Reflection Report on Software Engineering

Team 15, ASLingo
Andrew Kil
Cassidy Baldin
Edward Zhuang
Jeremy Langner
Stanley Chan

1 Changes in Response to Feedback

1.1 SRS and Hazard Analysis

Early into development and documentation, we consulted with the McMaster Sign Language club regarding the variations of different signs, as well as what we might need to expect when developing our sign recognition modules. Thanks to these meetings, we were able to gather necessary requirements for product early on into documentation, which helped us plan ahead as to what parts of American Sign Language we should limit our scope to.

For both our software requirements specification and hazard analysis documents, our peers gave us valuable feedback in terms of expanding on our requirements and risk assessments to make them less ambiguous. Feedback including the addition of concrete metrics to measure the criterion of our requirements, as well as changing the wording to make the requirements less susceptible to being misunderstood helped shape our SRS to be more understandable and easier to read.

1.2 Design and Design Documentation

Our Design and Documentation did change with respect to advice that Dr. Smith gave us. He adamantly insisted that we use GitHub's issue tracking feature to keep track of internal issues and bugs. Before which we simply used our internal communication channel; Discord, to resolve and notify each other of. Our Design change with respect to external research we did when we found out that we weren't able to utilize external datasets with as much control as we would have liked. Hence our design shifted towards developing our own internal data gathering tool to develop our own datasets to train the model on.

1.3 VnV Plan and Report

Planning for our VnV initially primarily involved on writing unit tests for our external and internal data gathering modules and their utility functions. As we encountered more issues during our usability testing, we made the decision to plan our tests around improving the usability of our application in order to ensure our application is easy to use for a wider range of users. As such, we employed the use of feedback forms to expedite the process of gathering any potential feedback we could use to improve the usability of our application and other services. Certain feedback such as changing the layout of all the quizzes on the exercises page and the layout of the webcam and the alphabet on the practice page showed that this was an effective way of gathering data from all of our end users. As such, we believe that our report gave us the necessary opportunity to address any usability concerns with our application.

2 Design Iteration (LO11)

Our first version was fairly primitive in its execution. We only had half of the sign alphabet functioning and the front end web page was only the landing page and a proof of concept exercises page. It is good to note that the sign recognition was not even connected to the front end then. The next iteration came about from us needing to test if the front-end to back-end connection was functioning, hence we implemented a practice page with the live webcam feed and the sign language detection ready to go. This proved useful in many ways, namely showing us that sending every single frame to the back-end for processing was overloading the Mediapipe library and causing a timestamp mismatch. An initial patch was provided to enforce concurrency but was later scrapped with a rework to the system where we moved the sign recognition module up to the front end so the only data being sent to the back-end was the landmarks from captured frames.

The practice page ended up being so useful and made sense for our application's purpose that it was fully integrated as part of the application. It was where we were able to test out new features for the exercises (Like the dynamic sign integration) as well as quickly see if the new signs added were functioning properly without having to go into a quiz. As mentioned previously, the next new things to be implemented were the addition of dynamic signs, both one-handed and two-handed. Figuring out a method for the user to input their motion was a difficult balance between usability and functionality, eventually compromising more on the usability front for the sake of functionality. The solution we arrived at was allowing the user to start a capture window of 30 frames for them to execute their sign, which would then get sent to the back-end for processing. This proved to be the best option was alternatives we considered would have resulted in the same issue as before with over loading the Mediapipe library.

The final design we arrived at was simply a refinement of the previous iteration, with a higher amount of data to work from as well as more signs added into the model's recognition. Arriving here was a matter of practicality as we had a limited amount of time before the Final Presentation and we had to determine how many more signs we wanted to add and could do so without compromising accuracy of detection. in he end, we feel we found the perfect balance for a preliminary product to showcase our applications functionality and merit.

