Sprint 1 retrospective

Section: R

SOEN 390
Winter 2025

Team: Bulat & friends

Presented by:

Azmi Abidi - 40248132
Nektarios Zampetoulakis - 40211948
Alimurat Dinchdonmez - 40245310
Ryad Alla - 40227731
Marc-Yves Malchev - 40265238
Bulat Abdullin - 40264963
Alessandro Tiseo - 40262416
Joseph Aladas - 40156616
Abdullah Taha - 40261146

I certify that this submission is my original work and meets the Faculty's Expectations of originality.

Azmi Abidi, Nektarios Zampetoulakis, Alimurat Dinchdonmez, Ryad Alla , Marc-Yves Malchev, Bulat Abdullin, Alessandro Tiseo, Joseph Aladas , Abdullah Taha

January 26, 2025

1. Introduction

Overview of the Project:
The **Concordia Campus Guide** is a mobile application designed to assist students, faculty, and visitors in navigating Concordia University's campuses efficiently. Built mainly with React Native, the app integrates outdoor and indoor navigation, real-time building information, and personalized features to enhance campus exploration and accessibility.

Purpose and Motivation:
Navigating university campuses, particularly large ones like Concordia's SGW and Loyola campuses, can be challenging, especially for new students, visitors, or those with accessibility needs. The primary motivation behind this project is to:

- Simplify navigation by providing clear directions for buildings, classrooms, and other points of interest.
- Offer real-time information, such as shuttle schedules and class locations.
- Enhance campus accessibility for all users, including those with disabilities.
- Create a smart and modern tool that connects with user calendars for seamless planning.

Targeted Users:

- **Students**: To help them locate classrooms, libraries, and other campus facilities efficiently.
- **Faculty and Staff**: For seamless navigation between offices, departments, and meeting rooms.
- **Visitors**: To explore the campus without prior knowledge of the layout.
- **Users with Accessibility Needs**: Special focus on wheelchair-accessible routes, elevators, and other essential facilities.

2. Requirements

Backlog

ID	Name	USP	Priority
#13	As a user I want user stories to be defined.	5	1
#12	As a user I want the design of the app to be defined (basic domain model, component diagram)	21	2
#11	As a user I want UI mockups to be created for user stories that will be implemented in sprint 2.	13	2
#10	As a user I want the personas to be defined to represent potential users and describe some of the tasks they will perform on the system.	8	2
#9	As a user I want the project backlog to be created in Zenhub.	3	1
#8	As a user I want the GitHub repository and yml scripts for GitHub workflows to be set and defined.	13	1
#14	As a user I want the project to be extensively documented.	5	3
Total		68	

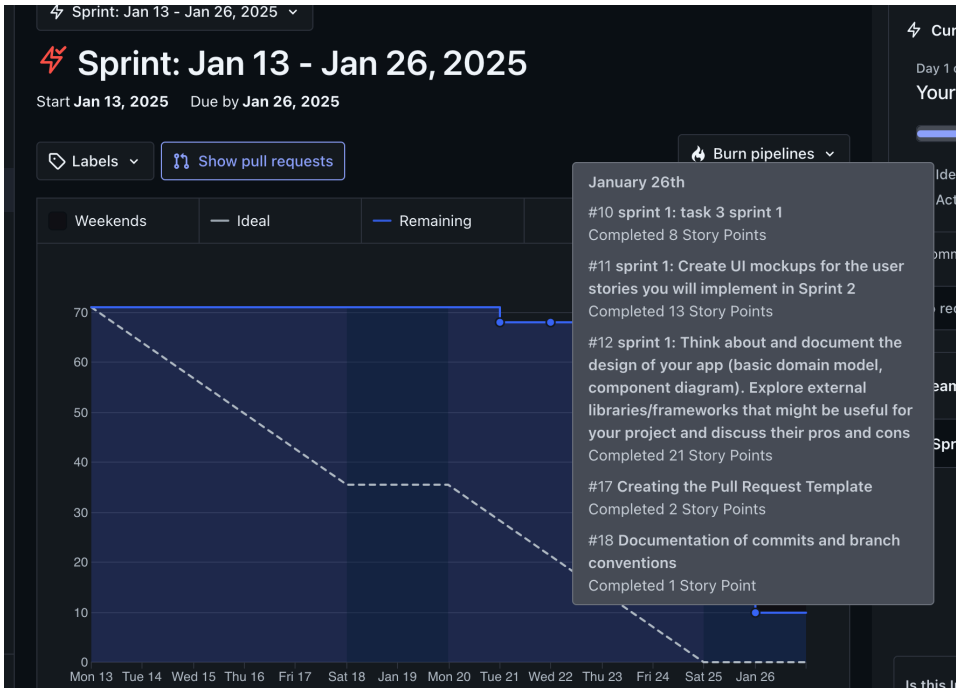
3. Release Planning

Sprint 1 Summary

This sprint focused on laying the groundwork for the project by completing essential setup tasks and delivering the foundational features of the app. The key objectives for this sprint included establishing the development environment, creating initial UI mockups and documenting the project requirements.

