CSCI 5708 - MOBILE COMPUTING

Fall 2023

PROJECT REPORT - ”MyCampus”

Submitted by

Name	**BannerID**	**Email**
Pratik Mukund Parmar	B00934515	pratikparmar@dal.ca
Badhri Arja	B00930493	bd861158@dal.ca
Naveen Kunapaneni	B00930132	nv676973@dal.ca
Lokeshwar Kumar Tabjula	B00936909	lokeshwartabjula@dal.ca
Kovarthanan Murugan	B00936880	kovarthanan.murugan@dal.ca
Kishoreganesh Sundararajan	B00934548	kishoreganesh@dal.ca
Emayan Vadivel	B00934556	em241005@dal.ca

Under the Guidance of

PROF. SHEHZEEN
Faculty of Computer Science

December 01, 2023

Contents

1 Project Description: **2**
1.1 Project Title 2
1.2 Project Contributors and Contact Information 2
1.3 Abstract 2

2 Introduction **3**
2.1 About the project 3
2.2 Need for this project 3
2.3 Intention towards the customers 3
2.4 Comparison with similar applications 3

3 Lessons **4**
3.1 Learnings from the project 4

4 Product **6**
4.1 Purpose 6
4.2 Why us? 6
4.3 Where and how will this application be used? 6
4.4 Constraints Impacting the Application 7
4.5 User's State of Mind While Using the Application and How It Will Affect Use: 7
4.6 Application Functionalities 7
4.7 Used Tools 8
4.8 What did you create? 8
4.9 How much minimum, desired, and bonus functionality did you implement? 8
4.10 What does your app actually do? 8
4.11 How does it differ from what you planned? 8
4.12 Planning complexity 9

5 Issues **10**
5.1 Challenges Encountered 10

6 Satisfaction **12**
6.1 What are you happy with? 12
6.2 Areas of Unhappiness 12
6.3 Did you learn a lot 12
6.4 Did you have insight into development? 12
6.5 Did you make friends? 12
6.6 QA 13
6.7 Task Distribution 13

7 Demonstration **14**
7.1 Screenshots 14

8 Conclusion **39**

1 Project Description:

1.1 Project Title

The title of our project is ”**MyCampus**” - Empowering Student Safety and Convenience. The MyCampus mobile application is designed to serve as an all-encompassing platform for university students, offering a range of functionalities aimed at enhancing safety and providing essential utilities within and around the campus. This report outlines the detailed functionalities of the application, along with corresponding use cases.

1.2 Project Contributors and Contact Information

Name	BannerID	Email
Pratik Mukund Parmar	B00934515	pratikparmar@dal.ca
Badhri Arja	B00930493	bd861158@dal.ca
Naveen Kunapaneni	B00930132	nv676973@dal.ca
Lokeshwar Kumar Tabjula	B00936909	lokeshwartabjula@dal.ca
Kovarthanan Murugan	B00936880	kovarthanan.murugan@dal.ca
Kishoreganesh Sundararajan	B00934548	kishoreganesh@dal.ca
Emayan Vadivel	B00934556	em241005@dal.ca

1.3 Abstract

”MyCampus” - Empowering Student Safety and Convenience

MyCampus is an innovative Android mobile application designed to cater to the safety and convenience needs of university students. This comprehensive platform offers a plethora of features, including emergency contacts, a panic button, location sharing, incident reporting, safe walking routes, campus maps, notification systems, and more. By conducting extensive user research and employing design thinking, we have developed a user-centric app that not only enhances safety and security but also streamlines daily campus life. With the integration of third-party APIs, such as weather alerts and transportation information, MyCampus ensures students are well-prepared for their university experience. This proposal delves into the project’s background, target users, purpose, benefits, functionalities, architecture, tools, and testing plans.

2 Introduction

2.1 About the project

The ”MyCampus” project is a comprehensive Android mobile application tailored to address the unique safety and utility requirements of university students. It offers a consolidated solution by amalgamating various essential features, such as emergency contacts, incident reporting, location sharing, campus maps, and more. MyCampus aims to provide a single point of access for students to enhance their safety and streamline daily campus life, ultimately simplifying their university experience.

2.2 Need for this project

This project is indispensable due to the intricacies of university life, which demand a versatile and comprehensive solution to cater to the diverse needs of students, encompassing both safety and convenience. What makes this endeavour particularly intriguing is its capacity to bridge the prevailing gap between numerous disparate standalone applications. MyCampus aspires to be the ultimate all-in-one, user-centric platform, dedicated to simplifying students’ daily lives on campus. By integrating essential safety and utility features into a single application, it strives to enhance the overall university experience, providing students with a seamless, efficient, and reliable solution for their diverse campus-related requirements.

2.3 Intention towards the customers

Our foremost aim is to provide university students with an intuitive mobile application that not only elevates their safety but also streamlines their day-to-day campus experience. Our aspiration is to deliver a seamless and frictionless user experience, making access to safety and utility features as effortless as a simple tap. MyCampus is meticulously designed to be a reliable companion, equipping students with essential resources and fostering a strong sense of community within the university. It’s not just an app; it’s a dependable ally that enriches the university experience by prioritizing the convenience, security, and overall well-being of its users.

