Development Plan
Student Evaluation App

Team 29
Nicholas Fabugais-Inaba
Casra Ghazanfari
Alex Verity
Jung Woo Lee

Table 1: Revision History

Date	Developer(s)	Change
September 23, 2024	NFI, JL, CG, AV	Initial Draft
Date2	Name(s)	Description of changes
...	...	...

1 Confidential Information?

2 IP to Protect

3 Copyright License

4 Team Meeting Plan

Team meetings will be held weekly on Tuesdays at 2:00pm at a Hatch conference room. However, team meetings are flexible, if a majority of the team (3/4 members) agrees to either cancel the weekly meeting, schedule an extra meeting, or hold the meeting virtually then the according changes will be made.

Supervisor meetings will be held every 2 weeks on Wednesdays at 3:00pm in JHE 373. Similar to team meetings, if either the supervisor requests or a majority of the team agrees to any of the previously stated changes, they will be made.

The meeting chair will head meetings and will present the weekly agenda at the beginning of each meeting which will cover the topics to be discussed. Each team member will provide a weekly summary covering the work they've done and outstanding issues they've been tackling. Discussion of these weekly agenda items and outstanding issues will be lead by the meeting chair. Every team member will leave the meeting with a list of TODO items that they plan to tackle next. The agenda, weekly summaries, outstanding issues, decisions made and TODOs will be recorded by the scribe in the meeting's assigned Github issue. Finally, meetings have a maximum duration of 1 hour.

5   Team Communication Plan

[Issues on GitHub should be part of your communication plan. —SS]
GitHub issues will be used to communicate technical information between group members.

Discord will be used as the main group communication channel for the team. A majority of group communication will be held through here including online meetings, admin details, and general unplanned discussions.

Call and Text will be used for direct communication between 2 team members and will generally be resereved for high priority and urgent communcation or campus location coordination.

6   Team Member Roles

[You should identify the types of roles you anticipate, like notetaker, leader, meeting chair, reviewer. Assigning specific people to those roles is not necessary at this stage. In a student team the role of the individuals will likely change throughout the year. —SS]

- Jung Woo Lee
    - Scrum Master
    - Developer
- Alex Verity
    - Developer
- Nicholas Fabugais-Inaba
    - Developer
- Casra Ghazanfari
    - Developer

7   Workflow Plan

- How will you be using git, including branches, pull request, etc.?

- How will you be managing issues, including template issues, issue classification, etc.?

- Use of CI/CD

Git is the most important part in managing issues, editing documentation, and developing features. Before any changes should be made, a "git pull" should be entered into the Github capstone directory for updating the local repository to the latest changes on the remote branch. Once this is complete, a branch should be created, from the main branch, for any commits involving documentation, development code, or any other changes present within the repository. With these commits, there should be comments detailing the changes that have been made. Furthermore, each commit should have the designated code name associated with it. After these commits are tested, committed, and then pushed from the local to the remote repository, a pull request should be made to merge the branch commit to the main branch. At least one approval will be required from a reviewer for each pull request to be merged with the main branch. Finally, when the commits have been successfully merged to the main branch, the branch the commits were made on will be deleted.

For managing merge conflicts, the developer should identify the conflicting changes, making sure to resolve any conflicts before new changes are made. Once all conflicts are resolved, the developer should be able to proceed in testing, committing, and pushing their changes to the remote repository.

8   Project Decomposition and Scheduling

- How will you be using GitHub projects?

- Include a link to your GitHub project

[How will the project be scheduled? This is the big picture schedule, not details. You will need to reproduce information that is in the course outline for deadlines. —SS]

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

Topics to discuss include the following:

- Specific programming language

    The project can roughly be divided into 3 main components:

    1. A database where the bulk of the site's data will be stored.
    2. A webserver which will host the website's visual data and code.
    3. Middleware which will allow for communication between the webserver and database.

    The middleware will be written in Python due to its ease of use and familiarity among team members.

    We will either use PostgreSQL scripts or an ORM (like SQLalchemy) to implement the database, the language used will depend on the implementation chosen. PostgreSQL scripts would require using only SQL, while an ORM would likely live where with middleware and as such would be written in Python.

    The webserver will be written in Javascript using the React framework. This is because of its ease of use, team member experience, and large set of available libraries which will be useful for implementing the visual elements of the website. Additionally, it was expressed to us by stakeholders that the website should be easily maintainable. Implementing the website using an extremely popular and widespread framework such as React means that there are countless resources online to help future maintainers keep the project alive.

- Specific libraries

    Depending on the implementation, the middleware will use some combination of the FastAPI and SQLalchemy libraries. FastAPI will be used to implement HTTPS commmunication rotuers between the database and webserver. SQLalchemy will be used to implement the database if it is decided that it will be implemented using an ORM.

The webserver will use a large set of both funcational and visual react libraries. For example react-navigation will be used for its page traversal functionality while libraries such as react-datepicker and react-calendar will be very helpful in implement the visual elements of a scheduling system. Additionally, the axios library will be used to form and send the HTTPS requests to the middleware from the webserver.

- Pre-trained models

This project does not include an AI component and will not use a machine learning model.

- Specific linter tool (if appropriate)

ESLint will be used for linting Javascript code while flake8 will be used to lint Python code. We chose these linters due to our previous experience using them and because they provide our preferred formatting style.

- Specific unit testing framework

The pytest framework will be used to create unit tests for the middleware code. We chose pytest over other python testing frameworks due to its simplicity, small amount of boilerplate code, and plugins which can add useful functionalities like coverage reporting. We plan to incorporate these pytest unit tests as a part of our CI plans for the project via Github actions.

Testing the database will likely be done using a dummy / development PostgreSQL database prior to making any changes to the production database to ensure that minimal migrations are required during development.

Our plans for testing our webserver's React code are not decided yet, but we're currently investigating potential testing options.

- Investigation of code coverage measuring tools

The Coverage.py Python library will be used to measure the code coverage of our middleware program. For the webserver's React code, Jest is included by default when using the "create-react-app" command and will be used to measure the code coverage of the webserver.

- Specific plans for Continuous Integration (CI), or an explanation that CI is not being done

- Specific performance measuring tools (like Valgrind), if appropriate

lighthouse for react, cProfile included with python

The webserver will be hosted on azure's web app services, meaning that azure's suite of performance measrement tools and metrics will be used as the webserver's main performance measurement system. Additionally, we plan to do practical performance tests with stakeholders by

having them use the website casually to ensure that performance during regular use is up to their standards / expectations.

Similarly, the database will be hosted on an azure container, meaning that azure's suite of performance meaurement tools and metrics will once again be used as the databases main performance measurement system. Additionally, performance of queries will be timed throughout development to determine what indices should exist on the database for practical performance.

Finally, the middleware will use the profile library included with Python to measure the performance of its HTTPS routes.

- Tools you will likely be using?

There are a number of tools/programs/services that we plan to use that were either mentioned in passing previously or not mentioned at at all.

Azure containers and/or virtual machines will be used to host both the database and middleware components of this project. Additionally, the webserver will be hosted using azure's web app services. This allows all main components of the project to be hosted in one place.

Node.js will be the server environment which we run our webserver on and npm will be what we use to manage our Javascript packages.

Git/GitHub will be used for version control on the project. GitHub projects will be used as a general project management tool to help keep track of issues, work done, and available tasks. Finally, GitHub actions will be used for CI of tests as the repository is modified.

[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

[How will you build team cohesion (fun time, group rituals, etc.)? —SS]

Decision Making

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