Story ID		Planned USP	Status
#13		5	**DONE**
#12		21	**DONE**
#11		13	**DONE**
#10		8	**DONE**
#9		3	**DONE**
#8		13	**DONE**
#14		5	**DONE**
Total		**68**	**68**

Project velocity after 1st sprint:



Sprint 1 Retrospective:

Keep doing: Daily scrums and short meetings.

Start doing: Sharing all our different opinions.

Stop doing: Speaking all at the same time. Coming late to the meetings.

Do more of: Commenting on GitHub pull requests.

Do less of: Making the reviews of Pull Request in Discord chat.

4. Architecture

4.1 Overview of the architecture

The **Concordia Campus Guide** adopts a **Client-Server Architecture** and an **MVC Architecture** with a **Component-Based Design**. This style is chosen to ensure modularity, scalability, and maintainability while providing an efficient and seamless user experience across different devices.

1. **Client-Server Architecture**:

In this architecture, the mobile app (client) interacts with a server to retrieve and update data.

This ensures real-time data synchronization and allows the client to remain lightweight while the server handles heavy computations and database interactions.

2. **Component-Based Design**:

The app will be structured into reusable components for UI, business logic, and data handling.

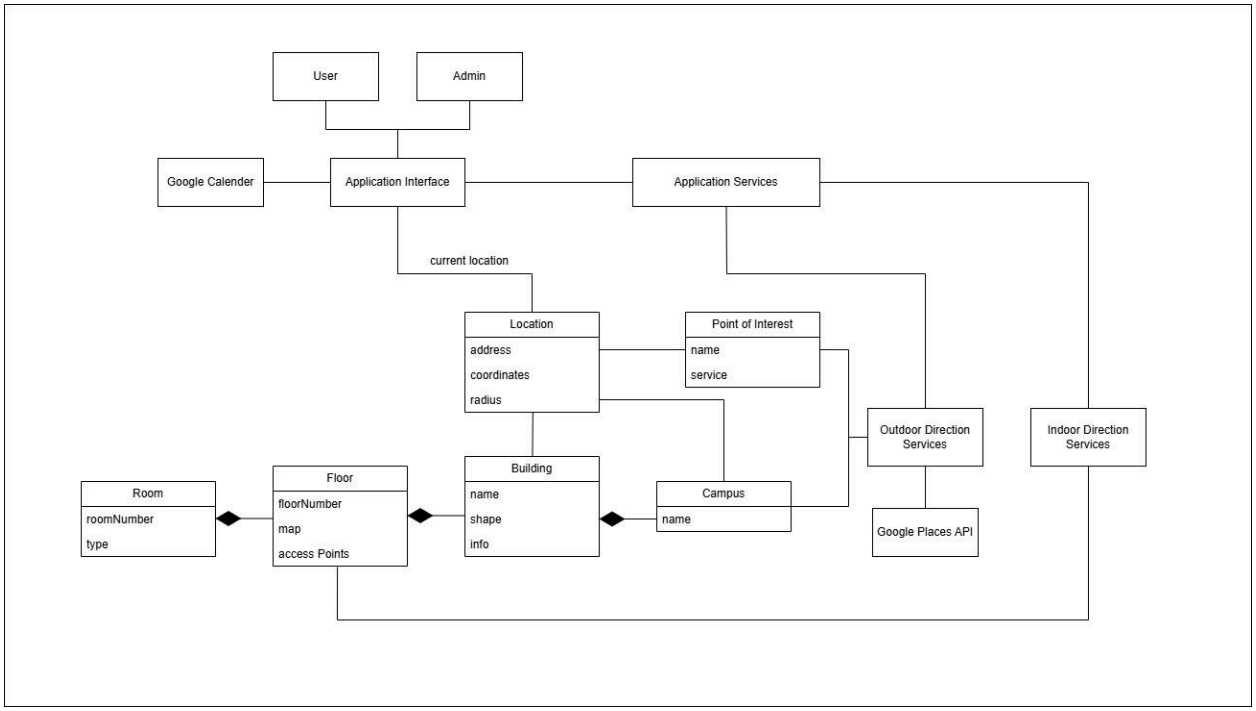
This enhances code reusability (e.g., reusable components for buttons, modals, and input forms) and simplifies development and testing by isolating components.