2.4 Comparison with similar applications

”MyCampus” distinguishes itself from similar applications by offering a unique set of features tailored to the specific needs of university students. While existing apps may provide isolated features, MyCampus combines a comprehensive array of functionalities that encompass safety, convenience, and effective communication within the academic environment. Key features such as the Panic Button, Incident Reporting, and Safe Walking Routes emphasize campus safety and swift response to emergencies. The Campus Map goes beyond conventional mapping apps, offering detailed location information, including washrooms, vending machines, and garbage cans. Additionally, MyCampus integrates a Notification System dedicated to university-wide announcements, creating a direct channel for campus-wide communication. It even incorporates a Vending Machine Locator, a feature often overlooked by other applications. The app’s focus on student well-being is evident through its provision of Safety Tips and Resources, making it a reliable source of guidance for students. The integration of a third-party Weather API ensures students are well-prepared for classes by delivering timely weather alerts.

Furthermore, the Campus Events Calendar streamlines event management by providing important academic dates and university events. For commuting students, MyCampus offers Transportation Information, including popular bus routes and modes of transport, addressing specific university transportation needs. To promote inclusivity, the Accessibility Request feature allows users to raise concerns and requests for assistance, fostering an inclusive campus environment. Moreover, the app facilitates direct communication with the administration for support and suggestions through its Contact Us feature, streamlining user feedback. Notably, MyCampus goes the extra mile with a Waste Management feature, offering instructions for proper waste disposal and providing utilities and waste management location information, a unique and practical addition.

3 Lessons

3.1 Learnings from the project

Teamwork and Collaboration:

- **Communication:** Effective communication involves active listening, clarity in conveying ideas, and prompt responsiveness. Establishing clear channels for communication (e.g., Slack, regular meetings) fosters a collaborative environment.
- **Collaboration:** Leveraging each team member's strengths, sharing responsibilities, and respecting diverse opinions lead to innovative solutions. Embracing teamwork promotes a sense of unity and collective ownership of the project's success.

Version Control and Repository Systems:

- **Git and Version Control:** Understanding Git fundamentals, such as branching strategies, pull requests, and code reviews, ensures a structured approach to managing code changes. Utilizing platforms like GitHub or GitLab facilitates efficient collaboration and code review processes.

Permission Management and Security in Android Development:

- Through the challenges faced while implementing the SOS feature, we gained valuable insights into permission management and security considerations in Android development. Understanding the importance of specific permissions, such as the **CALL_PHONE permission**, not only resolved immediate issues but also highlighted the critical role permissions play in ensuring the security and functionality of app features. This learning emphasized the need for a comprehensive approach to permissions to enhance user experience and maintain the integrity of security-sensitive features.

Effective Schema Evolution Strategies with RoomDB:

- Dealing with data integrity issues during schema changes in RoomDB provided us with a deep understanding of effective strategies for schema evolution. Learning to navigate and manage changes in the database schema, including the use of versioning and uninstallation strategies, contributed to a more robust and adaptable data persistence layer. This technical learning is valuable not only for the current project but also lays a foundation for handling future enhancements and updates, ensuring a seamless evolution of the application's data structure.

Scheduling and Planning:

- **Project Management:** Utilizing project management tools (e.g., Jira, Trello) or methodologies like Scrum or Kanban aids in organizing tasks, setting milestones, and tracking progress. Regularly reassessing timelines and adjusting plans based on feedback ensures project alignment with goals.
- **Adaptability:** Embracing change requests or adapting to unforeseen challenges demands flexibility in planning. Iterative development cycles enable quick adjustments to meet evolving project needs.

Problem-solving and Adaptability:

- **Debugging and Troubleshooting:** Developing a structured approach to debug issues, utilizing logging, and conducting systematic tests aid in identifying and resolving problems efficiently. Collaborative problem-solving sessions encourage knowledge sharing and faster issue resolution.
- **Adaptability:** Being open to pivoting strategies or technologies based on project requirements or market demands fosters agility. Flexibility allows for quick adaptation and implementation of changes.

Quality Assurance and Testing:

- **Testing Practices:** Incorporating various testing methodologies (unit tests, integration tests, user acceptance tests) ensures the application's reliability and functionality. Iterative testing cycles identify and rectify issues early in the development phase.

User-Centric Approach:

- **UX and Feedback:** Prioritizing user experience by incorporating user feedback into the development process ensures the application aligns with user expectations. Conducting usability testing and gathering user insights are crucial for refining the app's usability.

Continuous Learning:

- **Skill Enhancement:** Encouraging a culture of continuous learning, attending workshops, webinars, or training sessions keeps the team updated on emerging technologies and industry best practices. Knowledge sharing among team members enriches the collective skill set.

Time Management:

- **Task Prioritization:** Efficiently managing tasks by setting priorities and breaking down complex tasks into manageable chunks enhances productivity. Adhering to deadlines helps maintain project momentum and avoids delays.

Documentation and Knowledge Sharing:

- **Comprehensive Documentation:** Creating well-structured documentation for code, system architecture, and project decisions facilitates knowledge transfer and onboarding of new team members. Documenting processes and lessons learned aids in future project improvements.

Embracing Challenges:

- **Resilience and Growth:** Encountering challenges as opportunities for growth and improvement cultivates resilience. Embracing a positive mindset toward setbacks encourages creativity and innovation in problem-solving.

These lessons collectively shape a holistic understanding of software development, emphasizing the importance of not only technical skills but also soft skills, adaptability, and a collaborative mindset in delivering successful projects.

