11. Conclusion

11.1 Introduction

This chapter concludes the project by examining the project plan; methodology used; choices made in the course of developing the project; and offers recommendations to future final year students.

11.2 Project Plan

The initial project plan (Appendix M) was poorly planned and the author strayed widely from it whilst developing the project. It’s difficult to accurately plan for a project that’s never been embarked on before and this was clearly obvious in the initial Gantt chart. Initially the author planned to use embedded databases for storing data; Bootstrap for styling; and sync data in embedded data with data in database server. But in the end the author chose a different approach, Ionic Framework, that didn’t require embedded database and had no need for Bootstrap. Even the amount of time allocated to write the report was inadequate: 5 days! Another reason the author could not follow the initial project plan was because of the demand from other course-works -- allowance wasn’t given for other course-works.

The resultant effect of poor planning and lack of compliance to project plan was an uncoordinated approach that can’t be described as an agile development lifecycle. A new life was breathed into the development when the author was re-assigned to another project supervisor, who was instrumental in helping the author define the software architecture and general approach to developing the project.

On 1st of January 2016, the author was behind schedule by three months: the high-fidelity design was supposed to be completed by 22 October 2015 but the actual completion date was 25 January 2016. The following steps were taken to turn things around:

* The author re-planned the project (Appendix N)
* The author tracked hours spent on the project using Hours Tracker app
* The author kept notes of problems encountered and what worked well
* The author quit his work to focus on the project
* The author put in long hours to develop the project

The choice of Ionic framework and agile methodology worked well but the author wasn’t able to implement TDD and BDD during the course of this project.

11.3 Agile Methodology

The author used an agile methodology in development this project. Key traits include adaptive planning, continuous improvement and collaborations with client. Though an agile methodology requires co-operation between the client and the developer, the author was unable to maintain daily or weekly communications because of the demand of other course-works. And whilst face-to-face is the best mode of communication with the client, the author improvised by chatting on Slack and Skype with the client.

11.4 Ionic Framework

The Ionic ecosystem helped in rapidly developing the project. The Ionic creator was used to prototype the project by dragging and dropping UI components. The Ionic framework was used to implement the API and it easily manipulates the DOM by two-way binding. Testing and debugging on device was possible by running the terminal command “ionic android –l –c -s” and using the chrome browser respectively. Sharing the app was made easy by the use of the Ionic View app.

11.5 TDD and BDD

Though the author acknowledges the merits of TDD and BDD and planned to use these methodologies to develop this project, the author could not due to the tight schedule. TDD and BDD feel like putting the cart before the horse to the author especially when the author is unsure of how the project will proceed. For instance, the author initially wanted to use an embedded database and had he written tests to accommodate the embedded database, they would have been a complete waste of time and effort. The author believes to use TDD one needs to be at least familiar with the project to be developed. Also the author doesn’t believe TDD is possible when on a tight schedule. Therefore, proper planning and time allowance are needed for TDD and BDD to be a success.

11.7 Ionic apps

Native apps are the best when it comes to performance and complex UI designs but because mobile platforms support different programming languages for developing apps, code re-use is impossible.

Ionic apps show great promise but the performance doesn’t mirror those of native apps. In the end they are a compromise between performance and maintainability.

The author believes the type of project determines whether native or hybrid apps should be used. This project didn’t require a complex UI, the size of CiaB’s development team is small and Ionic, a JavaScript framework, is far easier for the development team to learn. More tests still need to be carried out on older Android OS though.

11.8 Mobile Learning

The author is of the belief this mobile app, with some fine tuning, will make it easier for Coachees to manage their bookings and generally be more engaged with their coaching programmes thereby meeting the objective of developing the this project and leading to an increased completion rate of coaching programmes.

11.9 Tools Used

The author in the development of this project used the following tools:

Atom IDE, JSON Editor Online, Windows Tidy (enabled author place two or more documents or applications side by side), Git, version control, Ionic Creator, Ionic View and Trello.

11.10 Recommendations

Recommendations for someone embarking on a similar project are:

* Start early
* Develop major bits of the project before writing your report
* Project plan should accommodate time dedicated to other course-works
* Use a time tracking app to record time spent
* Keep a journal

In all, considering where I was at the start of the year, this project has been a massive learning experience. It meets most of the requirements at the time of writing this report. Collaboration with the client to complete the project will continue after the exams.