figure 9: Notifications Page

- The functionality of refresh has been given to the user through the **Swipe Refresh**. Once the user refreshes, the app will go to the home page and refresh the data from the server.
- The appropriate data validations are imposed on creating new comments and threads, and error messages in form of an Alerts and Toasts are flashed if the data is found to be of an incorrect format.

2 Implementation Details [1] [6]

- The application makes GET requests to the locally hosted server and gets a JSON response. Modularity has been ensured in interaction of application with the server and Parsing of the JSON objects to extract relevant information.
- There is a single static class which is responsible for interaction of the server and the application. The class contains methods which take in the parameters (when required), make the GET requests to the endpoints mentioned in the problem statement and wait for the responses using callbacks. The String responses are registered as data items in the static class and are accessed by the JSON parsing objects. This eliminates the need of passing the required data using bundles to other activities in the application and makes changing the format of get requests easy.
- The basic problem which arises here is that the callbacks are asynchronous and one needs to move over to the subsequent fragments or pages once a response of the GET requests has been registered. To ensure this, each of the callback method sets a flag to true on response and further action is taken only when the flag is set to true.
- The parsing of JSON objects [5] has also been done in a modular manner and every JSON response has a corresponding class associated with it. This takes in the string response and sets the data items of the class preserving the structure of the JSON data.

Whenever a JSON object is to be parsed, the corresponding class' object is instantiated and parsing is done.

- The user interface of the application has been created using a Navigation Drawer activity and Swiping activity using fragments. There are only two activities in the application - Login and the Home Page. The notifications, grades, threads, assignments are all implemented using Fragments within the Login activity or the Home Page Activity. The lists of assignments, threads and grades are populated by creating ListViews and dynamically populating them.
- The refresh functionality is implemented using swipe refresh. It waits for a response for a period of 5 seconds on exceeding which, it displays a server timeout error.
- The three basic data validations which are used in this application are basically during the creation of new threads and addition of comments to existing threads. The check is for empty title, description and comments.
- The android library volley was used to interact with the server. The data was obtained on the server through an HTTP-GET request [3] to the server endpoints specified in the problem statement. The string response obtained from the server was parsed using the JSON parsing classes and used further. [2].

The code for the project is being maintained in this repository [4]:

<https://github.com/vaibhavbhagee/Assignment-1-src.git> .

References

- [1] Android developers.
<http://developer.android.com/index.html>.
- [2] Android network tutorial.
<http://developer.android.com/training/basics/network-ops/index.html>.
- [3] Get request using volley.
<http://developer.android.com/training/volley/requestqueue.html>.
- [4] Github tutorial.
<https://www.codecademy.com/learn/learn-git>.
- [5] Parsing json objects in android.
http://www.tutorialspoint.com/android/android_json_parser.htm/.
- [6] Stackoverflow.
<http://stackoverflow.com/>.