4 Product

4.1 Purpose

MyCampus serves a larger purpose of improving the overall campus experience for all students and faculty. It aims to create a safer and more convenient environment for everyone on campus, fostering a sense of community and enhancing the university experience as a whole. At an individual level, the app provides personalized features to help students and faculty in their day-to-day lives, making it easier to navigate the campus, communicate with peers, and address personal safety concerns.

In developing the ”MyCampus” application, we meticulously consider two types of context: the **”Big-C” context**, which delves into the ”what,” focusing on user benefits and their mental model, ultimately enhancing their overall experience, and the **”Small-C” context**, which is concerned with the ”where” and ”how,” encompassing the mode, medium, and environment in which the application operates [4].

”MyCampus” adapts to the campus environment, catering to diverse modes and mediums. It operates on Android smartphones, seamlessly integrating into the campus experience, considering transportation, connectivity, and high-stress situations. In essence, ”MyCampus” excels in user-centric enhancements while seamlessly navigating the campus’s dynamic realm [4].

4.2 Why us?

1. **Safety:** The app places a strong emphasis on safety by providing features such as the Panic Button, Emergency Contacts, and Incident Reporting. In stressful or emergency situations, students can quickly reach out for help through the app. This significantly enhances their sense of security while on campus.

2. **Convenience:** MyCampus simplifies and streamlines various aspects of daily campus life. It offers functionalities such as locating vending machines, identifying safe walking routes, exploring detailed campus maps, and receiving important notifications for class updates and weather alerts. These features make everyday tasks more convenient, saving time and effort.

3. **Community:** MyCampus promotes a sense of community by enabling students to connect with peers, faculty, and the university administration. Even during stressful times or emergencies, the app provides a means for users to reach out and support one another, fostering a strong sense of belonging within the campus community.

4.3 Where and how will this application be used?

MyCampus is designed to be primarily used within the university campus environment. Users will access the application on their Android smartphones. Here are some common scenarios for the application’s use:

- **Classrooms:** Students can use the app to check for notifications related to class updates. It ensures that they receive important academic information promptly and can navigate the campus effectively.

- **Late-Night Study Sessions:** During late-night study sessions or extracurricular activities, students can benefit from the app’s features. They can share their real-time location with friends for safety, access safe walking routes for secure journeys, and utilize the app’s notification system for event updates.

- **Extracurricular Activities and Events:** While participating in extracurricular activities or events, students can benefit from instant notifications and access to emergency contact features. This ensures that they stay informed and connected during such activities.

4.4 Constraints Impacting the Application

Several constraints can impact the app's performance and functionality:

- **Connectivity:** MyCampus may require an internet connection for certain features, such as location sharing and weather alerts. Limited or unreliable connectivity may affect the real-time functionality of these features.
- **Location of Use:** The app is primarily designed for use within the confines of the university campus. While it offers utility and safety within this area, its effectiveness may decrease when used outside this specific environment.
- **Memory Needs:** Running the app and maintaining its functionalities may consume device memory. Users with limited available memory may experience slower performance or potential issues with app functionality.

4.5 User's State of Mind While Using the Application and How It Will Affect Use:

Users of MyCampus may find themselves in various states of mind while using the application:

- **Distracted:** During routine use, students may be distracted by various tasks such as studying, walking between classes, or attending lectures. In such scenarios, it's crucial that the app is designed with simplicity and efficiency in mind. Important features, especially safety-related ones like the Panic Button or Emergency Contacts, should be easily accessible with minimal effort and attention.
- **Under Stress:** During emergency situations or high-stress moments, users may not be in the best state of mind to navigate a complex application. Therefore, the app should be designed with clear, intuitive, and straightforward interfaces that allow users to access safety features rapidly. The Panic Button and Emergency Contacts, in particular, should be easily accessible in such situations to ensure swift response and support.

In summary, MyCampus is designed to improve the campus experience by providing safety and convenience features for the broader university community (Big-C) and personal, day-to-day use (small-C). It will be primarily used on the university campus, but constraints such as connectivity, location, and memory may impact its performance. The app's design should consider users' distracted and stressed states of mind, ensuring ease of access, particularly during emergencies. This approach aims to create a more secure and user-friendly campus environment for all.

4.6 Application Functionalities

Table 4.1: Functionalities of MyCampus Mobile Application

Functionality	**Use-case**
Emergency Contacts	Displays the list of emergency contacts with a call option.
Panic (SOS) Button	Allows users to make an emergency call directly by pressing the SOS Button.
Incident Reporting	Allows users to submit a report about any campus incident.
Accessibility Request	Allows users to submit a request for any university-based resources.
History	Displays the history of past user requests.
Campus Map	Uses Google Maps API for effective campus navigation, including location pinpoints for popular campus amenities.
Transportation Information	Displays real-time transport accessibility information, such as bus timings.
Notification System	Displays all recent notifications and allows the user to view them in detail.
Resources	Permits users to access a list of essential campus web-based resources.
Home Screen	Presents the user with a visually appealing home screen with a real-time weather widget and recent campus news.
User Account	Presents basic user information and allows users to update their personal information at any time.

