BodhiTree App: An Android Application for BodhiTree

Aman Kansal, Abhishek Sharma

Abstract—

With the growing usage of mobile computing devices like smart-phones and tablets, it becomes imperative to design and develop a mobile application for the web services being offered to users. Towards this, the project aimed at developing and improving the android application for BodhiTree - the online learning platform developed and managed by SYNERG (SYstems and NEtworks Research Group), IIT Bombay. While the base-level code was available, numerous changes and tinkering was required to make the app work, making it available on play store and adding new features to it.

Keywords—Android Studio, Django Framework, REST API

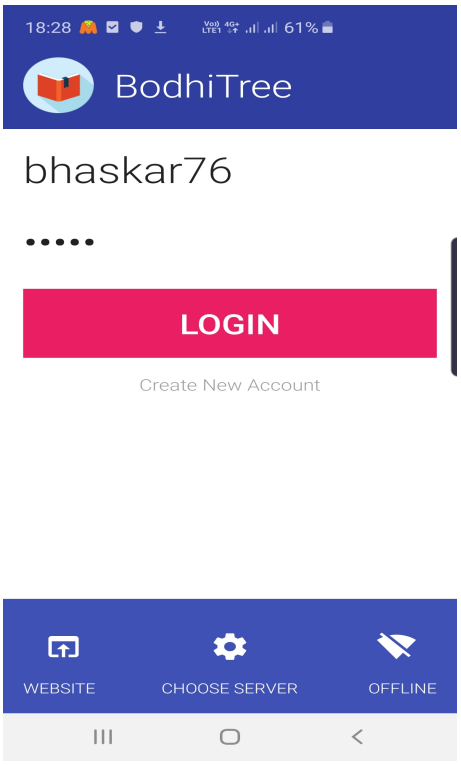
1 Introduction

The Bodhitree App was first worked upon by Bijoy Singh four years back. The app already had the code for UI and basic functionality (fetching videos, comments etc) written up, but had gradle compilation issues with it. Thus the project aimed at getting the app working in the first place, getting it released on Google Play Store (which had some issues earlier) to build its user-base and adding some new interesting features for better user satisfaction and experience.

2 Implementation Details and Process

2.1 Getting The App Working

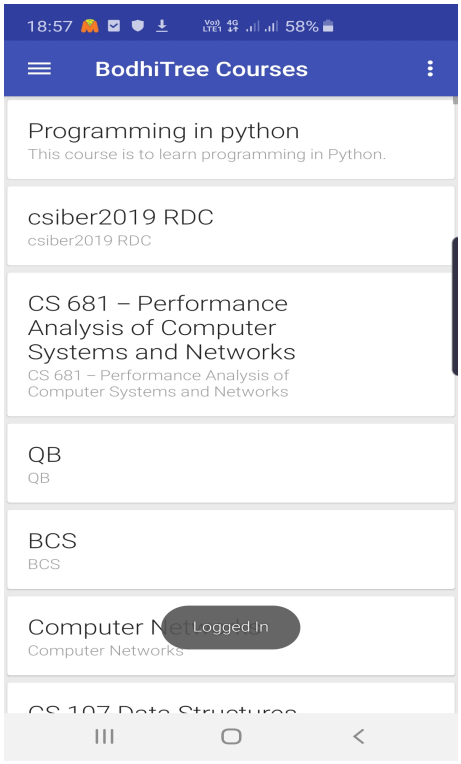
After spending a small amount of time trying to understand the code-base, we switched to a do and learn approach where we just myopically looked at the piece of code that was not working. This approach came of use at many places but had to be discarded while implementing some of the new features. To be compiling and working properly, app required changes to its build.gradle where apart from updating the gradle version, a call to Google API (as mandated by newer versions of android studio) was required. SDK version of the app was also increased because this would have been anyway required later on for publishing the app to Google Play Store. The newer API switches from the 'compile' keyword to 'implementation' keyword, while specifying dependencies. All these changes, combined with fixing some blunt syntax errors, made the app compile successfully.



