Campus Sync Final Project Report

UNIVERSITY OF MINNESOTA
Driven to Discover®

Guided by: Prof. Kathryn Wust & Prof. Scott Hareland

Team Members:

Nick Kanning	Chandan Umesh Pai	Saad Saleem	David Tomlinson
kanni023@umn.edu	pai00040@umn.edu	salee036@umn.edu	tomli104@d.umn.edu

Submission Date: December 9, 2025

Executive Summary

The Campus-Sync project set out to solve a persistent challenge faced by students at the University of Minnesota: navigating the extensive Gopher Way system of tunnels and skyways. Many students are unaware of how to use the network effectively, and building access hours are scattered across multiple sources. As a result, students often take longer routes, rely on outdoor paths during severe winter weather, and lose valuable time moving between classes. The goal of CampusSync was to provide a unified, accurate, and easy-to-use navigation platform dedicated to the tunnel system.

The solution is a responsive web application that overlays tunnel and skyway routes on a detailed campus map and incorporates building-hour data directly into the routing engine. The system calculates the shortest feasible path while distinguishing between indoor and outdoor travel time. It is fully browser-based, requires no installation, and is designed to function smoothly across mobile and desktop devices. A supporting domain, campus-sync.org, hosts the live application.

Throughout the semester, the team completed several major milestones. The Figma UX prototype was finalized early in the process, followed by the MVP release on November 3, 2025, which introduced basic map layers and routing. The functional prototype, delivered on November 17, added accurate tunnel navigation and building-hour integration. We then conducted structured pilot testing with students, collected feedback, and implemented key usability improvements. The final revised release was deployed on December 9, achieving approximately ninety percent routing accuracy and meeting all explicit project requirements.

The project was carried out using a structured plan that included a Statement of Work, Work Breakdown Structure, risk analysis, and communication plan. Early risks included the complexity of geospatial data, inconsistencies in map formats, and potential delays due to limited web development experience. These risks were managed through early data validation, staged development, and continuous cross-device testing. A contingency framework ensured that the scope remained achievable within the semester timeline.

CampusSync remained within its defined scope and avoided unnecessary feature expansion, allowing the team to focus on reliability, performance, and usability. The final system demonstrates clear value, especially during winter months when indoor navigation is essential for student comfort and safety. In addition, the codebase, documentation, and prototypes provide a foundation for continued development by student groups or university partners.

By the end of the project, CampusSync achieved its main objectives: delivering a functional and user-tested navigation system, integrating accurate building-hour data, and establishing a strong framework for future enhancements. The project not only met all academic requirements but also created a practical tool that can significantly improve the daily experience of students navigating the University of Minnesota campus.

This summary provides an overview of the project objectives, the approach taken, major outcomes, and the final status of the solution relative to the original plan.

Table of Contents

Project Overview and Opportunity Statement

Problem Statement: Students lack a centralized and accurate indoor navigation tool to support time-efficient winter movement across campus of University of Minnesota.

The Gopher Way system connects multiple buildings on the University of Minnesota campus through tunnels and skyways. Despite its value, the system is difficult to navigate due to dispersed information, limited official maps, and inconsistent building access hours. First-year students, in particular, struggle with identifying which buildings remain open, which pathways are accessible indoors, and how to efficiently travel across campus during extreme winter conditions.

The CampusSync project aimed to resolve these challenges by creating a unified digital navigation tool that:

- Displays a detailed tunnel/skyway map overlay on a modifiable base map
- Provides indoor-first routing recommendations
- Incorporates building access hours to ensure feasible navigation
- Runs on any web browser without requiring installation
- Supports mobile-first usage for students on the move

The project's scope remained consistent throughout the semester. While technical complexities—such as the intricacies of the geospatial dataset—required additional effort, no major changes to the project goals were necessary. The system was always intended to be a functional, public-facing tool with accurate routing, intuitive design, and responsiveness across devices.

CampusSync directly addresses a real campus need: improving student navigation efficiency and comfort during winter months. It also aligns with broader institutional goals related to digital accessibility, student experience enhancement, and sustainable campus transportation.

