METCS673 Group 6 - Software Project Management Plan

TerrierMap

Team Members:

- Ryan Christopher
- Misael Gared
- Jasmine Hughes
- Siddhraj Parmar

Organization:

This project will be using a flat structure which is defined by having no rigid hierarchy and allows for collaboration in the decision-making process. Since the team is small, everyone's input and point of view greatly influences the direction of the project. As decisions are made, certain roles will have a greater impact on the final decision. For example, the preference of the UI/UX designer for the color scheme will take some precedence over the preference of the backend developers.

Roles	**Responsibilities**
Frontend Developer	Manage development for React frontend; implement HTML/CSS and JavaScript code
Backend Developer	Manage backend architecture with Firebase; implement JavaScript APIs
Full Stack Developer	*See Responsibilities for Frontend Developer and Backend Developer
UI/UX Design	Design and write user interfaces, typography and layouts; produce branding such as fonts, color scheme and logo; write CSS for frontend
Testing	Write code with Python to test the functionality; create automated tests

QA	Focuses on quality of life for stakeholders; creates additional tests
Project Manager	Manage the development of overall application

Project Management Tools:

GitHub Issues will be utilized as the project management tool for this project. It will allow the project manager to create To-Do's with brief descriptions and assign them to team members. It will allow for milestone tracking and issue notifications.

The To-Do's will have the following labels: "Question," "Backlog," "In-Progress," "Review," and "Completed".

- Question: Signifies more information/help is needed from another team member before proceeding
- Backlog: Shows To-Do's that have not been started
- In-Progress: Marks the To-Do's that are actively being worked on
- Review: Signifies a To-Do has been completed but needs review before committing
- Completed: Marks a fully completed To-Do

Risk Management Plan:

Risks

- Development Delays: This can occur due to various circumstances from technical issues to personal matters.
- Feature Creep: Additional features might get added during the development process that can lead to delays or an unfinished product.
- Data Loss: Computer crashes, codebase errors, and other unfortunate accidents occur often enough that data loss is a concern.
- Lack of Clarity: Unclear expectations from stakeholders or team leads is often an issue.

Mitigation

- Development Delays: To avoid such risks, thorough research into technologies will be completed throughout the development process. To account for any additional delays, overestimating the time it takes to complete tasks will be implemented into the timeline.

- Feature Creep: To minimize this risk, features will be added iteratively until completion to ensure a complete project.
- Data Loss: To minimize lost data, regular backups of code and data will be implemented.
- Lack of Clarity: The project manager setting clear expectations for everyone to accomplish or work on during the week will mitigate this risk.

Owners

- Development Delays: The project manager and developers will monitor this risk and will be held responsible if the project gets drastically delayed.
- Feature Creep: The entire team will contribute to avoiding this risk during team meetings and other times decisions are made.
- Data Loss: It will be the responsibility of the frontend and backend developers to ensure data is being backed up and will be held accountable if this risk occurs.
- Lack of Clarity: It will be the responsibility of the project manager to monitor this communication risk and will be held accountable for this risk.

Estimation and Scheduling:

- Cost: Due to this project being for academic purposes, there will be no monetary cost. We will use free tier services and API's in order to not spend unnecessary money on our project.
- Duration: 1 to 3 months. We do not have the time for a 16-fold range; however, the initial goal for our project is the user being able to find their way to a specific building and room number from somewhere on campus first. After that threshold is passed, we plan on adding functionalities to the application, such as nearby areas for food/drink on the user's way to their on-campus destination and information on nearby clubs/study groups tailored to the user.
- Iterations: Our team will focus on intervals of 1 week for iterations.
  The following chart describes our iteration plan:

Week 1 [Beginning Iteration] -Set up repository -Begin making project folders -Create branches for development	**Week 2** [Plan Prototype] -Create wireframes -Create frontend files -Begin populating backend data -Finalize API's chosen/design APIs	**Week 3** [Implement Prototype] -Connect frontend to Firebase -Update documentations with user stories and technologies used -Initial prototype design

Week 4 [Implement Prototype] -Write tests for use cases -Implement page design from wireframes to prototype	**Week 5** [Implement Prototype] -API implementation -Testing -Log test results to GitHub issues -Update documentation on how to run project	**Week 6** [Prototype Testing] -Update requirements -Modify code based on results of testing -Update/close GitHub issues based on status
Week 7 [Prototype to App] -Deploy as a live site -Update documentation on accessing and navigating from web -Design tests for live site	**Week 8** [Prototype to App] -Testing full app -Update documentation -Implement nearby areas functionality and study groups if time permits	**Week 9** [Prototype to App] -Bug fixes to live app -Provide updates to any outstanding GitHub issues -If unable to add extra features, include “what’s next” to documentation (nearby areas and study groups)

Documentation and Monitoring:

Progress Reporting

During the weekly stand-ups, each team member will report what they have been working on as well as any roadblocks. During this time, the project manager will take notes to ensure the To-Do’s are updated in the project management tool.

During the weekly meeting, the facilitator will allot 20 minutes for the team to review the overall progress toward the current milestone. With input from the development team, the project manager will adjust goals and milestone deadlines as appropriate. After the weekly meeting, the assigned team member will write out a brief progress report.

Metrics

- Team Member Productivity: This will be measured by the number of To-Do's marked with another label in GitHub Issues.
- Schedule Variance: This will track the amount of time budgeted to spend on a task versus the actual time spent on a task. The following equation will be used for calculations.

Schedule Variance = Budgeted Time - Actual Time

- On-Time Deliverables: This will help measure the overall success of meeting the project's milestone deadlines. It will allow the team and project manager to detect where any product delays are occurring and assist in helping minimize them. This will be measured by the number of To-Do's in "Backlog," "In-Progress," "Review," and "Question" that are two weeks behind schedule.