4.7 Used Tools

While developing the application we intend to use the tools below:

- **Git:** As there are multiple people working on different features there should be a centralized repository which holds the stable and most updated code, we have decided to use Git which is a version control system that allows multiple developers to collaborate on a project [8]. It helps in tracking changes in the source code, enables branching for parallel development, and facilitates code merging [8].
- **Android Studio:** Android Studio is the official integrated development environment (IDE) for Android app development. Within the Android studio, it provides a robust set of tools for designing, dependencies, coding, testing, and deploying Android applications [10].
- **JUnit:** JUnit is a widely used testing framework for Kotlin/Java. It provides annotations and assertions to simplify unit testing, making it easier to write and execute test cases. This ensures that individual units of code (e.g., functions, methods) function as expected, helping to catch and address bugs early in the development process.
- **Microsoft Teams:** For all communications online, we choose to use Microsoft Teams.

4.8 What did you create?

We created a minimalistic Android application called MyCampus, designed to enhance the overall campus experience for university students and faculty. The application amalgamates various essential features, including emergency contacts, incident reporting, accessibility request submission, location sharing, campus maps, notification system, and a Panic (SOS) Button. MyCampus aims to provide a comprehensive solution for safety and convenience, streamlining daily campus life.

4.9 How much minimum, desired, and bonus functionality did you implement?

- **Minimum Functionality:** All minimum functionalities have been successfully implemented. These include emergency contacts, incident reporting, accessibility request submission, campus map with real-time transportation information, notification system, and the Panic (SOS) Button.
- **Desired Functionality:** One out of two desired functionalities has been implemented. The implemented functionality is related to simplifying and streamlining various aspects of daily campus life, including locating vending machines, identifying safe walking routes, exploring detailed campus maps, and receiving important notifications for class updates and weather alerts.
- **Bonus Functionality:** One out of two bonus functionalities has been implemented. The implemented bonus functionality is related to fostering a sense of community by enabling students to connect with peers, faculty, and the university administration, even during stressful times or emergencies.

4.10 What does your app actually do?

The MyCampus app serves as a comprehensive solution for enhancing safety and convenience on university campuses. It provides features like emergency contacts, incident reporting, accessibility request submission, real-time campus navigation, transportation information, notification system, and a Panic (SOS) Button for emergency situations. The app also promotes a sense of community by facilitating communication and support among students, faculty, and the university administration.

4.11 How does it differ from what you planned?

Design Principles:

- **Simplicity Over Complexity:** During the development process, there was a conscious decision to prioritize simplicity in design. This was based on the design principle that a clean and straightforward user interface enhances user experience. As a result, certain features that might have added complexity were either streamlined or omitted, aligning with the minimalist design approach.

- **User-Centric Focus:** The emphasis on user-centric enhancements led to a slight deviation from the original plan. The features that directly contribute to user safety and community building took precedence over additional functionalities. This aligns with the user-centered design principle of addressing primary user needs first, ensuring a more impactful and meaningful user experience.

Practicality:

- **Resource Constraints:** Practical considerations, such as resource limitations in terms of development time and available technology, influenced the final product. Certain features that required extensive development time or advanced technologies were either simplified or postponed. This pragmatic approach ensured the timely delivery of a functional and reliable application.
- **Feedback Integration:** Continuous feedback from potential users during the development process prompted adjustments to the original plan. Some features were refined or reimagined based on early user feedback, ensuring that the final product better met the practical needs and expectations of the target audience.

User Sensitivity:

- **Emergency Focus:** The decision to prioritize safety features, such as the Panic (SOS) Button and Emergency Contacts, was driven by a heightened sensitivity to the critical nature of these scenarios. In emergency situations, user interfaces need to be intuitive, and functionalities must be easily accessible. This focus on user sensitivity influenced the design choices to ensure a swift and effective response during urgent situations.
- **Inclusive Design:** Sensitivity towards diverse user needs and abilities influenced the design choices related to accessibility features. The application incorporates an Accessibility Request functionality, aligning with the principles of inclusive design to cater to users with varying needs and capabilities.

In summary, the differences between the planned and implemented functionalities can be attributed to a conscious adherence to design principles, practical considerations related to available resources, and a heightened sensitivity to user needs, particularly in emergency situations. The final product reflects a strategic and pragmatic approach that prioritizes essential features while ensuring a positive and user-friendly experience.

4.12 Planning complexity

While the development plan exhibited a comprehensive vision for MyCampus, certain aspects proved to be more ambitious than initially anticipated. Several factors contributed to the challenges faced during implementation:

Feature Complexity:

The plan initially aimed to include a wide array of features to enhance both safety and daily convenience. However, as development progressed, it became evident that the complexity of some features, especially those related to detailed campus mapping and vending machine location services, surpassed the initially estimated development effort. This complexity posed challenges in meeting the desired timelines and resource allocations.

Integration Challenges:

The integration of real-time transportation information using the Google Maps API presented unforeseen challenges. The initial plan did not fully account for the intricacies of seamlessly integrating this dynamic data source into the application. As a result, additional time and effort were required to ensure a smooth and accurate transportation information system.

User Experience Iterations:

The plan did not adequately foresee the iterative nature of user experience (UX) design. As user testing and feedback revealed, certain design choices needed refinement to align more closely with user preferences and expectations. These iterations, while crucial for delivering an optimal user experience, extended the development timeline beyond the initially estimated duration.

