Development Plan
Software Engineering

Team #22, TeleHealth Insights
Mitchell Weingust
Parisha Nizam
Promish Kandel
Jasmine Sun-Hu

Table 1: Revision History

Date	**Developer(s)**	**Change**
20/09/24	Jasmine Sun-Hu, Mitchell Weingust	Added: Team Identifiers, Confidential Information, Intellectual Property, Copyright License, Team Meeting Plan
22/09/24	Jasmine Sun-Hu	Added: Team Communication Plan, Team Member Roles, Workflow Plan, Project Decomposition and Scheduling
Date	Name(s)	Change(s)

[Put your introductory blurb here. Often the blurb is a brief roadmap of what is contained in the report. —SS]

[Additional information on the development plan can be found in the lecture slides. —SS]

1 Confidential Information?

There is no confidential information to protect, therefore there is no agreement.

2 IP to Protect

There is no intellectual property to protect, therefore there is no agreement.

3 Copyright License

Mozilla Public License 2.0 (MPL-2.0)
https://github.com/parishanizam/TeleHealth/blob/main/LICENSE

4 Team Meeting Plan

The team will meet in-person at least once a week every Monday from 3:30-4:30 pm. Exceptions to this may include when the University is closed, statuatory holidays, or a group consensus to postpone the meeting is agreed upon. Additional meetings can be held in person or virtually through the team's discord server on a per need basis. Location and timing will be decided as a group at least 3 hours in advance. Team meetings will be structured as follows:

- 5-10 minutes of progress check-in
- 5 minutes of agenda debrief
- 40 minutes of executing the agenda
- 5-10 minutes of discussing next steps

The meeting chair will be decided at least 24 hours prior to the meeting, and rotate on a weekly basis.

Meetings with the project's supervisor will take place in-person every Tuesday from 9:45-10:15 am. Exceptions to this may include when the University is closed, statuatory holidays, or a group consensus to postpone the meeting is agreed upon.

5 Team Communication Plan

Communication is essential for a successful project. The following is an outline of how the team will communicate, the tools/platforms we will use, and the expectations for each team member regarding communication.

Communication Tools

- Github: a github repository will be used for code versioning, project tracking, and technical documentation. Additional details are as follows:
  - Project board: used to track milestones and visualize the workflow of the project using Kanban style columns
  - Issues: used to keep track of tasks and meetings, and delegating tasks among team members. Labels are used to categorize issues.
- Discord: a discord server will be used for day-to-day communication and online meetings. Below outlines the discord server structure:
  - general: text channel for general updates, quick questions and informal discussions
  - documents-and-resources: text channel for relevant files or useful links that do not belong in the github project folder
  - meetings: text channel for co-ordinating ad-hoc meetings between some or all team members.
  - external-meetings: text channel for co-ordinating and reviewing meeting agendas with individuals outside the core capstone team (e.g. capstone supervisor).
  - help: text channel for questions or issues that need prioritized attention.
  - Office: voice channel to hold any online meetings.
- E-mail: school emails will be used to communicate with individuals outside the core capstone team such as the capstone supervisor, capstone professor, external professionals, etc.

6 Team Member Roles

All team members are responsible for writing documentation, coding, testing, and creating/commenting on issues no matter their role.

Team Member	Role	Responsibilities
TBD	Team Liaison	Chairs external meetings, handles the communication between the team and the capstone supervisor, course instructors, TAs and any other external individuals relevant to the project.
TBD	Project Manager	Chairs team meetings, oversees the project timeline, ensures milestones and deadlines are met, and that team members contribute appropriately.
TBD	Lead Developer	In charge of managing and leading the technical design, coordination and testing of the project, responsible for helping teammates with technical challenges.
TBD	UI/UX Design Lead	In charge of overseeing the user interface and user experience components of the project, responsible for user research and usability testing.

7 Workflow Plan

7.1 Git Strategy

- The main branch will contain code that has been approved, tested, and is considered production-ready by the team.
- Branches will be created based on deliverables, features, or bug fixes and be named clearly based on their purpose (e.g. problem-statement, development-plan, front-end/navbar, etc.)
- Once the deliverable, feature or bug fix is completed and merged into the main branch it may be deleted.
- Once a deliverable or bug fix is complete, a pull request will be made. The PR description will include:
  - The purpose of the change
  - The issue number it addresses with a link to the issue
  - Details of testing and/or any relevant references
- All PRs require at least one teammate to review it, with the exception of documentation updates that are able to automatically merge.

- GitHub PR comments will be used to provide feedback
- Issues will be created on Git for project management (see 7.2)

7.2 Issue Management

- Every new task (documentation, features, bugs, updates, meetings) will be logged as a GitHub Issue.
- Team members may either use one of the provided templates or create a blank issue. For blank issues, they must include:
  - Title that gives a clear overview of the issue
  - Description of the task, including any necessary background or context
  - Links to related issues necessary for the completion of the current issue if applicable.
  - Label tags, Milestone category, and Project assignment*.
  - Assignees
- *Label tags are based on the type of issue (e.g. documentation), milestone categories are the type of deliverable (e.g. Development Plan), and all issues are displayed on one project board (see 8.1)

7.3 Use of CI/CD

- GitHub Actions will be used to create and manage continuous integration workflow scripts.
  - Running a code linting tool and automatically enforcing style guides
  - restricting the ability to merge pull requests that do not meet our coding standards (see section 11)
  - Ensure that the code builds successfully on each push
- If all tests pass, a pull request may be merged into a branch.
- The lead developer will ensure all necessary code is covered by automated tests.
- A rollback strategy will be included in the CD pipeline in case the project needs to be reverted to a previously stable state.

