

# Development Plan

## Software Engineering

Team #2, Campus Connections

Waseef Nayeem

Zihao Du

Matthew Miller

Firas Elayan

Abhiram Neelamraju

Michael Kim

Table 1: Revision History

Date	Developer(s)	Change
Sept 25	Everyone	Revision 0
Date2	Name(s)	Description of changes
...	...	...

This document outlines the development plan for an XR App for Social Connections on Campus. It includes plans for team meetings and team communication, team member roles and the workflow plan. Furthermore, it discusses the technologies and coding standard that the team will be using and the plan for the proof of concept demo.

## 1 Team Meeting Plan

The team will have weekly meetings at 3:30 PM every Monday virtually on our Microsoft Teams group. The purpose of these meetings is to align the team around current deliverables, exchange ideas and thoughts, and assign tasks to team members. In addition to weekly meetings, the team will have ad hoc meetings if necessary. Formal meetings with our supervisor, Dr. Irene Ye Yuan will also be held every Thursday from 3:30 PM - 4:30 PM. All the meetings listed above will be held by our meeting chair and have meeting minutes created as GitHub issues. Our meeting scribe will be responsible for creating meeting minutes which include the meeting date and time, agenda, team actions and meeting notes. All team members must join weekly team meetings and meetings with our supervisor virtually or in person. If a member is absent, they must notify

the rest of the team half an hour before the meeting starts. If a member is absent from a meeting, he must reach out to other members and read the meeting notes to understand what happened in that meeting. If most of the team agrees to postpone a meeting, they are responsible for informing the rest of the team and finding another time that works for everyone.

## **2 Team Communication Plan**

The team will communicate primarily through Microsoft Teams, emails and GitHub issues. Microsoft Teams group chat will be mainly used for asking and answering questions, scheduling meetings and sharing resources related to the project. Teams meetings will be used for weekly meetings and supervisor meetings. The team will use emails for any formal requests sent to the professor, supervisors and TAs. Any emails sent must have all teammates CC'd on it. For code-related issues and task assigning, GitHub issues will be used. The team has set up a Kanban board to track all tasks assigned to everyone. Members are responsible to update these issues and report their progress in weekly meetings.

## **3 Team Member Roles**

Every member will occupy multiple roles in order to build such a big project. As the project consists of mainly three parts of development, our team will have three different kinds of roles focusing on these technical challenges. Members assigned these roles are required to have a deeper understanding of that specific topic and potential technical challenges. Other than development roles, there will be other roles assigned to everyone in order to organize the team. The following table shows all team member roles.

Table 2: Specific Member Roles

Team Member	Role(s)	Responsibilities
Firas Elayan	Server Developer	Work on the server-based aspect of the design
	GitHub Expert	Review coming PRs and manage feature branches
Abhiram Nee-lamraju	Server Developer	Work on the server-based aspect of the design
	Lead Tester	Lead the testing process and generate test reports
Zihao Du	User Interaction Developer	Work on the interactions between users and the environment
	Meeting Scribe	Take meeting minutes and book meetings for the team
Michael Kim	User Interaction Developer	Work on the interactions between users and the environment
	Meeting Chair	Organize all meetings and create meeting agendas
Matthew Miller	User Communication Developers	Work on the interactions between users
	Issue tracker	Track all GitHub issues, manage the Kanban board and assign tasks
Waseef Nayeem	User Communication Developers	Work on the interactions between users
	Game Engine Expert	Deal with bugs in the development process and answer questions of other members

Though everyone has their own roles, they still need to have a basic understanding of all parts and join all team meetings, otherwise it will be quite risky for the project if something happens to one expert.

## 4 Workflow Plan

The following steps will be used for workflow:

1. Pull new changes from the main branch
2. Create a new branch from the main branch to develop on
  - **NOTE:** Never develop on the main branch

- **Branch naming convention:** Branch Name should start with the following descriptive keywords: **chores/docs/feat/fix**, followed by brief descriptions linked with -
  - **Branch naming example:** *chores-update-readme*
3. Develop new code, as well as tests for that code if applicable
  4. Commit new code with descriptive messages
    - **Commit message convention:** Commit message should start with the following descriptive keywords: **chores/docs/feat/fix**, followed related issue number, wrapped with **():**. After that, write a brief description of the commit
    - **Commit Message example:** *chores(#2): update readme*
  5. Push those changes to the branch you created
  6. Merge the branch you created into the main branch via a pull/merge request
    - **NOTE:** Make sure all the tests pass before merging

The following tags will be used:

- **rev0:** Use for revision 0 of the project
- **rev1:** Use for revision 1 of the project

The following labels will be used for issue tracking:

- **documentation:** Use when documentation needs to be updated
- **meeting:** Use when meeting minutes need to be added
- **question:** Use when more information is needed
- **enhancement:** Use when a new feature or request needs to be added
- **bug:** Use when there is a bug in the code that needs to be fixed
- **invalid:** Use when there is something that needs improvement
- **help wanted:** Use when help is wanted on a piece of code
- **duplicate:** Use when there is another identical issue or pull request
- **other:** Use when the issue does not match any of the other labels

## 5 Proof of Concept Demonstration Plan

What is the main risk, or risks, for the success of your project? What will you demonstrate during your proof of concept demonstration to convince yourself that you will be able to overcome this risk?

## 6 Technology

- Specific programming language
- Specific linter tool (if appropriate)
- Specific unit testing framework
- Investigation of code coverage measuring tools
- Specific plans for Continuous Integration (CI), or an explanation that CI is not being done
- Specific performance measuring tools (like Valgrind), if appropriate
- Libraries you will likely be using?
- Tools you will likely be using?

## 7 Coding Standard

## 8 Project Scheduling

[\[How will the project be scheduled? —SS\]](#)