5 Issues

5.1 Challenges Encountered

SOS Feature and CALL_PHONE Permission:

- **Challenge:** Initially, we faced challenges implementing the SOS feature as our goal was to enable a direct phone call. However, we encountered permission issues, specifically with the CALL_PHONE permission.
- **Learning:** Through debugging, we discovered the necessity of the CALL_PHONE permission for initiating phone calls. This realization prompted us to incorporate the required permission, overcoming the hindrance and ensuring the proper functionality of the SOS feature.
- **Resolution:** After debugging, it was identified that the CALL_PHONE permission was required. Adding this permission in the **AndroidManifest.xml** file resolved the issue. A proper understanding of Android permissions and thorough testing helped in overcoming this challenge.

RoomDB Data Integrity Issues:

- **Challenge:** Maintaining data integrity became a concern, particularly when there were schema changes in the **RoomDB**. This issue manifested as disruptions in app persistence.
- **Resolution:** Uninstalling the app in the **Android Virtual Device (AVD)** or incrementing the version number of the database resolved the problem. This solution ensured that the database was recreated with the updated schema, maintaining data integrity.

Map Feature Integration:

- **Challenge:** During the development of the map feature, one of the significant challenges we encountered was ensuring the proper rendering of the map within a fragment without affecting the positioning of other elements. Achieving a visually seamless integration of the map while avoiding any offsetting or overlapping with other UI components required careful attention to layout and styling.
- **Resolution:** Careful attention to layout and styling was crucial. Balancing visual appeal and functionality of UI buttons over the map required a delicate process. Regular testing and adjustments in layout design resolved these challenges, ensuring a seamless integration of the map.

Live Location Sharing Feature:

- **Challenge:** In location-sharing implementation, we faced a challenge involved managing share permissions. Verifying whether the application had the necessary permissions for location sharing and handling the permission request process presented a challenge.
- **Resolution:** This required implementing logic to check and request permissions, as well as dealing with scenarios where users might deny access, ensuring a smooth and user-friendly experience while respecting privacy concerns. As one of the team member is faced this issue prior has helped us in tracking down the issue and resolve it quickly.

Transit Data:

- **Challenge:** Fetching schedules of transit busses of routes 1, 7, 10, 4 data based on the current time is challenging.
- **Resolution:** We built a dataset using transit schedules from the official website for 1, 7, 4 10 bus routes and store the dataset inside a function which is then deployed in AWS lambda. When the user clicks on transit tab the app sends the current time and the api sends the bus timings of the 4 routes which handled by the application by storing it inside the Room DB and display the elements from the DB records.

Emergency Contacts Feature:

- **Challenge:** Making the intent - when the user clicked the call button - it should redirect to the dialler with that emergency contact number. Also, making the emergency contacts appear in a visually appealing way - layout issues.
- **Resolution:** We have regrouped and discussed the implementation of intents and addressing layout issues in the XML file resolved the challenges. Using the visual editor for designing and testing contributed to a visually appealing and functional emergency contacts feature.

Incident Report and Time/Date Picker Issues:

- **Challenge:** Faced issues in Time Picker and Date Picker - storing the correct type data in the database and the fitting of time picker and date picker within the parent layout in Incident Report Activity.
- **Resolution:** Using the visual editor for proper design, addressing layout issues, and ensuring correct data storage in the database resolved these challenges. Thorough testing was crucial to avoid issues related to user interaction with the time and date pickers.

Accessibility Feature:

- **Challenge:** Faced issue while developing the accessibility form feature, where it stores the accessibility request data in the database and retrieving them.
- **Resolution:** Referring to the documentation and implementing features based on guidelines resolved the challenge. Understanding the specific requirements for accessibility features and ensuring proper database interactions contributed to the success of this feature.

Dependency Issues:

- **Challenge:** Dependency issues in the configuration file while testing the test case. Even thought we all have decided basic template and used the same, group members adding dependencies related to their features caused this issue.
- **Resolution:** Correcting versions in the build.gradle file resolved the dependency issues. Ensuring that dependencies across the project were compatible and had consistent versions contributed to a stable testing environment by ensuring consistent versions and configurations. Regularly updating dependencies and checking for compatibility helped prevent runtime errors.

Loading Resources in RecyclerView with Glide:

- **Challenge:** Loading images in a RecyclerView based on their types.
- **Resolution:** Implementing logic to differentiate between types and using the Glide library to load images based on the type. Proper handling of image loading in a RecyclerView ensured a smooth user experience.

Handling WebView Navigation:

- **Challenge:** Handling WebView navigation and managing visibility between WebView and RecyclerView. This issue is caused because of incorrect handling of WebView navigation or visibility toggling
- **Resolution:** Ensuring correct handling of WebView navigation and toggling visibility appropriately. Addressing issues related to incorrect handling of WebView interactions prevented unexpected behaviour.

Merge Conflicts:

- **Challenge:** Conflicts while integrating code of different users and different features. While most of these issues are related to dependency issues, due to different implementation of same code like DB connection initiation caused the merge conflicts.
- **Solution:** Regular communication among team members, utilizing version control systems effectively, and resolving conflicts promptly. Planning regular code integration sessions and establishing coding standards helped prevent major merge conflicts. Although we faced merge conflicts these doesn't take too much time to solve and didn't impact the project timeline hugely.