8 Project Decomposition and Scheduling

8.1 GitHub Projects

GitHub Projects will be used to manage and keep track of the project's schedule. The link to our project board can be found here The project board is organized by rows and columns, where each row is a different milestone that can be minimized and expanded, and each column is as follows:

- Backlog: Issues that have assigned low priority
- To Do: Newly created issues
- In Progress: Issues actively being worked on
- In Review: Issues awaiting approval from other teammates
- Done: Completed Issues

Each issue is categorized into a milestone, and team members will update their statuses as they progress through the project deliverables.

8.2 Project Timeline

8.3 Forming Team + Project Selection (Due Sept 16)

- Form a team (Sept 6)
- Meet with potential supervisors (Sept 12)
- Select project

8.4 Project Planning Documentation (Due Sept 24)

- Draft Problem Statement
- Create POC Plan
- Create Development Plan

8.5 Requirements Document Revision 0 (Due October 9)

- Research stakeholders
- Define scope, purpose and context of the system
- Define use cases and functional requirements
- Define non-functional requirements
- Brainstorm potential challenges
- Create traceability matrices and graphs

8.6 Hazard Analysis 0 (Due October 23)

- Define scope and purpose of hazard Analysis
- Define boundaries, components and assumptions
- Create FMEA table
- Define safety and security requirements
- Create roadmap

8.7 V&V Plan 0 (Due November 1)

- Define V&V plan and logistics
- Define system tests for functional and nonfunctional requirements
- Define unit tests

8.8 Proof of Concept Demonstration (November 11 - 22)

- Prepare POC demonstration

8.9 Design Document (Due January 15)

- List potential changes
- Define connections between requirements and design
- Define module decomposition
- Design user interface
- Schedule timeline

8.10 Revision 0 Demonstration (February 3 - February 14)

- Plan demonstration
- Conduct user testing and feedback
- Finalize demonstration

8.11 V&V Report Revision 0 (Due March 7)

- Evaluate functional requirements
- Evaluate nonfunctional requirements
- Evaluate testing methods
- Trace to requirements and modules
- evaluate code coverage metrics

8.12 Final Demonstration (Revision 1) (March 24 - March 30)

- Test and finalize project
- Create script and slideshow
- Practice/prepare demonstration presentation

8.13 EXPO Demonstration (April TBD)

- Create EXPO event poster
- Set up project for live user interaction
- Prepare main talking points

8.14 Final Documentation (Revision 1) (April 2)

- Problem statement
- Development plan
- Proof of Concept (POC) Plan
- Requirements Document
- Hazard Analysis
- Design Document
- V&V Plan
- V&V Report
- User's Guide
- Source Code

9 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?

10 Expected Technology

[What programming language or languages do you expect to use? What external libraries? What frameworks? What technologies. Are there major components of the implementation that you expect you will implement, despite the existence of libraries that provide the required functionality. For projects with machine learning, will you use pre-trained models, or be training your own model? —SS]

[The implementation decisions can, and likely will, change over the course of the project. The initial documentation should be written in an abstract way; it should be agnostic of the implementation choices, unless the implementation choices are project constraints. However, recording our initial thoughts on implementation helps understand the challenge level and feasibility of a project. It may also help with early identification of areas where project members will need to augment their training. —SS]

Topics to discuss include the following:

- Specific programming language
- Specific libraries
- Pre-trained models
- 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
- Tools you will likely be using?

[git, GitHub and GitHub projects should be part of your technology. —SS]

11 Coding Standard

[What coding standard will you adopt? —SS]

Appendix — Reflection

[Not required for CAS 741 —SS]

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. Why is it important to create a development plan prior to starting the project?
2. In your opinion, what are the advantages and disadvantages of using CI/CD?
3. What disagreements did your group have in this deliverable, if any, and how did you resolve them?

Appendix — Team Charter

[borrows from University of Portland Team Charter —SS]

External Goals

[What are your team's external goals for this project? These are not the goals related to the functionality or quality fo the project. These are the goals on what the team wishes to achieve with the project. Potential goals are to win a prize at the Capstone EXPO, or to have something to talk about in interviews, or to get an A+, etc. —SS]

Attendance

Expectations

[What are your team's expectations regarding meeting attendance (being on time, leaving early, missing meetings, etc.)? —SS]

Acceptable Excuse

[What constitutes an acceptable excuse for missing a meeting or a deadline? What types of excuses will not be considered acceptable? —SS]

In Case of Emergency

[What process will team members follow if they have an emergency and cannot attend a team meeting or complete their individual work promised for a team deliverable? —SS]

Accountability and Teamwork

Quality

[What are your team's expectations regarding the quality of team members' preparation for team meetings and the quality of the deliverables that members bring to the team? —SS]

Attitude

[What are your team's expectations regarding team members' ideas, interactions with the team, cooperation, attitudes, and anything else regarding team member contributions? Do you want to introduce a code of conduct? Do you want a conflict resolution plan? Can adopt existing codes of conduct. —SS]

Stay on Track

[What methods will be used to keep the team on track? How will your team ensure that members contribute as expected to the team and that the team performs as expected? How will your team reward members who do well and manage members whose performance is below expectations? What are the consequences for someone not contributing their fair share? —SS]

[You may wish to use the project management metrics collected for the TA and instructor for this. —SS]

[You can set target metrics for attendance, commits, etc. What are the consequences if someone doesn't hit their targets? Do they need to bring the coffee to the next team meeting? Does the team need to make an appointment with their TA, or the instructor? Are there incentives for reaching targets early? —SS]

Team Building

The team will build cohesion through (minimum) monthly team socials. The details of the events will be decided upon unanimously, during a time of low-stress, that works for all team members.

Decision Making

[How will you make decisions in your group? Consensus? Vote? How will you handle disagreements? —SS]