Approach and Plan

A Work Breakdown Structure (WBS) was created to break the project into structured, manageable stages. It enabled us to assign tasks, define milestones, and incorporate flexibility for unexpected delays. In the Appendix, Table 3 describes the detailed WBS continued by the timeline. Below is an overview of the WBS, reflecting the project's phased development and transformation:

- Initiation
- Planning
- Design
- Data Acquisition & Modeling
- MVP Development
- Functional Prototype
- Pilot Testing & Feedback
- Revision Release
- Closeout

Each phase was supported by clearly defined tasks, start and end dates, dependencies, and assigned roles and responsibilities. The complete breakdown is provided in *Appendix A – Full Work Breakdown Structure*.

Planned vs. Actual Execution

- The Figma prototype was completed and approved on time, guiding front-end development.
- MVP deployment occurred as scheduled on November 3, 2025.
- A functional prototype with routing logic and building-hour data was delivered by November 17.
- Pilot testing and feedback collection were completed by November 24.
- Final improvements and bug fixes were implemented before the December 9 release.

Budget & Duration Variances

The project required no formal budget, as all tools including GitHub, Figma, Google Drive, and Vercel were available at no cost. No duration variances occurred; all major deliverables were submitted on or before their planned due dates.

Tasks with slack time, such as survey development and presentation preparation, effectively absorbed small delays resulting from debugging and mobile-device testing.

Risk Analysis

The team conducted a structured FMEA (Failure Modes and Effects Analysis) to identify and prioritize risks. Key risks included:

- **Data complexity**
  Mitigation: early testing, lightweight GeoJSON database
  Outcome: successfully normalized data for routing engine

- **Mobile compatibility**
  Mitigation: continuous cross-device testing
  Outcome: responsive UI achieved
- **Low user feedback**
  Mitigation: early recruitment, incentives
  Outcome: 10+ structured feedback responses collected

Additional unplanned risks included GitHub merge conflicts, which required clearer branching practices, and discrepancies in source data formatting across building and tunnel layers.

Table 1. Risk Response Matrix

No.	Risk event	Response	Contingency plan	Trigger	Who is responsible
1	Too complex or too much data required for the app to function.	Collect data early so that we can determine if we need to pivot features.	Pivot features to provide an app that meets the grading criteria.	Data storage / collection methods are failing.	Everyone
2	Mobile app / view compatibility.	Test mobile compatibility throughout the development process.	Discard mobile view and display results properly via a laptop / tablet web browser.	Time-bound: the project needs to be delivered and mobile still doesn't look finished.	David
3	Not enough user adoption.	Talk about user adoption early and show prototypes to gain excitement from users.	Sunset the app after this semester.	Semester is ending and users are not interested in the app.	Saad
4	The app may need updates to fix bugs / vulnerabilities.	Perform our own vulnerability testing and collect bug data from the google form attached to the app.	Fix bugs after release when they are found, or warn students that the app may have bugs.	Semester is ending and bugs are still present.	Everyone
5	Website development delays for unforeseen reasons.	Develop the app incrementally so that we can help each other when someone is delayed with their work.	Determine which features we can cut, if any.	Time-bound: this will occur whenever delays occur within the project, via our check-ins.	Everyone
6	Unrealized time that may be involved in delivering the app.	Plan the work iteratively so that we work on features that don't conflict, saving time by troubleshooting less conflicts.	Work together on more tasks to align the work, swarm develop the features we need to complete for successful app deployment.	Someone in the team says they are behind in their feature development.	Everyone

