Development Plan
Software Engineering

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

Table 1: Revision History

Date	Developer(s)	Change
September 22, 2023	Stanley Chan	Proposed proof of concept demonstration plan
September 22, 2023	Andrew Kil	Proposed workflow plan
September 22, 2023	Edward Zhuang	Proposed some initial stage technology suggestions
September 22, 2023	Everyone	Finished the remainder of the Dev Plan
September 22, 2023	Edward Zhuang	Edited the Dev Plan for deliverable submission

1 Team Meeting Plan

As a baseline, when there is no lecture or tutorial for this course, all group members should aim to meet together. Meetings should occur at least once a week. For additional meetings, members will use the When2Meet schedule planner to determine an ideal time slot for meetings. Team meetings should always try to accommodate all members and will occur over voice call using the Discord platform, unless otherwise specified.

2 Team Communication Plan

Team communications will primarily be carried out using Discord or during in-person lecture meetings. Communication regarding project issues is encouraged

Table 2: Team Roles

Name	**Primary Responsibilities**
Andrew Kil	UI/UX, testing, computer vision
Cassidy Baldin	Frontend, backend, ASL
Edward Zhuang	Backend, computer vision
Jeremy Langner	Frontend, backend
Stanley Chan	Frontend, computer vision

to be done using GitHub issues, so that comments are directly attached to the code they concern. Communications with any supervisors, stakeholders, TAs, or instructors will be carried out with email and all team members should be added to the CC list.

3 Team Member Roles

All members will be responsible in some part for documentation, general development, testing, code review, and stakeholder outreach.

4 Workflow Plan

GitHub will be used to manage the project. The general structure of development will be as follows: Development branches called "Milestone Branches" will be opened at the start of a new deliverable. All members must then ensure that they are up to date on all the latest documents before syncing to the Milestone Branch. Once synced, members will work on that branch until that Milestone is reached. Final tests will occur to ensure no unintended errors, and after a group agreement, the Milestone Branch will be merged into main. When being merged, it must be done so with a comment on the changes made, and pushed so that all members can see the new documentation.

The GitHub Issues tracking feature will be used for all issue management as well as for keeping track of the current state of work. Issues will be raised via this feature whenever anything of concern occurs, e.g. bugs. All future issues with the same classification will work off this template.

5 Proof of Concept Demonstration Plan

One significant risk we have identified pertains to the accuracy and reliability of the computer vision aspect of this application. As this is the primary functionality for this application, it is necessary that our system should demonstrate the capability to detect and recognize complex hand signs and motions. One way we could validate our ability to mitigate this risk would be to prove that we can build a system that can accurately recognize all the letters of the American Sign Language alphabet.

Another risk relates to ensuring that the system can provide realtime feedback based on the user's hand sign inputs with an acceptable time delay. As this will be a web application, we suspect that much of the computer vision computation will be on the server/backend side. This may introduce uncertainty in creating a realtime feedback system, particularly since computer vision is more computationally intensive compared to text recognition based applications like Duolingo. For our proof of concept, we will deploy our computer vision model onto a server, and test the delay in the server's response when it receives a request to recognize a hand sign from another device.

6 Technology

- Frontend languages: JavaScript, CSS
- Backend languages: Python
- Frontend libraries: React (JavaScript), React Router (JavaScript), Redux (JavaScript), Tailwind CSS (CSS), Bootstrap (CSS)
- Backend libraries: Flask (Python)
- Computer vision libraries: OpenCV (Python), pandas (Python), NumPy (Python)
- Unit testing: Jest (JavaScript), Pytest (Python)
- Code coverage: Istanbul (JavaScript), Coverage.py (Python)
- Linters: Flake8 (Python)
- Continuous integration and branch protection: GitHub Actions
- Other software tools: Git, Overleaf, VSCode
- Hardware tools: A camera (live video capture)

7 Coding Standard

Coding standards will be language specific, with all Python code being upheld to the PEP 8 style guide and all JavaScript code being upheld to Airbnb's style guide.

8 Project Scheduling

Refer to the schedule within the course outline. In general, we will aim to stay ahead by one deliverable.