6 Satisfaction

6.1 What are you happy with?

We are delighted with the smooth teamwork we experienced. The allocation of features among us happened seamlessly, and each team member implemented their respective features without conflicts, meeting the project deadline successfully. It's worth noting that each of us took ownership of our tasks, conducted individual research, delved into our topics, and studied external components required for feature implementation. This collaborative effort ensured the successful functioning of the app, showcasing our team's commitment and effective coordination.

6.2 Areas of Unhappiness

While reflecting on the MyCampus project, there were instances that brought about a sense of dissatisfaction, and two notable aspects stand out. Firstly, the limited user base posed challenges in conducting thorough beta testing. Due to constraints in reaching a diverse user pool, the beta testing phase was not as comprehensive as desired.

Another source of unhappiness was the necessity to prioritize essential functionalities over the implementation of certain interesting bonus features. As much as we had envisioned enriching the user experience with these bonus features, resource constraints, and the need to focus on must-have functionalities took precedence. While this prioritization was essential for delivering a functional product within the given timeframe, it did lead to a sense of missed opportunities.

6.3 Did you learn a lot

Yes, indeed. Many of us were new to Android development and Kotlin. Through a combination of coursework and individual research tailored to our app, we gained substantial knowledge and experience. The learning curve was steep, but it allowed us to acquire valuable insights into Android development practices, Kotlin programming, and the intricacies of building a functional and cohesive mobile application. Our collective learning journey significantly contributed to the successful development of the project.

6.4 Did you have insight into development?

Yes, from a team perspective in an Android project, we collectively had insight into the development process. Each team member contributed their expertise, insights, and experiences, creating a collaborative environment. We engaged in regular discussions, shared knowledge, and collaborated on decision-making, ensuring that everyone was well-informed about the project's development aspects. This collective insight played a crucial role in problem-solving, optimizing workflows, and achieving a cohesive development approach.

6.5 Did you make friends?

Throughout the course of the project, we not only worked together but also developed meaningful friendships. Prior to the project, we were not familiar with each other, but as we collaborated on tasks, engaged in discussions, and faced challenges together, a sense of camaraderie emerged. We now communicate through chats and have enjoyable interactions during in-person meetings. These connections have grown beyond the confines of the project, and we have developed friendships that extend into our personal lives. This collaborative experience has not only enriched our professional skills but has also created lasting bonds among us.

6.6 QA

In terms of quality assurance, we implemented a robust testing strategy for our Android project. This included:

- **Unit Testing:** We ensured the reliability of our application components by conducting thorough unit testing. This involved writing tests to assess the functionality of individual units of code.
- **Functionality Testing:** Our quality assurance process involved extensive functionality testing. We systematically experimented with the application workflow to ensure that all features worked as intended and met the specified requirements.
- **Test Coverage:** Achieving a comprehensive test coverage of approximately 80%, we diligently tested various aspects of the application to identify and address potential issues. This helped enhance the overall reliability and stability of the app.
- **Libraries & Frameworks:** Leveraging industry-standard tools, we utilized JUnit and Mockito for effective testing. These libraries and frameworks played a key role in streamlining our testing processes and ensuring the accuracy of our results.

This testing approach was integral to our commitment to delivering a high-quality and reliable Android application.

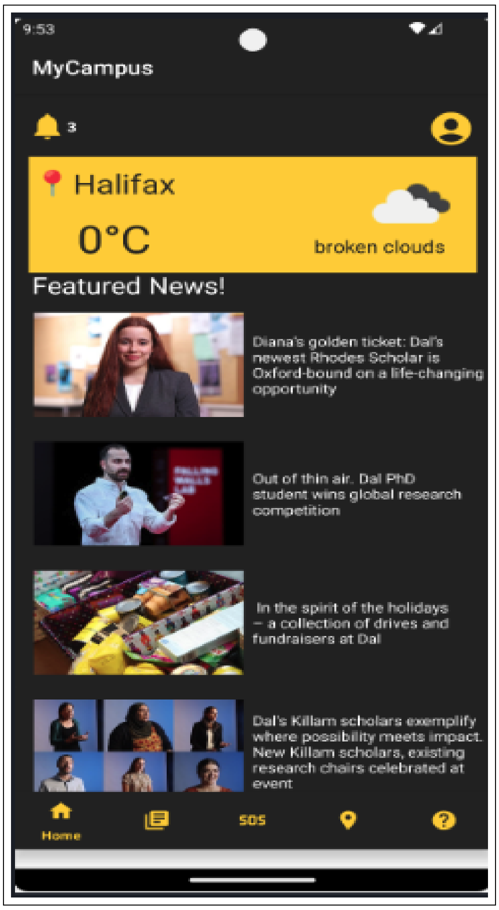
6.7 Task Distribution

We have divided the tasks equally. We regularly scheduled meetings to check on blockheads, issues, and errors. The permission issue for SOS phone calls was notified to all teammates in one of these meetings which gave a heads-up for others. Everyone contributed equally and fairly. Almost all tasks were shared among us and everyone helped and chimed their inputs and elaborated on their work, tasks, and progress in the deliverables.