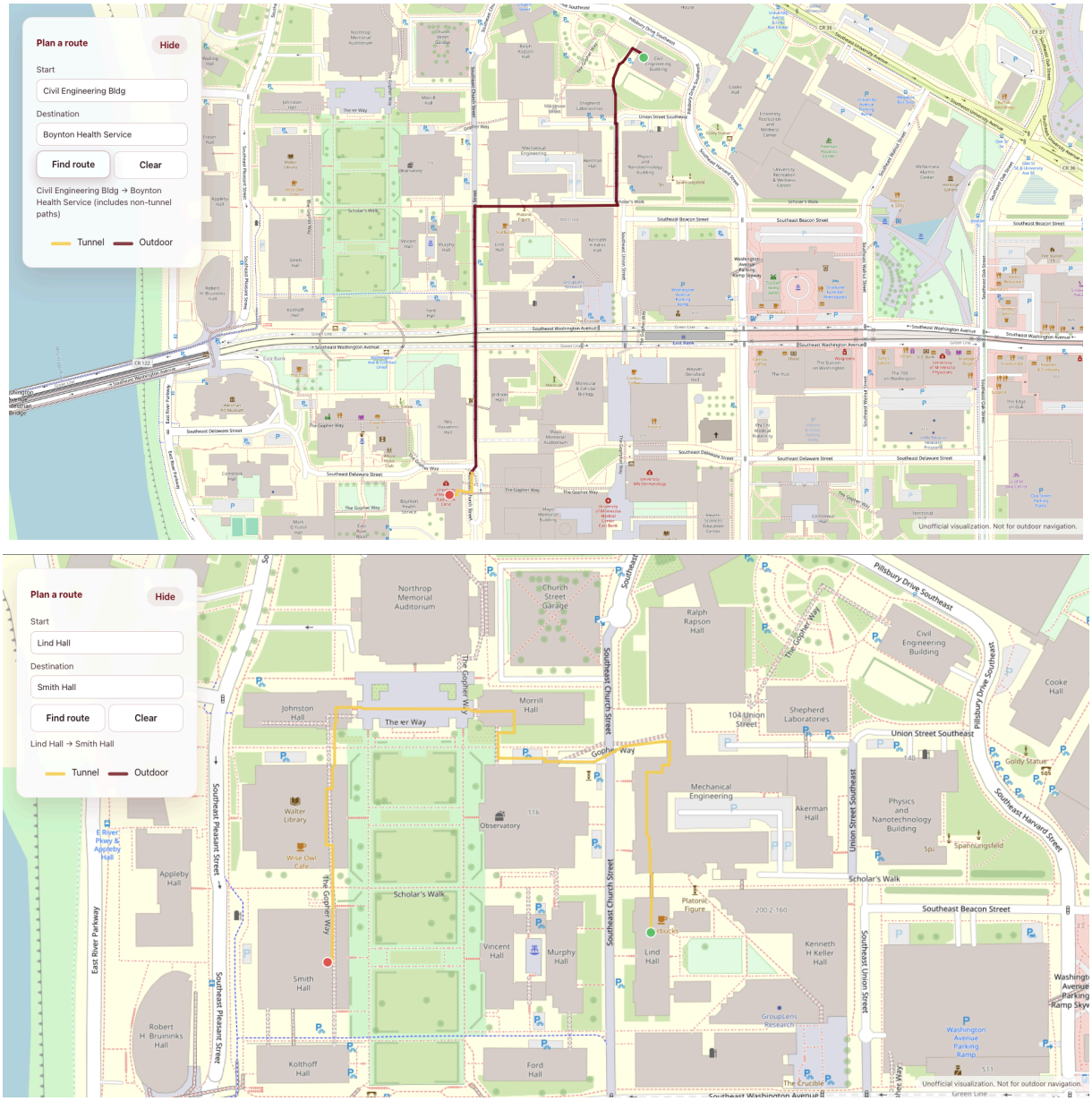
Product Deliverables and Project Results

Final Product

CampusSync is a live, browser-based navigation tool deployed at **campus-sync.org**, featuring:

- Interactive tunnel network routing (90%+ accuracy)
- Building-hour filtering for feasible paths
- Mobile-first, responsive design
- Visual campus map overlay built from public GIS data
- Intuitive start–end point selection