3 Design Decisions (LO12)

We decided to train our own machine learning model ourselves due to the limited amount of quality hand sign datasets found online. These datasets had typically not been updated for years, which we considered would be a potential problem due to the ever-changing nature of American Sign Language. We also wanted to have the ability to add our own data, allowing us to add new signs, something we would not be able to do if we had relied completely on an external dataset. Furthermore, there were virtually no datasets that existed for dynamic hand signs, which would significantly hamper the quality of our application, as a majority of the signs in ASL relied on the movement of the hands. While there were online databases with videos of different hand signs (lifeprint.com), these databases had no concrete data that we could use to train a neural network with. As such, we needed a way to streamline the data gathering process. With all of these factors in consideration, we decided to develop our own internal data gathering tool that would assist us in gathering data for both static and dynamic signs.

4 Economic Considerations (LO23)

There is certainly a market for our product as evidenced by other ASL learning applications (The ASL App, Lingvano, SignSchool) that already exist out on the market. Marketing our product would be a relatively simple process as our application is the only one that integrates real-time feedback for practicing signs. All other market competitors utilize primitive learning techniques such as "What is the sign" given a video clip or "try out the sign for yourself". In terms of cost, we would need to scale our budget according to server costs as the plan is to deploy it onto a server for anyone to access with a web address. Ideally we would like to not charge for our product in order to keep things in line with our competitors in the market, however, it is good to acknowledge that they are standalone downloadable applications while ours has a maintenance and overall server costs to be concerned about. Hence, adopting a subscription based model with premium features like a majority of freemium-ware is most likely the optimal choice for monetization.

5 Reflection on Project Management (LO24)

5.1 How Does Your Project Management Compare to Your Development Plan

For the most part, we stuck to the original development plan. For team communication, we stayed true to using Discord and meeting during class time (when there wasn't class). Additionally, we used the programming languages we planned to use—JavaScript and Python, as well as React and OpenCV. We did however need to use PyTorch and Google Mediapipe in addition to what we had in our development plan to help aid in building our machine learning model. Finally, the team member roles also did not deviate much from the plan, as we all took charge in multiple aspects of the project.

5.2 What Went Well?

Our team's strength lay in our splitting of roles (Jeremy and Cassidy on the front-end and Edward, Stanley, and Andrew on the backend) and our constant communication. We leveraged Discord's channels and pins to separate our concerns and stay organized within the project. We also stuck to meeting times well as a group and made sure to continuously update the front-end and back-end teams of any completed features or anything one team needed from the other.

Our team also met with our stakeholder group, the McMaster Sign Language Club (MSLC), frequently throughout both the planning and development phases of our project. This was a great boon for us as the MSLC gave us a lot of great feedback and general guidance, especially in the sign language department where the team did not have much experience in. We gained a lot of insight into the intricacies of ASL and what features we wanted to capture in our app to best cater to the wider ASL community (including beginners). The MSLC also helped test our app and gave us further feedback leading up to the final presentation, which we were able to implement and bolster the product.

5.3 What Went Wrong?

We ambitiously planned too many features for our app that we did not end up delivering on, like account login, generating custom ASL courses for users, and server hosting. These are features that would have been great in helping us fulfill our goal of creating an all-around ASL educational app, using tried and true methods of training similar to Duolingo. We overestimated how much time we would need to achieve all of these features, especially with how much time we needed to spend on fixing bugs and getting things to work. Fortunately, while we did not end up accomplishing all of our goals set at the start, we did capture the essence of a multi-faceted educational ASL app with multiple stages of learning implemented, i.e. a learning page, a practice page, and an exercises page.

5.4 What Would You Do Differently Next Time?

What our team would do better on next time is with regard to the project management side of things. We would like to spend more time setting up continuous integration as it would have been a great help to have full continuous integration for our project. As well, we would like to use the issue tracker better on GitHub, as it is a great way to cross-reference our work, not only for the instructors sake, but our own too. As well, with a stronger working knowledge and understanding of machine learning, we would set up a way to use external datasets with our data gathering system so that we could have access to a lot more data, as well as lose the dependency on our own data.