2.2 Logging Into The App

The app initially was crashing while the user tried to log in. Fixing this was easy. The app was sending the request to bodhitree2.cse.iitb.ac.in

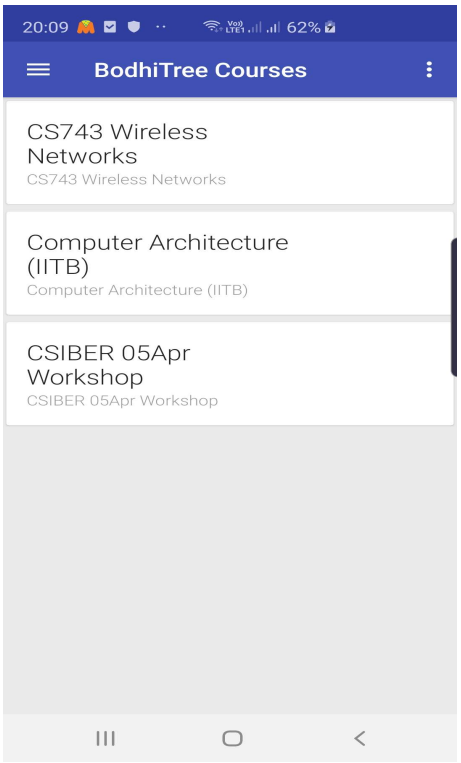
which not being functional (but might have been during initial development) sent back an invalid response. Changing it to some working bodhitree URL like bodhitree2019.cse.iitb.ac.in solved the issue. The user is now redirected to a screen like this -



2.3 Filtering User Only Courses

As might have been clear from the previous image upon logging in the user is bombarded with a list of about 500 courses. This comprises of all the courses available across bodhitree network which were being fetched by the get_all_courses api. This makes the app unusable as it is practically impossible for the user to scroll through these and find the course he/she is looking for. The situation becomes worse when multiple courses (run by different authorities) have the same common name e.g 'Computer Networks', in which case the user will have to go into each of them and check which one of them he has access too. The only problem was - the API to exclusively get student courses was not listed in server api list.

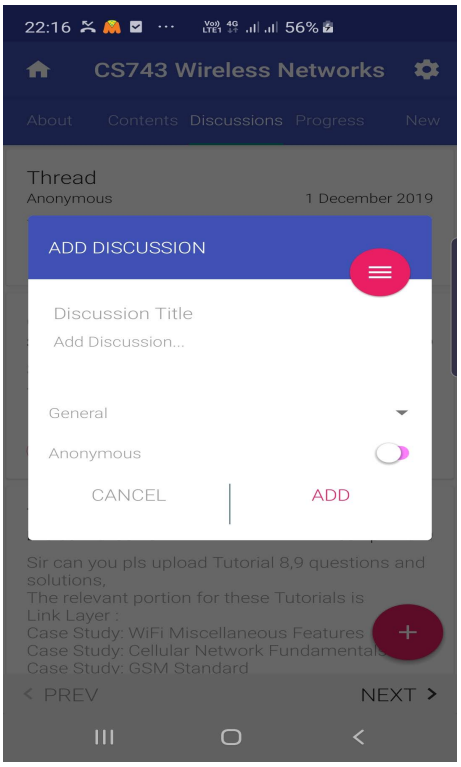
Hence, after a sufficient amount of code-surfing through the Django web-app codebase, we were finally able to obtain the required API - '/courseware/student_courses_json/'. Quite surprisingly, it was not under any API head (means the URL was not of the form '.../api/...'). So, finally using the previous API we obtained all the courses and the relevant useful metadata, and using this API the courses student has enrolled in, to finally show only user specific courses.



2.4 Fixing Video and Quiz Crashes

This was one of the most interesting and challenging problems to solve in the project, the problem being - if while a video is playing and the user presses the home button and comes back again, the video is blacked out or the app crashes. For this, according to [1], the surface is detached from the player object if one exits while the video is playing. For resuming from the same point,

reattachment of that surface is required. Since the video library being used in the app was not the traditional one that is used nowadays, and we did not want to write code from scratch (which would have been time expensive), we resorted to some logical hit-and-trial and were finally able to get around the bug. This however caused the in-video quizzes to crash, the reason being, we were always attaching object to a surface when resuming from a pause (like the one caused by pressing the home button). However, during an in-video quiz, on pausing, the surface was never detached in the first place. We took time in realising this subtlety but once visible, the bug was an easy fix.