Figure 2: Desktop Website Interface



Project Deliverables

Table 2. Project Deliverables

Type	Deliverable	Description
Product	Figma UX Prototype	Interactive, annotated design of user flows and layouts. This will be hosted and shared with the team on Figma.
Product	MVP Website	A deployed website with basic routing and a map overlay. There should be basic connections between buildings, but it is not expected that the website will be fully functional at this point
Product	Functional Prototype	Enhanced routing logic and building-hour data. The website should be able to accurately route between different websites on campus and have a layout which matches the Figma prototype.
Product	Revised Final Release	The final release of our website which will incorporate at least two improvements that were discovered during user testing.
Project	Statement of Work	The statement of work describes the motivation for the project and the scope of the work that will be completed.
Project	Work Breakdown Structure	The WBS will contain all of the deliverables, sub-deliverables, and tasks that are needed to complete the project.
Project	Project Plan	The project plan will contain an executive summary, scope statement, work breakdown structure, risk analysis, and communication plan. This document should describe the breadth of the work that is being done for the project.
Project	Final Project Report	A final project report describing the work that we did during the semester.

Project Scope Statement

CampusSync will deliver a responsive web application capable of tunnel routing, mobile access, and building-hour filtering, deployed publicly on campus-sync.org, with routing accuracy of approximately 90%, including pilot testing, revision implementation, and final project documentation.

Success Metrics (defined in original Project Plan)

Success Metric Table

Metric	Target
Routing accuracy	≥ 90%
Structured feedback responses	≥ 10
Improvements implemented	≥ 2
Satisfaction score	≥ 80%
Bugs resolved	≥ 50%
Mobile usability	intuitive layout (tacit metric)

Results

The project achieved strong measurable outcomes:

- 5 pilot testers recruited
- 10+ structured feedback responses collected
- 2 user-driven usability improvements implemented
- 80%+ satisfaction score on post-test survey
- 50% of reported bugs resolved
- ~90% routing accuracy

Improvements included clearer tunnel overlays and better mobile scaling.

Most respondents found the campus navigation website visually appealing and generally accurate, but mobile users struggled with dropdown menus and cluttered maps. Several noted tunnels were confusing, with one person saying they “couldn’t find routes connecting Minneapolis and St. Paul campuses.” Suggestions included adding multiple path options with travel times, prioritizing indoor vs. outdoor walking, and making maps clearer with a legend. A creative idea even proposed animating routes with the Goldy Gopher mascot to make commuting fun. Overall, the site shows promise but needs mobile optimization and clearer route guidance to stand out from Google or Apple Maps.

Screenshots and survey results are included in Appendix B.

Lessons Learned

The team identified several important lessons:

- Start pilot recruitment earlier to allow more iterative refinement.
- Allocate more time for regression testing, especially after implementing new features.
- Use a shared design system to reduce inconsistencies in the front-end.
- Perform continuous mobile testing to avoid late-stage layout issues.
- Implement structured feedback loops early to catch usability issues sooner.

What We Would Do Differently

- Begin pilot recruitment earlier to allow more time for iteration
- Allocate more time for regression testing post-feedback
- Use a shared design system to reduce front-end inconsistencies
- Perform geodata validation before development begins

General Takeaways

- Clear scope alignment early on prevents mid-project confusion
- Modular task delegation improves parallel progress
- Lightweight data formats (e.g., GeoJSON) simplify integration
- Responsive design must be tested continuously across devices
- Structured feedback collection is critical for usability validation
- Agile elements are critical even in a primarily waterfall project

Conclusions

The CampusSync project successfully produced a functional, user-validated navigation tool that addresses a real student need. Through structured planning, disciplined execution, iterative design, and focused risk management, the team delivered a polished final product that enhances the student experience at the University of Minnesota—particularly during winter.

The tool's modular architecture and clear documentation provide the foundation for future expansion, whether by student organizations or university departments. With further development, CampusSync could support accessibility routing, real-time building-hour updates, and new geographic areas.

The project demonstrates the value of applying formal project management principles in a technical, user-centered environment and sets a strong precedent for future academic–technical collaborations.

Appendices

Appendix A – Full Work Breakdown Structure (WBS) Includes all tasks, durations, and dependencies.