Name	**Feature Developed**
Pratik Mukund Parmar	Account Info, Notifications
Badhri Arja	Campus Map, Location Sharing
Naveen Kunapaneni	Transit Data, Map Markers
Lokeshwar Kumar Tabjula	SOS, History, Main Application Layout
Kovarthanan Murugan	Resources, Updates
Kishoreganesh Sundararajan	Weather API, Featured News
Emayan Vadivel	Emergency Contacts, Accessibility & Incident Reports, Unit Testing

7 Demonstration

7.1 Screenshots



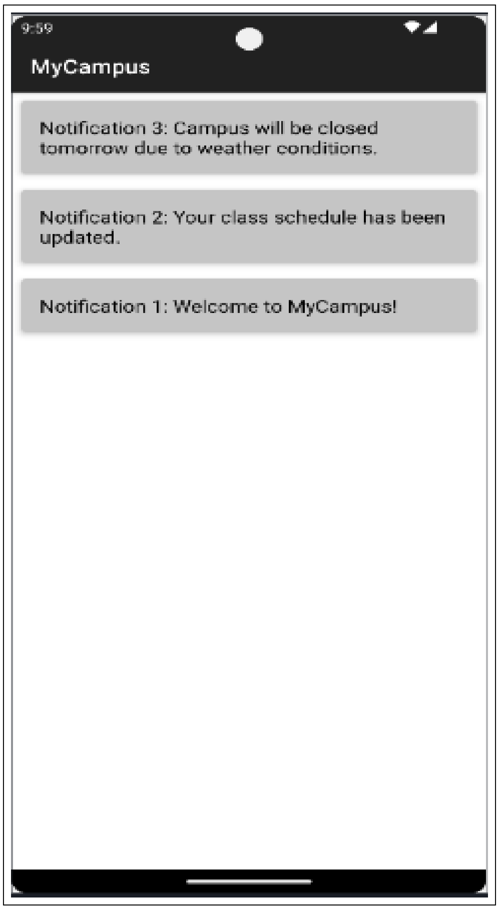
Diana's golden ticket: Dal's newest Rhodes Scholar is Oxford-bound on a life-changing opportunity

Out of thin air. Dal PhD student wins global research competition

In the spirit of the holidays – a collection of drives and fundraisers at Dal

Dal's Killam scholars exemplify where possibility meets impact. New Killam scholars, existing research chairs celebrated at event

Figure 7.1: **MyCampus Home Screen with Weather widget and News Feed**



9:59

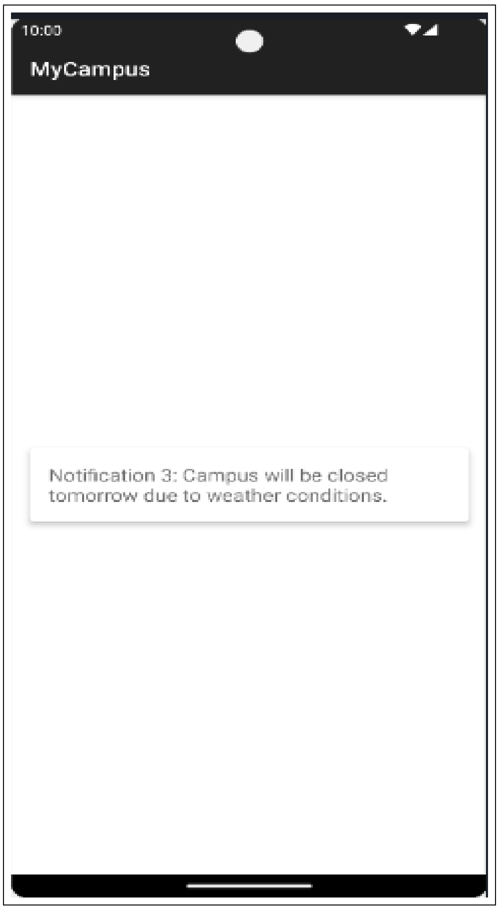
MyCampus

Notification 3: Campus will be closed tomorrow due to weather conditions.

Notification 2: Your class schedule has been updated.

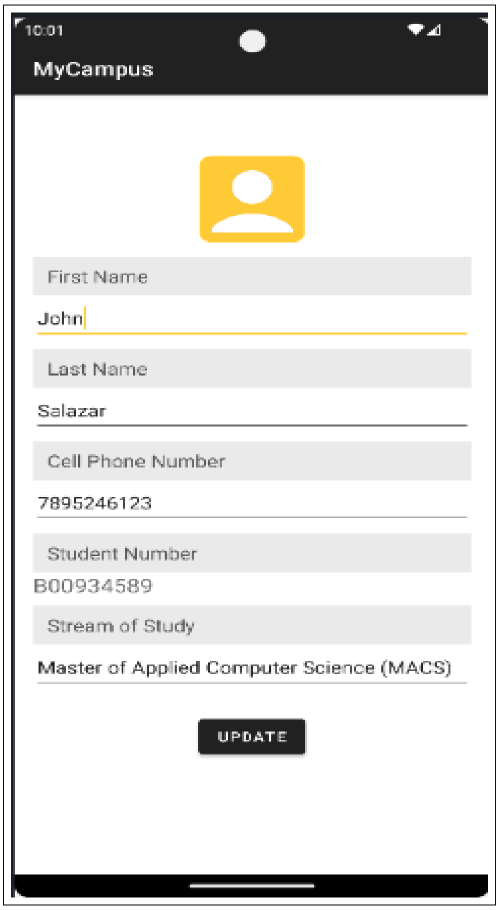
Notification 1: Welcome to MyCampus!