3. **Model-View-Controller (MVC)**:

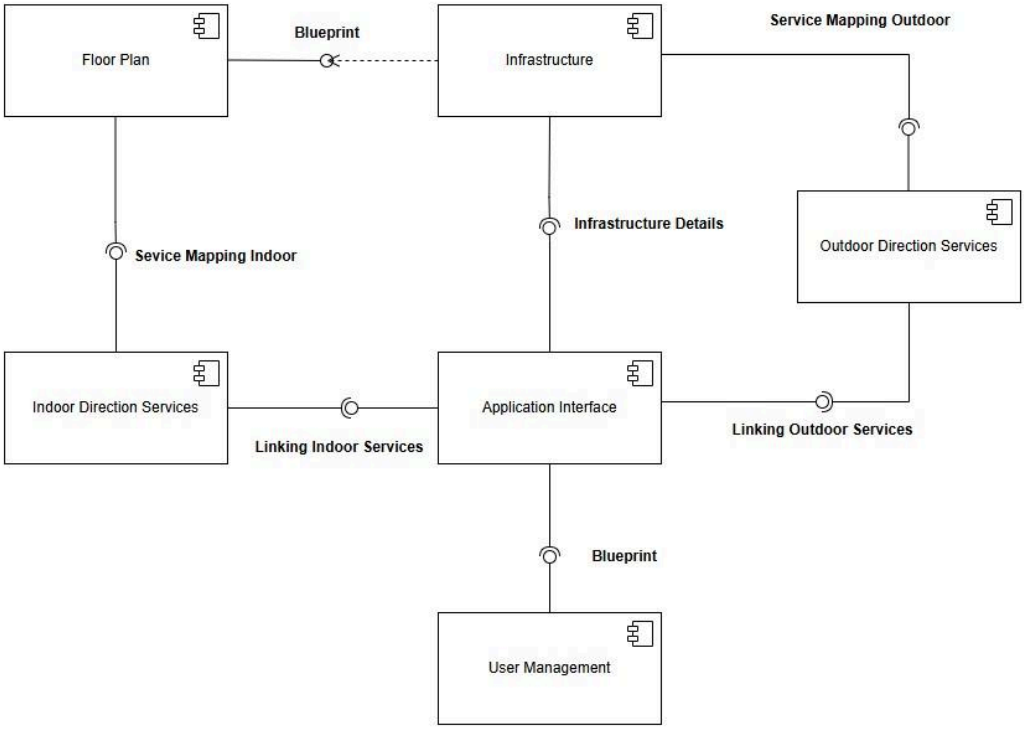
The app loosely follows the MVC pattern to separate concerns:

- **Model**: Manages data (e.g., user information, building details).
- **View**: Handles the UI (React Native components).
- **Controller**: Acts as the intermediary, managing logic and API calls.

4.2 Domain Model



4.3 Component Diagram



4.4 Design Patterns

The app incorporates several **Gang of Four (GoF)** design patterns, along with framework-specific patterns, to promote modularity, scalability, and maintainability. These patterns ensure the system can evolve efficiently as new requirements are introduced.

1. Singleton Pattern

The Singleton Pattern is implemented to ensure a single instance of a class, such as managing application-wide data like user sessions or API clients. For instance, a SessionManager ensures that only one session is maintained for user authentication and API requests throughout the app. This is implemented using React's Context API to manage global state effectively. By centralizing configuration and state management, the pattern ensures consistent behavior across the app and avoids issues caused by multiple instances.

2. Factory Pattern

The Factory Pattern is used to abstract the creation of complex objects, such as navigation flows, based on input parameters. For example, a NavigationFactory generates navigation routes for various scenarios, including outdoor navigation, indoor navigation, or inter-campus navigation. This abstraction simplifies the creation process while hiding the instantiation details, making the system flexible and maintainable. It also makes it easy to introduce new navigation types, such as accessibility-specific routes, by extending the factory to support additional modes without modifying existing code.

3. Strategy Pattern

The Strategy Pattern is employed to define a family of algorithms, encapsulate them, and make them interchangeable, such as in selecting transportation modes for outdoor navigation. Each transportation mode, like walking, driving, or shuttle, is encapsulated as a separate strategy, allowing seamless switching based on user preferences or environmental factors. This modular design promotes clean and flexible code, making it easier to add new strategies, such as biking routes or eco-friendly paths, without altering the existing logic.

4. Framework-Specific Patterns

The Component Pattern, a fundamental aspect of React Native, is used to encapsulate UI elements into reusable, self-contained components. Examples include buttons, modals, and map markers, which are designed to be modular and easy to test. This approach ensures that the codebase remains clean and adheres to the DRY (Don't Repeat Yourself) principle, enabling rapid development and straightforward scaling of the app's interface as new features are introduced.