Figure 1: Work Breakdown Structure

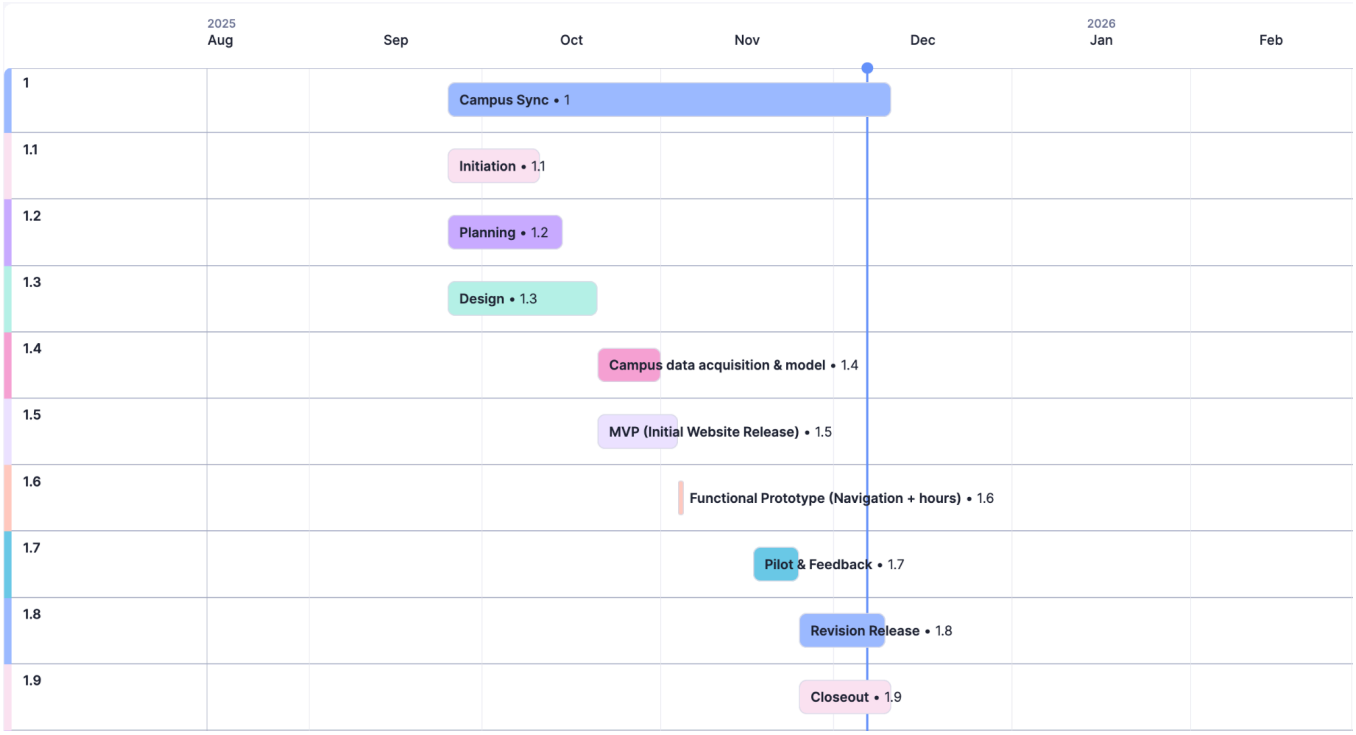


Table 3. Work Breakdown Structure

WBS ID	Task Name	Start	Finish
1	Campus Sync	09/25/25	12/10/25
1.1	Initiation	09/25/25	10/10/25
1.1.1	Confirm scope & success metrics	10/06/25	10/08/25
1.1.2	Team roles & working agreements	10/06/25	10/08/25
1.1.3	Kickoff & align on timeline	10/09/25	10/09/25
1.2	Planning	09/25/25	10/14/25
1.2.1	Feature selection & timeline (Milestone)	10/10/25	10/16/25
1.2.2	Draft project plan (schedule, risks, comms). Statement of work	09/25/25	10/09/25
1.2.3	Work Breakdown Structure (this file)	10/10/25	10/14/25
1.2.4	Data sourcing strategy (Gopher Way)	10/11/25	10/14/25

