COP290 - Moodle Android App: Design Document

Aditi(2014CS10205), Ayush Bhardwaj(2014CS10091), Nikhil Gupta(2014CS5140462)

February 23, 2016

1 Overall Design

Our overall design of the app would consist of three main activities. The first one would be the splash screen, which would be displayed for a fixed amount of time. The next activity is the main login page, where there are separate fields to login as a student or as a teacher. After the user logs in successfully, the main dashboard appears. It is designed differently for students and teachers, to give personalised experience.
As per the requirements of the assignment,

- Different layouts [3] have been made for tablet and mobile phone view for all the three activities.
- Different classes have been defined in separate files to maintain modularity in code.
- **Doxygen** has been implemented to create an HTML documentation of the application structure.
- Various resources, such as strings, images and styles, have been included in separate files.

2 User Interface

2.1 Splash Screen

The files activity_msplash.xml and content_msplash.xml define the layout of the splash screen. It consists of a big image that contains the logo and name of the application. An animation [1] has been applied on the app logo to make it grow and shrink. Below is the screen shot of the splash screen in tablet view.

2.2 Design Of LoginPage Activity

- The layout of main login page has been defined by activity_login_page.xml and content_login_page.xml files in the layout folder.

The following are the screen shots of the main page on different orientations as well as screen sizes [2]:

2.3 DashBoard (Tab_view)

It appears when the user has been logged in successfully, as a student. It is dividied into multiple tabs, as explained below:

- **Overview**
- **Grades**
- **Courses** : The elements in the list of courses are clickable, and a new page(CourseTab) containing the details of the particular course opens up on clicking on a course.

To make the navigation among various sections of the app easier, a **Navigation drawer** has also been included in the Tab_view activity. It appears on clicking on the icon on the top left, or also by swiping to the right on the screen. Below is the screen shot of the dashboard , with and without the navigation drawer.

2.4 Course Page (CourseTab)

This page appears on clicking on any course in the list on the dashboard. It contains a TabView, comprising of the five fragments, listed as follows:

- **Overview**
- **Assignments**
- **Grades**
- **Threads**
- **Resources**

3 Implementation Details

The application structure is divided into two parts, one set of classes to handle the view and events of the user interface, and the other set forms the back-end.

3.1 Layout Handling

These classes are used to populate data in the fragments created in the CourseTab page.

3.1.1 Course_Assignments

3.1.2 CustomListAdapter

This class has been made to implement a custom List View. It extends the BaseAdapter class, which basically handles the poplulation of data into the items of the ListView.
Fields : It has the following fields:

- *Titles* - This is the list of strings that populate the title text box of each custom list item.
- *Times* - This is the list of the time of creation/latest update of a particular item in the list.
- *Serial* - It is a list of serial numbers, denoting the order in which the items are displayed in the list.
- *context* - It denotes the main application over which the fragments containing the custom list view are being inflated.
- *inflater* - It is an object of type LayoutInflater.

Methods : The following methods,other than th econstructor have been defined in the class to implement the custom List adapter -

- **getCount()** - Returns the number of items in the ListView.
- **getItem(int)** - Returns the list view item corresponding to a particular position.
- **getItemId()** - Returns the position of a listView item.
- **getView(int,View,ViewGroup)** - Returns the inflated view of the

3.2 BackEnd Classes

3.2.1 MSplash Class

This class contains the various auto-generated methods to initialize an activity.
A handler has been added to execute the method Run() of a runnable after the fixed time specified by a private variable, splash_time.
The Run() method creates a new Intent to start the main activity.

3.2.2 LoginPage Class

The LoginPage class handles the dynamics and events of the main login page. The following methods have been included other than the standard methods:

- **grow1** :
- **grow2** :
- **login** :
- **login1** :

3.2.3 MyApp_cookie

This class has been made to handle cookies.

3.2.4

4 Future endeavours

- Keep local cache of changes done, at mobile level and sync them with the global server as soon as internet connectivity is supplied.

5 Source Code

The source code of the project is maintained in the following repository:
https://github.com/aditi741997/Moodle-ANA.git

References

[1] Adding animations. http://www.tutorialspoint.com/android/android_animations.htm.

[2] Adding figures in latex. https://www.latex-tutorial.com/tutorials/beginners/latex-figures//.

[3] Supporting tab and phone view. http://developer.android.com/training/multiscreen/screensizes.html.