Reflection Report on Campus Connections

Team #2, Campus Connections
Waseef Nayeem
Zihao Du
Matthew Miller
Firas Elayan
Abhiram Neelamraju
Michael Kim

1 Changes in Response to Feedback

1.1 SRS and Hazard Analysis

We resolved all the TA feedback and team 11 feedback for HA (details viewable in git issue tracker). We updated the entire FMEA table since we misunderstood the meaning of 'Effect of Failure' and 'Cause of Failure' in revision 0, we also removed some failure modes that do not make sense, like the forget password information. The scope in HA is also updated 'any hazard to the user (not admin)'. The new requirements are added to robustness and security sections of SRS respectively.

We also resolve the TA and team 11 feedback for SRS. We rewrote SRS requirements in early February due to a change of scope. We discussed with our supervisor and removed some features from Map and Chat sections (e.g. audio message, friend with avatar on the map). Then we also polish some non functional requirements to make them more verifiable. A traceability table is added and all the sections are updated with the latest design. Some old designs like AWS server, ARKit/ARCore are replaced with the current design (Azure server, Vuforia engine). We also align SRS with later document (VnV Report and VnV Plan) as well whenever a change in made.

1.2 Design and Design Documentation

We resolved all feedback from the TA and Team 11 (details viewable in git issue tracker). The Module Guide (MG) was updated to match our implementation more closely. This involved restructuring our hierarchy and creating a new diagram. In order to resolve an issue regarding the readability of our module diagram, we created a new version that included smaller subsystem diagrams to make the overall diagram less visually complex. We also got rid of outdated modules and renamed older modules to be consisted with their current versions.

We also updated our traceability matrix to be consistent with our current list of requirements as per the SRS.

The Module Interface Specification (MIS) required a lot of updating. The semantics of various routines were very outdated as they had been created when the implementation was still partially done. We went over every module and made the semantics more closely follow the implementation while still remaining abstract. We resolved a long-standing issue regarding the MIS for 3rd party libraries after receiving feedback from the TA. Instead of trying to generate an MIS for libraries we used, we linked the external API documentation in their respective sections. We also included an MIS for the Hardware-Hiding module as a result of feedback.

1.3 VnV Plan and Report

We resolved the TA and team 11 feedback for the VnV Plan and Report. We changed many test due to complications with large packages and conflicts from GitHub due to how the Unity environment was set up. This led to moving many test that we initially planned to be automated to be manual and changing how we ran the automated test so that some of the code can be tested through GitHub Actions. Both documents were updated to match the current state of the project and all the tests that were run. There were some sections that we could not complete during our first write up that were completed now with more up to date information.

2 Design Iteration (LO11)

The initial pages were built without good graphical user interface was built purely to enhance our understanding of the project and as a demonstration to our supervisor of what was possible. With the help of our supervisor, we updated the User Interface to McMaster color scheme and made it look much more appealing. After the Usability test, we added functions and features that were requested, and fixed many bugs that were discovered that were not found previously due to us using it more carefully than the test participants. This allowed us to have a more robust and confident final version that was shared during the EXPO.

3 Design Decisions (LO12)

Due to several constraints, assumptions and other limitations, our design turned out quite different from how it was initially planned. Some of the decisions and trade-offs were as follows:

- We assumed that we only needed to target the Android mobile platform. This gave us more time to implement more features as well as leniency in terms of implementation as we did not need to juggle being able to run on multiple different platforms.

- One of the limitations we faced was a lack of team contribution, this ended up costing us a lot of time and heavily impacted the scope of the project. We ended up needing to cut several features such as heat-maps, 3D avatars and voice chatting. We needed to pick the top most useful features and focus on fully implementing those. We chose the AR Camera, Real-time User Map, Chatting, Lecture+Event management and User Profile/Social management as the most useful features.

- Due to time constraints we needed to prioritize certain features. We needed to make a trade-off between the quality and quantity of features in our app. We decided to prioritize polishing the UI and ended up completely overhauling the look and feel of the UI. We made the app's appearance consistent with official McMaster apps and implemented additional feedback from our supervisor to improve the usability of the app.

4 Economic Considerations (LO23)

Our product resonates strongly with McMaster University students. We believe in fostering a vibrant community and enhancing campus life. To achieve this, we plan to engage McMaster Student Centre members to join our community as administrators, thereby bolstering our product's visibility on campus.We also aspire to collaborate closely with club leaders, encouraging them to utilize our application for event organization and announcements. By doing so, we aim to streamline campus activities and promote a sense of inclusivity among students.

Our product will be freely available, aligning with other tools designed for McMaster students. Our primary focus is not profit-driven; instead, we are dedicated to enhancing student experiences. As such, our project is open-source, reflecting our ethos of community-driven collaboration.

Our target demographic includes all first-year students from the Computer Science and Engineering departments, alongside some club members. We estimate our potential user base to encompass approximately 1000 students.

5 Reflection on Project Management (LO24)

5.1 How Does Your Project Management Compare to Your Development Plan

There were many compared to the development plan that were caused due to specific issues related to the package being used. Another issue we encountered was complications of many automated components clashing such as GitHub and Unity having inconsistent automatic object resolver. We narrowed down our scope and did not use some of the technology we planned on using due to delays and lack of strong leadership and motivation. The team had significant issues regarding communication and following the roles and workflow initially, but after meetings with the TA and professor these issues were mitigated. We had weekly meetings going over work done and what work needs to be done, and later on had additional work meetings to try to push out more for before the deadlines.

5.2 What Went Well?

Throughout the project many issues were encountered. These issues were swiftly resolved or a workaround was found when it seemed too difficult or inefficient to fix. This allowed us to progress in many feature improvements and development quickly and meet deadlines. For technology, the back up system from Firebase Database saved us from losing significant amount of data. The commit history for GitHub was the same, where even if some of our work was overwritten by incorrect merge or conflicts, using GitHub's tools the important work was easily restored. Unity's automated android dependency resolver and object reference resolver allowed us to smoothly work on the project without having to mess with library versions and setups.

5.3 What Went Wrong?

Initially, many timelines were not met due to lack of communication and motivation. Not understanding the tools we were using from the beginning caused a lot of problems. There were solutions in the documentation or someone else had encountered similar issues which were solved much more efficiently that were missed until much later. Had we found some of these solution earlier on, much less time would have been spend troubleshooting and had more time in feature development.

5.4 What Would you Do Differently Next Time?

We should come up with a clear scope of the project. When we start the project, we are too ambitious to include too many features. And it turns out many of the features have to be removed from the scope due to the time constraint. We need to have a clear understanding of our capacity and the constraint of the project to decide a feasible goal, otherwise the project quality will be affected significantly because every member is overwhelmed by features and documents to finish. If the team find the scope is too ambitious due to whatever reason, they should immediately have a meeting with all team members and the supervisor to come up with a potential remedy if they recognize the need for adaptability. A change of the scope may not always be the preferred solution, but it represents a practical approach to effectively manage unforeseen complexities or constraints.