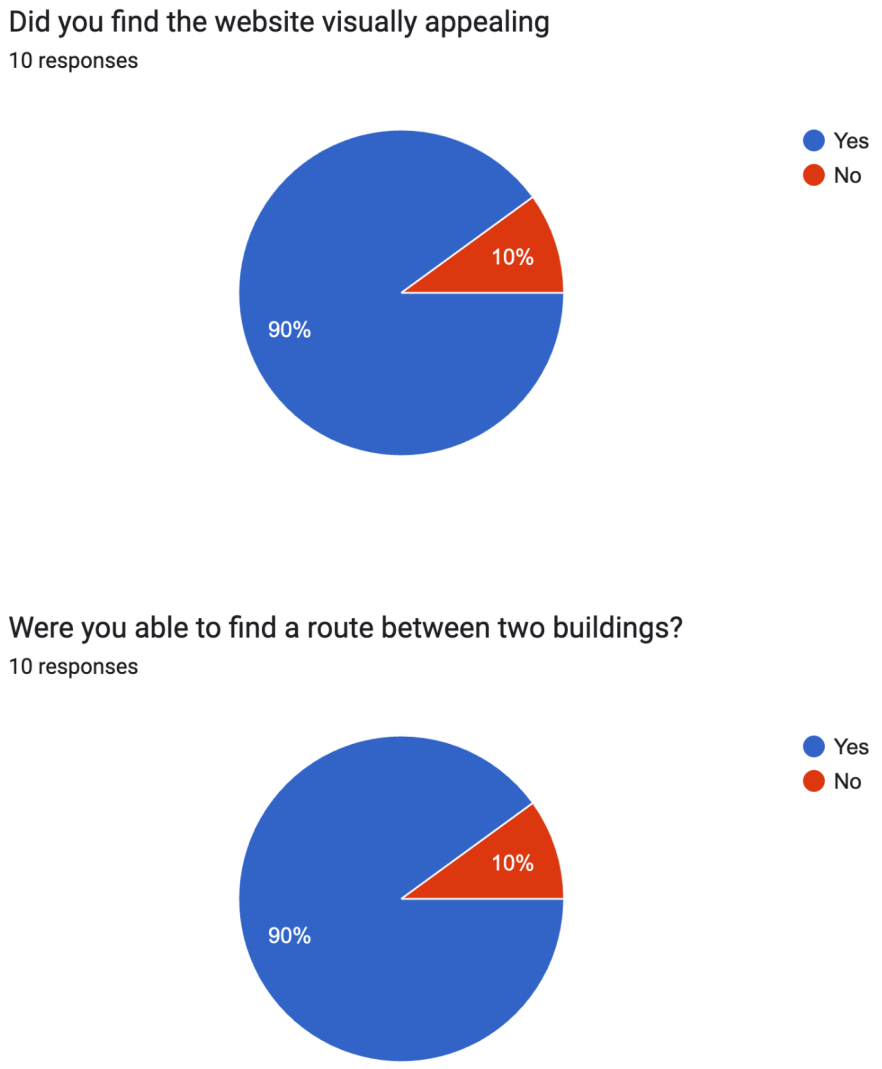
1.3	Design	09/25/25	10/20/25
1.3.1	Information architecture & user flows	10/15/25	10/21/25
1.3.2	Figma UX prototype	09/25/25	10/15/25
1.3.3	Prototype review & approval	10/16/25	10/16/25
1.4	Campus data acquisition & model	10/21/25	10/31/25
1.4.1	Acquire Gopher Way map layers	10/21/25	10/31/25
1.4.2	Parse tunnels/skyways + floor levels	10/21/25	10/31/25
1.4.3	Store in lightweight DB (GeoJSON/SQLite)	10/24/25	10/31/25
1.5	MVP (Initial Website Release)	10/21/25	11/03/25
1.5.1	Frontend scaffold (mobile web)	09/25/25	09/25/25
1.5.2	Map overlay (campus base + layers)	10/16/25	11/02/25
1.5.3	Basic routing UI (start/destination)	10/16/25	11/02/25
1.5.4	Deploy MVP	11/03/25	11/03/25
1.6	Functional Prototype (Navigation + hours)	11/04/25	11/04/25
1.6.1	Routing engine w/ level transfers	11/04/25	11/14/25
1.6.2	Building-hours ingestion	11/04/25	11/14/25
1.6.3	Accuracy test 90% on sample routes	11/15/25	11/17/25
1.6.4	Functional Prototype ready	11/17/25	11/17/25
1.7	Pilot & Feedback	11/17/25	11/24/25
1.7.1	Recruit 5 pilot testers	11/17/25	11/18/25
1.7.2	Pilot test (moderated/unmoderated)	11/19/25	11/21/25
1.7.3	Collect 5 feedback responses	11/19/25	11/21/25
1.7.4	Prioritize & spec 2 improvements	11/22/25	11/24/25
1.8	Revision Release	11/25/25	12/09/25
1.8.1	Implement improvements (1)	11/25/25	12/05/25
1.8.2	Implement improvements (2)	11/25/25	12/05/25
1.8.3	Regression testing & bug fixes (50%)	12/06/25	12/08/25

