Nikola Peevski
Spike Exercise
Dev log

Dev day 1. Decided to start it slow and steady. Since our project is going to be based on Swift we agreed on doing the Spike Exercise with the same stack. We talked about using Firebase as well for the back end, since writing a fully-fledged backend is a problem. I looked over tutorials regarding swift:
https://developer.apple.com/library/archive/referencelibrary/GettingStarted/DevelopiOSAppsSwift/BuildABasicUI.html
https://www.youtube.com/watch?v=6Zf79Ns8_oY
https://www.raywenderlich.com/464-storyboards-tutorial-for-ios-part-1

Dev day2. After some basic understanding I set up a Github repo, added git ignore to ignore the junk XCode leaves and started fiddling around with it and thinking about how the app should look like and which story should go after which.

Dev day3. It is a bit strange, I am not used to the whole story concept and am trying to still understand how the story board works and items interact there. Today I surfed a lot through stack overflow how to connect the views to the code, although there is an official tutorial things seem a lot more complicated than what apple's guide presents.

Dev day4. I think I'll start reading about Firebase now. I have not used it in 2-3 years and even then I didn't spend that much time with it. They seem to have a very well structured youtube channel which I plan on watching on 1.5 speed and trying to wrap my head around their stack.
https://www.youtube.com/user/Firebase - Various videos

Dev day5 Okay, Firebase seems pretty fascinating. I might fiddle around with it after this exercise. It seems to be NoSQL on top of a NodeJS stack. It offers Authentication so I don't have to reinvent the wheel again, woohoo, and their so called Firebase functions, which from what I can see offer some pseudo-backend building. I'll have to check it out later. I'll start thinking about the schema of the DB (schema on a NoSQL, haha).

Dev day6 I am kinda running behind schedule. Had to take care of some last minute assignments which took way more time than expected. I have an overview of how the DB should look and I'll populate it roughly so I have somewhat of a template when I start CRUDing through it (well I don't plan on deleting any data..) I'll try to get more work done on the actual project code base and connect the Firebase SDK to it and get things going.

Dev day 7. I somehow managed to connect the firebase SDK with Swift/Xcode and got things running. Made a big mistake of not updating my gitignore file which made my repo implode with so much dependency litter, it will take me way too much time to clean. I'll roll with it for now.I've made some sketches on how the UI should look like on paper and will start implementing it. Some guys needed help with connecting the SDK as well as some xcode related issues, which I helped them. I am really running behind schedule.

Dev day8. Started pushing through the interface and started running into problems. The drag n drop way of how Swift's user interface is seems really inconsistent. The preview and the simulation look very different. I'll drop having good alignment over having functionality. I've

Nikola Peevski
Spike Exercise
Dev log

almost implemented the unregistered UI. My plan is to go over it first and reuse some components and have state management to save time.

Dev day9. Running really behind on schedule, even though I got lucky that they extended the deadline I got sidetracked into other assignments and couldn't work much. I read more about Firebase functions and will use them to sort out the newly inserted data, automate computation of the average scores and on registering of users make some scaffolding for the database.
https://firebase.google.com/docs/functions/firestore-events

Dev day10. Last day and it's already 2 hours past the deadline. I decided to take my time and make the most of it instead of delivering a really half-assed product. I'll loose around 2-3% but it is what it is. I implemented Authentification, the unregistered views are completely done and I have made the story board for the rest of the app. I was reading a bit on how the 'back' button works and how it stacks/queues previous. I implemented adding semesters and courses very easy. Deviated a bit from the specs by having shared courses/assignments. The profile essentially ties to who wrote a review and who submitted a course. From there on it will be very easy to transition to private course list, but for now I'll settle with this. It took me a bit of time to figure out how to remove the previous view, since when adding a new review/assignment I didn't want to end up in the 'add assignment/review' screen by pressing the back button. Also I managed to implement all the Firebase functions I wanted to except for the Scaffolding for the user. It still saves the profile data to the db but I had to do it from the app itself instead of a async trigger from Firebase. It looks so nice when everything comes together.
Looking back at the time I invested into this exercise I think I should improve my time management or simply do assignments faster. I think I'll do some side projects with Firebase or the alternative "Parse" I found. Also Swift seems like a TypeScript like language which I found very comfortable to work with. The timesink mainly came in the form of figuring out why and what happened in Xcode and the story board rather than the language itself. On the flip side since time was of the essence, I had to write a lot of spaghetti code, having a lot of repetition, not using proper naming convention, unit testing or saving my hardcoded strings to a Constants file. I don't follow the notion of flooding you code with comments and feel like your code base should be readable without the need for comments. But for this exercise I am not sure, maybe I should've written more. Overall, I feel more confident to start working on the actual project.
https://parseplatform.org