5. User Interface Design

5.1 Personas

All the user personas defined in this sprint can be found at the following LINK.

5.2 Equity

The **Concordia Campus Guide** prioritizes accessibility and equity by incorporating features that support individuals with disabilities, illiterate users, and marginalized groups. To assist people with disabilities, the app provides high-contrast visuals for color-blind users, screen reader compatibility for blind users, visual and text-based alternatives for deaf users, and voice command support for those with hand mobility challenges. For illiterate users, the app emphasizes intuitive icons, images, and audio-based instructions to simplify navigation. Marginalized groups benefit from offline functionality, lightweight design for low-end devices, and culturally inclusive content. Compliance with accessibility standards such as WCAG and regular usability testing ensure the app remains accessible and equitable for all.

5.3 Mockups

All the UI mockups related to user stories for sprint 2 can be found at the following Figma LINK.

6. Testing Plan and Report

6.1 Unit Testing

Run tests

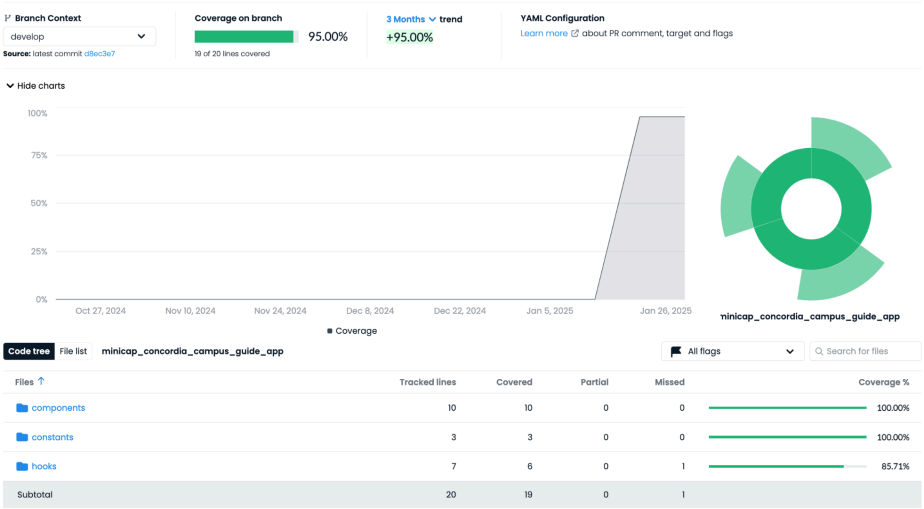
```
▶ Run npm run test -- --coverage

> minicap_concordia_campus_guide_app@1.0.0 test
> jest --ci --coverage

PASS components/__tests__/ThemedText-test.tsx
  ✓ renders correctly (420 ms)

--------------------|---------|----------|---------|---------|-------------------
File                | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
--------------------|---------|----------|---------|---------|-------------------
All files           |    90.9 |    53.33 |     100 |    90.9 |
 components         |     100 |    54.54 |     100 |     100 |
  ThemedText.tsx    |     100 |    54.54 |     100 |     100 | 24-28
 constants          |     100 |      100 |     100 |     100 |
  Colors.ts         |     100 |      100 |     100 |     100 |
 hooks              |      80 |       50 |     100 |      80 |
  useColorScheme.ts |       0 |        0 |       0 |       0 |
  useThemeColor.ts  |      80 |       50 |     100 |      80 | 17
--------------------|---------|----------|---------|---------|-------------------
Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   1 passed, 1 total
Time:        2.11 s
Ran all test suites.
```

6.2 Test code coverage



7. Code review report

Here is our top 5 pull requests for Sprint 1:
- https://github.com/AlimuratDinch/minicap_concordia_campus_guide_app/pull/35

- https://github.com/AlimuratDinch/minicap_concordia_campus_guide_app/pull/32

- https://github.com/AlimuratDinch/minicap_concordia_campus_guide_app/pull/34

- https://github.com/AlimuratDinch/minicap_concordia_campus_guide_app/pull/33

- https://github.com/AlimuratDinch/minicap_concordia_campus_guide_app/pull/30

8. Professional ethics and accountability

The **Concordia Campus Guide** is generally designed to have a positive impact by improving campus navigation and accessibility. However, ethical concerns could arise in areas such as data privacy, equity of access, and reliance on technology. The app collects user data, such as location, which raises concerns about privacy and security. To address this, the app follows strict data protection practices. Reliance on technology can pose challenges if users overly depend on the app and face disruptions during technical issues. To address this, the app includes fail-safe features, such as printable directions or emergency contact options, ensuring it supports users even in challenging scenarios. These measures demonstrate a commitment to ethical design and minimizing potential negative impacts.