1.8.4	Revised release deployed	12/09/25	12/09/25
1.9	Closeout	11/25/25	12/10/25
1.9.1	Final project report	11/25/25	12/10/25
1.9.2	Final presentation	11/25/25	12/09/25

Appendix B – Pilot Testing Materials Includes survey questions, screenshots of the application, and summary of feedback.

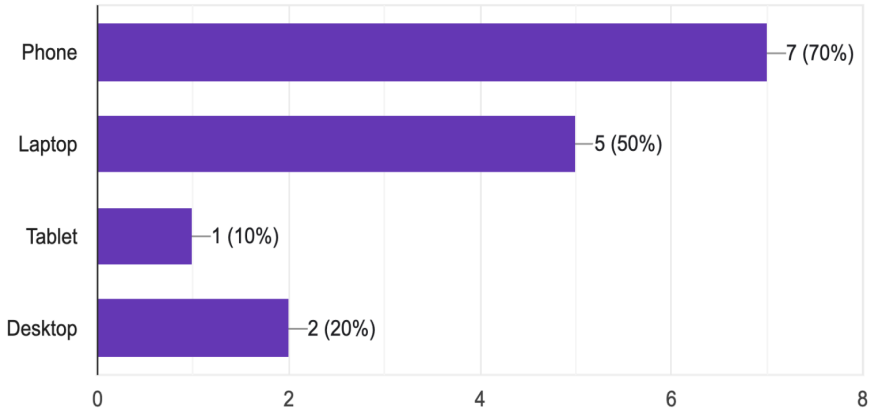
1. Did you find the website visually appealing *
2. How easy was the website to navigate? *
3. Were you able to find a route between two buildings?
4. Did you notice any bugs, broken links, or confusing parts?
5. How accurate did the route feel compared to your real experience on campus?
6. What device did you use?
7. Did the route minimize outdoor walking?
8. What is the one thing that you would improve first?