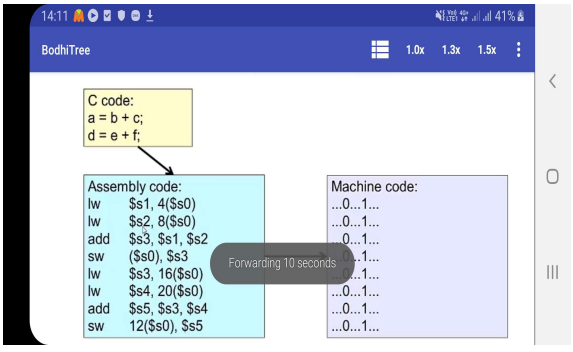
2.5 Making the Discussion Posting Work

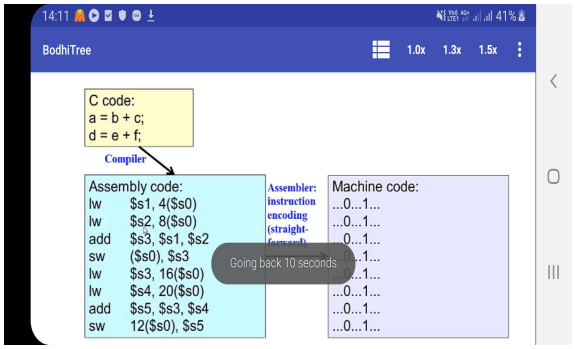
Just like the web application, the app also has functionality to view and post discussion threads and comments to the discussion forum. Although the functionality to view already existed, the post functionality required debugging. Solving this proved to be the most time consuming as well as the most easily fixed task of the whole project. Here the client-side code was fine, rather the server expected another 'referer' field in the client JSON Post Request. This field although was unused, but might have been present due to some legacy code. Thanks to Akshay Thakare, or we wouldn't have been able to get past this nasty bug!

2.6 Adding New Features

2.6.1 Video Forward and Rollback Using Swipe

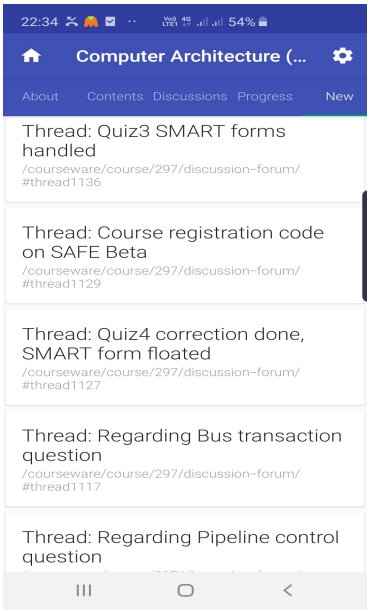
This was the first completely new functionality added to the app. It enabled the user to go forward or backward in video by 10 seconds, by doing a right or left swipe on the screen respectively. The main code for this task was obtained from [2]. Post that some debugging, tinkering and parameter tuning (for correctly being able to differentiate between up, down, left, right) was required to get the desired functionality. At a high level, once one had included [2], one can make use of the onSwipe function and simply change the video object timer according to the action.





