

COP290: User Registration App

Anuj Mahajan (2011CS50833)
Apoorva Gollapalli (2014CS50284)
Avinash Tantati (2014CS10259)

February 24, 2016

This is the description of a an android application that provides access to Moodle an educational service, The app works by using API calls to the web2py server provided.

1 User Interface

- We have implemented multiple activities for the application, we follow a tree like stucture that mimics the web application for moodle plus with respect to the APIs provided,

2 Implementation Details

- We have explicitly used static variables for stirng session informatio and sharing the same across activities
- We implemented some nice list views for the UI display
 - Since the UI was desired to be used on any size of the screens we, decides to use Listview with nested tree like activity browsing instead of fragments
 - For dynamic updates for Threads, notifications and comments we have cleverly implemented back buttons that refetch the data us- ing theURL request and then update the adapter for listview by clearing thr old data holder Array Lists

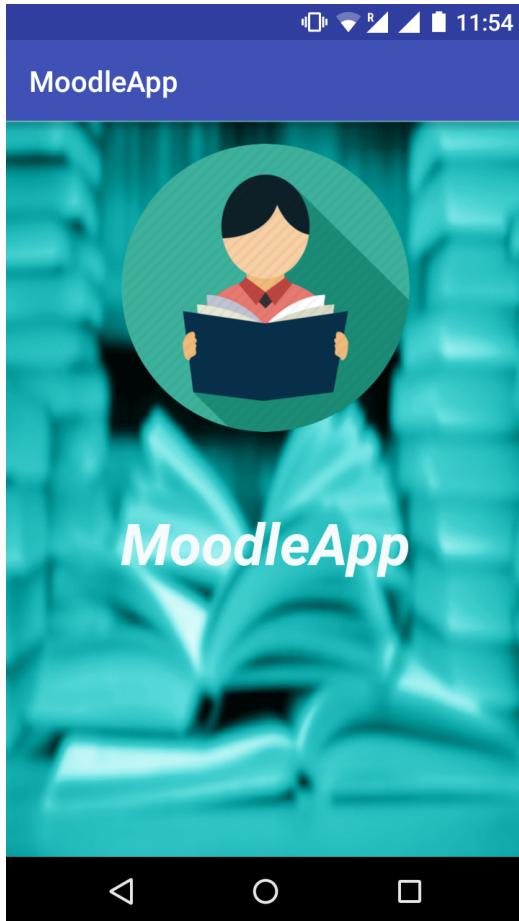


Figure 1: Splash Page

- We used the volley JSONObjectRequest request for fetching the data from the servers. We found the information provided at [2], [3] and [1] very useful.
- The generic Send Request function was modified for various API fetches by changing the URLs and adding proper JSON parser code with each request. We maintained the state of the application for the child activities by using the variables like Tsel, Csel that maintained the courses selected
- The comments and new threads generated are handled using a edit text

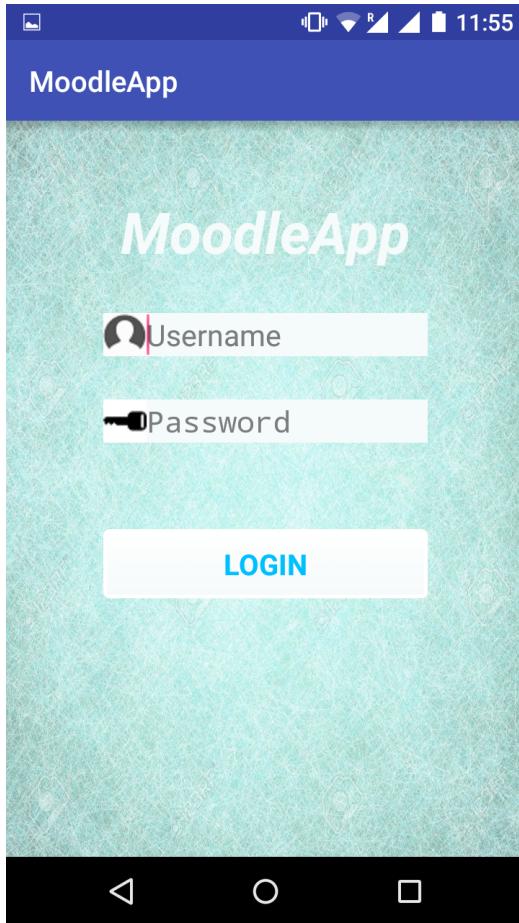


Figure 2: Login screen

entry that allows adding multiple items

- We also gave our application a cool design by exploring opportunities for changing the activity backgrounds, activity icons and changing the activity and button layouts and characteristics.