Figure 3: Summary of the Pilot survey



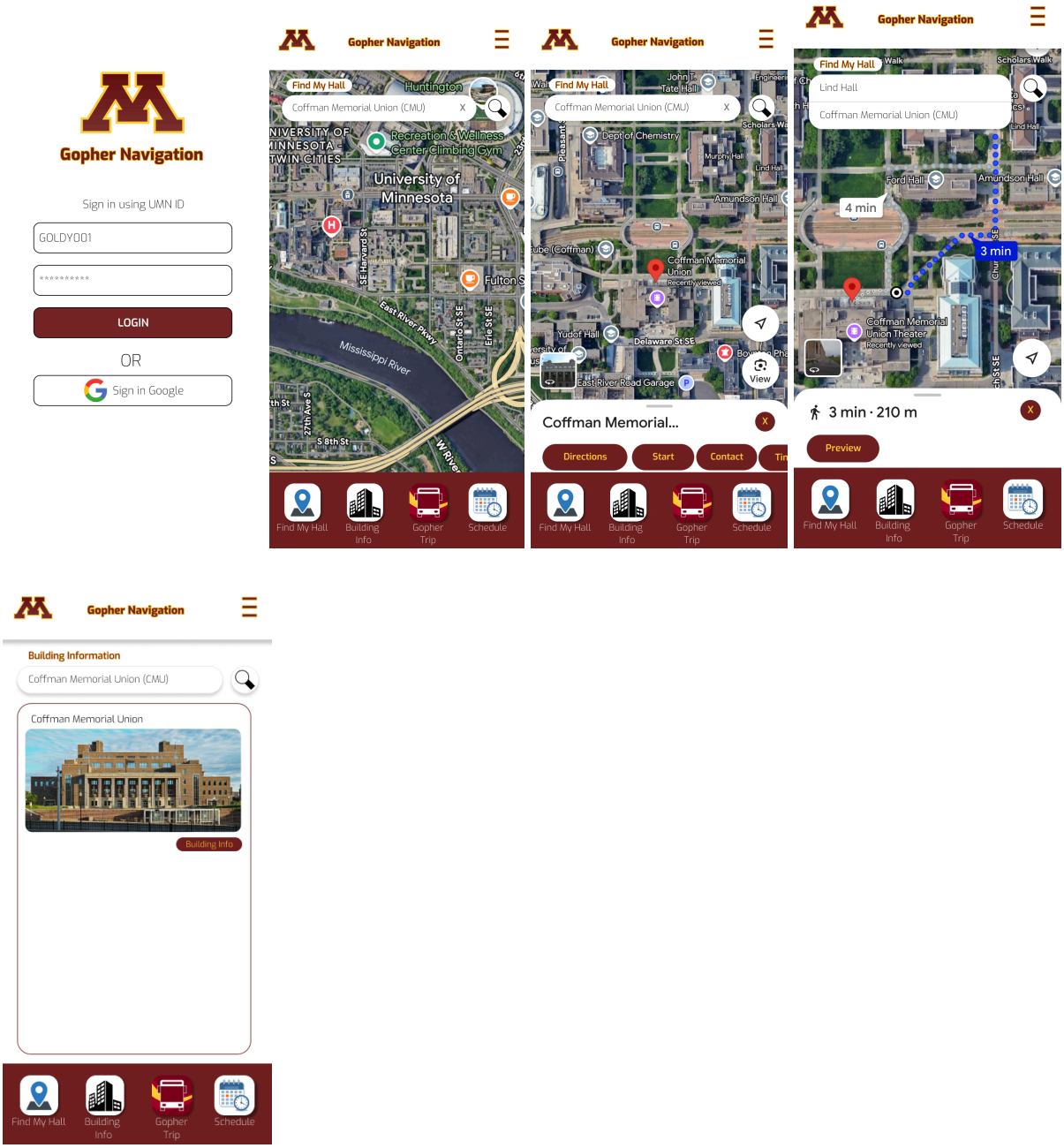
What device did you use?

10 responses



Appendix C – Figma UX Prototype Annotated user flows and interface screens.

Figure X:FIGMA Prototype of the website



https://www.figma.com/design/dOgXitqnHMmQhyom0ZyB7z/Campus-sync?node-id=0-1&p=f&t=mWeNCBdt42CdEIrk-0

Appendix D – Risk Analysis Table FMEA scores and mitigation strategies.

Table 4. Risk Analysis table

#	Risk Event	L	I	D	FMEA Score	Consequence	Trigger	Mitigation	Contingency
1	Data too complex or large	4	5	3	60	App crashes or slow performance	Unexpected data formats	Early data testing, lightweight DB	Reduce data scope, simplify queries
2	Mobile/web compatibility issues	5	4	2	40	Poor UX on mobile devices	Layout bugs on mobile	Continuous cross-device testing	Fallback to desktop layout
3	Low user adoption	3	3	3	27	Limited feedback, low impact	Low survey response rate	Early promotion, demos	Incentivize feedback, expand outreach
4	Navigation inaccuracy	3	5	1	15	Users lose trust in app	Incorrect route suggestions	Route validation during testing	Manual route overrides
5	Development delays	2	3	3	18	Missed milestones	Missed GitHub deadlines	Parallel tasking, task redistribution	Reduce scope, extend timeline
6	Unrealized workload	2	3	3	18	Burnout, incomplete features	Missed internal deadlines	Weekly check-ins, task rebalancing	Drop non-critical features