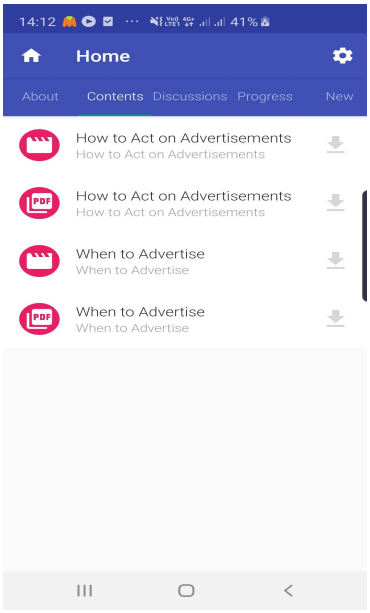
2.6.2 The NEW Tab

The web app has the functionality to let the user see what all new content has been added since he last cleared his notifications. A similar functionality in the app was implemented. For this, again, a search through the server code for correct API to use was required. After that, to make sure that the feature worked in the first run, following one of the existing tabs' implementation (Contents for case specific) proved to be a good idea. This feature has further scope of improvement where on clicking a notification user is taken to the correct place in web-application or for more challenge to a correct place in the app!



2.6.3 Slides Along-with Videos

This was a much needed feature in the app. Earlier the app only had support for displaying the videos as the course content. The corresponding slides were not displayed. This time however instead of another API giving information about the slides, the API being used to get the videos had the slide link as supplementary information. Changing the JSON DataHandler of the app and adding new panels for slides got this feature working. There was an apparent lack of uniformity in the way slide URLs were being sent by the server, so cleaning the URL properly proved to be critical for this feature as well as the bug that existed in file rendering earlier.



3 Publishing the App

Google requires newly published apps to have compile and sdk versions to be near latest. For this just changing the app metadata does not work and the manifest of the project also has to be changed, especially if the app is using some old legacy libraries. We weren't able to publish the app in the first go but thanks to the help received from Palash Jain, we were successfully able to publish the app later on and the app is live on Play Store with a current user base of 89 active installs.

4 Future Work

A lot of functionality and features were covered in this project but they are minuscule compared to what all can be done, especially in this young phase of the app. Some of things that might be worth implementing in future are listed below

- The app currently uses a number of old deprecated libraries (for eg. Video Rendering, Login Request etc). Although for

now this does not results in any hindrance in app's functionality and usage, if more sophisticated features are to be added later on, migrating the app to newer libraries might prove to be useful

- In the NEW tab, it would be interesting to add a functionality where on clicking on the notification, the user is redirected to the corresponding content/discussion in the app. Although this is not straightforward since the server doesn't respond with a direct pointer to the correct place in app, but rather a web URL. One might have to parse this URL appropriately to achieve this task
- The app although supports all the Bodhitree servers, it was developed keeping in view the flamingo server. Since the app doesn't have much user-base of people using other servers, trying to develop that, and finding and solving associated bugs (which might arise due to different format of JSON responses for the same client API) is imperative.
- Latency is observable at various places in the app - like coming back to a video after exiting the app and replaying it, display of courses upon entering the app. Fixing this may lead to better user experience.
- Currently upon downloading content it is saved not in the global local files, but rather in the local files of the app only. Enabling the content to get stored in global local files can be one of the crucial functionalities that can be added.
- There has not been a formal security analysis of the app. Although the use of inbuilt libraries should ensure that to an extent, the older libraries might be creating loopholes. Analyzing and fixing any such issues would be a good extension.
- The data limit in the app has currently be fixed at 1000 MBs. Making this configurable from the UI would enable the user to set it as appropriate to his/her device.

5 Conclusion

This project was successfully able to bring a simple working app to the students for use. Many functionalities were fixed, bugs were removed and features were added. Development is a never-ending journey and there is a lot of ground to cover, some of which has been mentioned in the previous section. We would like to express our gratefulness to the whole SYNERG team and especially to Bhaskar Sir (for giving direction to the project and helping with setting intermittent goals), Akshay Thakare (who helped with the server side), Palash Jain (for help in publishing) and Laxman Sir (for help in testing of the app). We are grateful for such a wonderful learning opportunity and wish all the best for future endeavours of students in this project and the rest.

References

[1] Stackoverflow : https://stackoverflow.com/questions/6347924/black-screen-when-returning-to-video-playback-activity-in-android

[2] VideoView with Gesture(Swipe function) in android - Stackoverflow https://stackoverflow.com/questions/22904826/videoview-with-gestureswipe-function-in-android