GPS Based Campus Room Finder

Sprint 1
2025-09-02

Name	**Email Address**
Aaron Downing	aaron.downing652@topper.wku.edu
Ryerson Brower	ryerson.brower178@topper.wku.edu
Kaden Hunt	kaden.hunt144@topper.wku.edu

CS 360
Fall 2025
Project Organization Documentation

Contents

List of Figures

1 Project Team's Organizational Approach

Sprint 1: Kaden Hunt served as the Project Manager for this sprint. The entire team met in person once per week for hour-long working sessions at the Commons on Thursdays at 2:00 PM. To maintain continuous collaboration, the group utilized a Discord server for ongoing chat, video calls, and file sharing, along with text messaging for quick updates. Furthermore, the team held brief, 20-minute meetings with the client every Tuesday to present a prepared progress report, ensuring alignment and feedback on our development process.

Sprint 2:

Sprint 3:

Sprint 4:

2 Schedule Organization

2.1 Gantt Chart v1:

The focus for Sprint 1 was focused on project initiation, team formation, and foundational planning. The primary tasks involved establishing team roles and responsibilities, setting up collaboration tools like GitHub and Google Docs, and drafting the initial versions of both the Organizational and Technical documentation. This chart outlined the critical path for researching project requirements, defining the scope, and performing an initial risk analysis to ensure the project's viability. The chart is located in the project directory at: `ToppperNav/docs/Organization_Technical_docs/TopperNav_Gantt_Chart.png`

2.2 Gantt Chart v2:

Text goes here.

2.3 Gantt Chart v3:

Text goes here.

2.4 Final Gantt Chart:

Text goes here.

3 Progress Visibility

3.1 Sprint 1 Progress Visibility

The team progressed steadily through Sprint 1 and completed the deliverables before the due date. Tasks were divided among members during class and reinforced during weekly meetings. Kaden Hunt managed the task list, prepared the presentation, and contributed to both the Organizational and Technical documentation. Aaron created the Gantt chart and also worked on sections of both documents, while Ryerson focused on contributing to the documentation. Progress visibility within the group was maintained through frequent conversations, updates in the group chat and Discord, and use of the GitHub Projects page. The Gantt chart also provided a clear overview of tasks and deadlines, ensuring accountability. Through these resources, members could report when tasks were complete, request assistance, or flag dependencies. Overall, consistent communication and shared tools allowed the team to remain coordinated and deliver Sprint 1 successfully.

3.2 Sprint 2 Progress Visibility

Text goes here.

3.3 Sprint 3 Progress Visibility

Text goes here.

3.4 Sprint 4 Progress Visibility

Text goes here.

4 Software Process Model

Our team adopted the Iterative Software Process Model, which emphasizes building the project in small, repeatable cycles. This approach allowed us to steadily refine our work by dividing development into manageable stages and reviewing progress continuously. Unlike a linear model, the iterative approach gave us the flexibility to adjust plans based on what we learned along the way, which helps reduce risk and increases the quality of the final deliverables.

During Sprint 1, our focus was on preparation and foundational work. Quality control steps included producing clear documentation, researching relevant tools and requirements, and delivering a presentation to communicate progress. We also treated environment setup as part of quality assurance by ensuring all team members had the same software installed, including Android Studio, the JDK, TeX Live, and TeXmaker. These checks provided consistency across the team and created a strong base for future development cycles.

5 Risk Management

5.1 Risk Identification

The following risks are listed and categorized by potential risk level.

- **Database Performance (High Priority):** The database may become too slow under heavy loads, affecting responsiveness and user experience.
- **Route Generation API Failure (High Priority):** The external route generation API may not function as expected, preventing key functionality of the application.
- **Compatibility Issues (Medium Priority):** The system may encounter compatibility problems across different devices, operating systems, or software versions.
- **Access to Android Phones (Medium Priority):** Limited availability of Android devices for testing may hinder progress.
- **Large File Download (Low Priority):** Users may face issues when downloading or uploading large files, leading to performance bottlenecks.

5.2 Risk Planning

The team has outlined the following plans for risk planning:

- **Database Performance:** Implement query optimization, proper indexing, and caching strategies. Load testing will be performed early to identify bottlenecks.
- **Route Generation API Failure:** Identify and test backup APIs during early development.
- **Compatibility Issues:** Use emulators and virtual machines to ensure broader compatibility across various platforms.
- **Access to Android Phones:** Gain access to an android phone temporarily and share them among team members as needed.
- **Large File Download:** Implement file compression. Clearly communicate file size limitations to users.

5.3 Risk Monitoring

The following monitoring strategies will be used for the remaining duration of the project:

- **Database Performance:** Monitor query execution times and system performance.
- **Route Generation API Failure:** Implement automated alerts for API failures or unusual latency.
- **Compatibility Issues:** Maintain a compatibility checklist and update it with each system change.
- **Access to Android Phones:** Track device usage and availability within the team. .
- **Large File Download:** Monitor client feedback related to file transfers.