The code for the project is being maintained in this repository:
<https://github.com/anujmahajan/MoodleApp.git> .



Figure 3: Profile page

References

- [1] Android network tutorial. <http://developer.android.com/training/basics/network-ops/index.html>.
- [2] volley tutorial. <https://www.simplifiedcoding.net/android-volley-post-request-tutorial/>.
- [3] volley tutorial technoburg. <http://www.technoburgh.com/android/android-studio-volley/>.

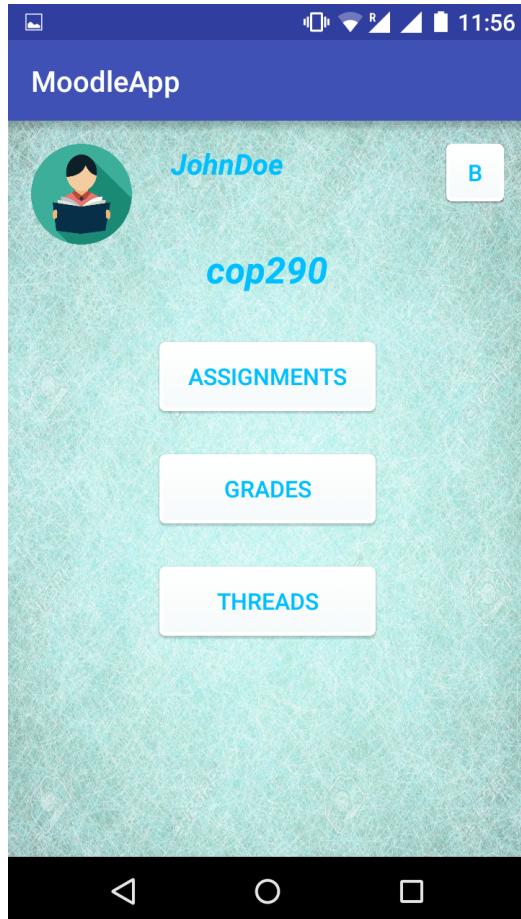
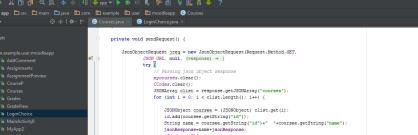


Figure 4: Course page



The screenshot shows the Android Studio interface with the Java code for the `LogCheck` class. The code handles a `POST` request to the `/check` endpoint, validating the `username` and `password` fields. It then performs a database query to check if the user exists and returns a success or failure response.

```
private void handleLogCheck() {
    JsonObjectRequest jsonObjectRequest = new JsonObjectRequest(Request.Method.POST,
            "/check", null, new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {
            try {
                JSONArray jsonArray = response.getJSONArray("users");
                for (int i = 0; i < jsonArray.length(); i++) {
                    JSONObject jsonObject = jsonArray.getJSONObject(i);
                    String name = jsonObject.getString("name");
                    String password = jsonObject.getString("password");
                    if (name.equals(username) && password.equals(password)) {
                        response.put("status", "success");
                        response.put("message", "User found!");
                    } else {
                        response.put("status", "failure");
                        response.put("message", "User not found!");
                    }
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            // Handle error
        }
    });
}
```

Figure 5: Typical Request

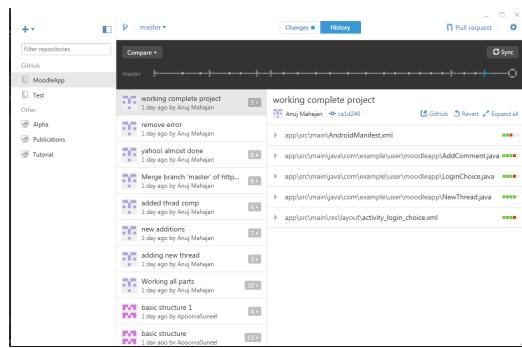


Figure 6: Git Commits