Figure 7.2: **Notification Screen**



Notification 3: Campus will be closed tomorrow due to weather conditions.

Figure 7.3: **Viewing notification in detail**



First Name

John

Last Name

Salazar

Cell Phone Number

7895246123

Student Number

B00934589

Stream of Study

Master of Applied Computer Science (MACS)

Figure 7.4: **User Account Screen**

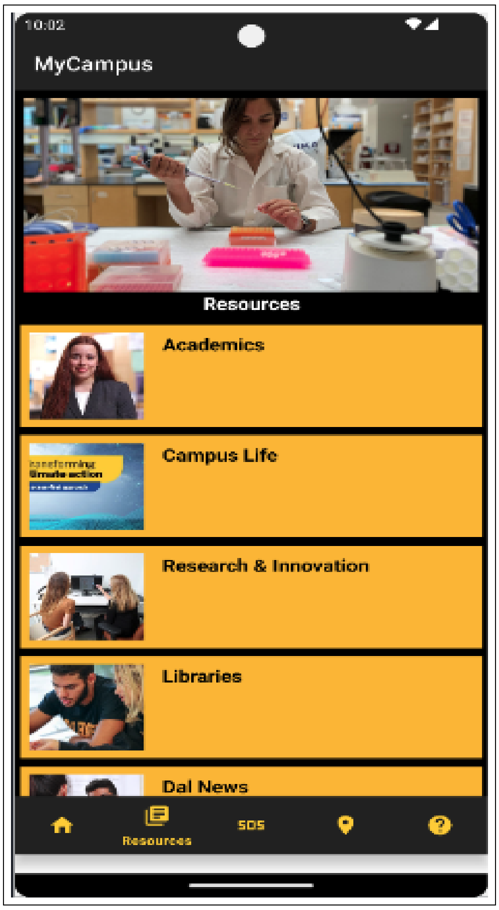
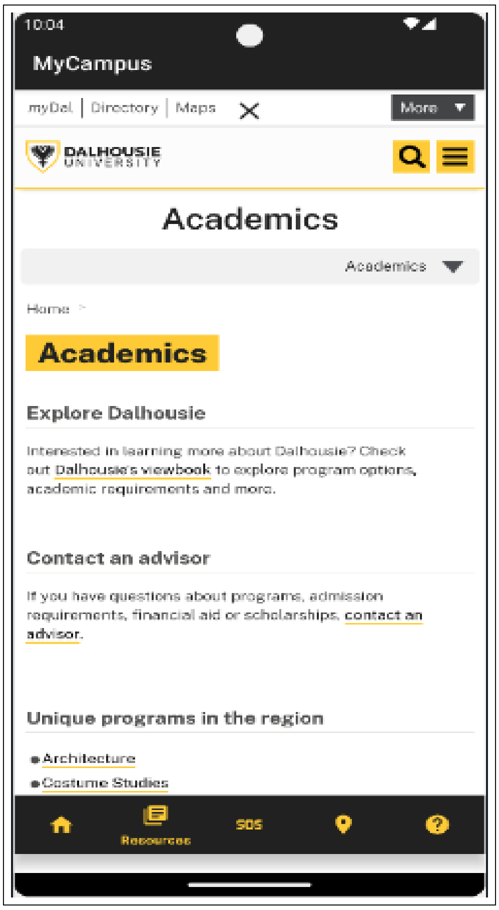


Figure 7.5: **Resources Screen with the list of campus web-based resources**



Academics

Explore Dalhousie

Interested in learning more about Dalhousie? Check out Dalhousie's viewbook to explore program options, academic requirements and more.

Contact an advisor

If you have questions about programs, admission requirements, financial aid or scholarships, contact an advisor.

Unique programs in the region

- Architecture
- Costume Studies

Figure 7.6: **Viewing a specific resource - Academics**

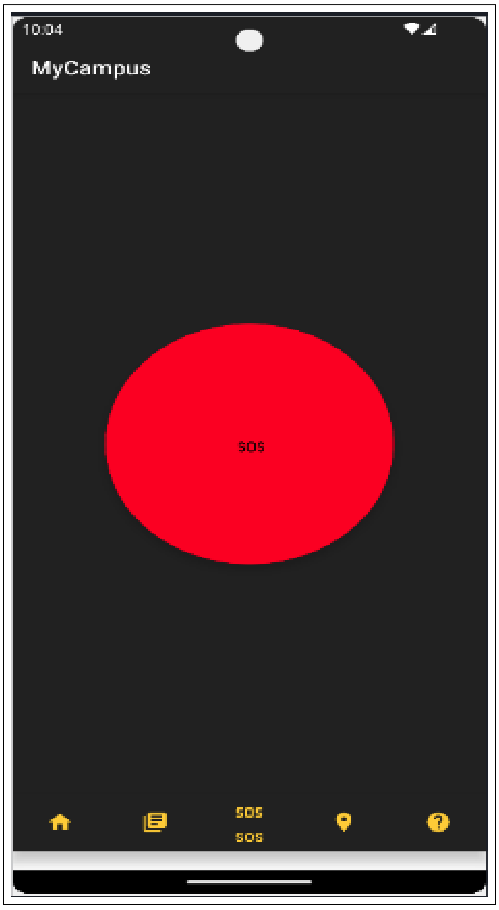
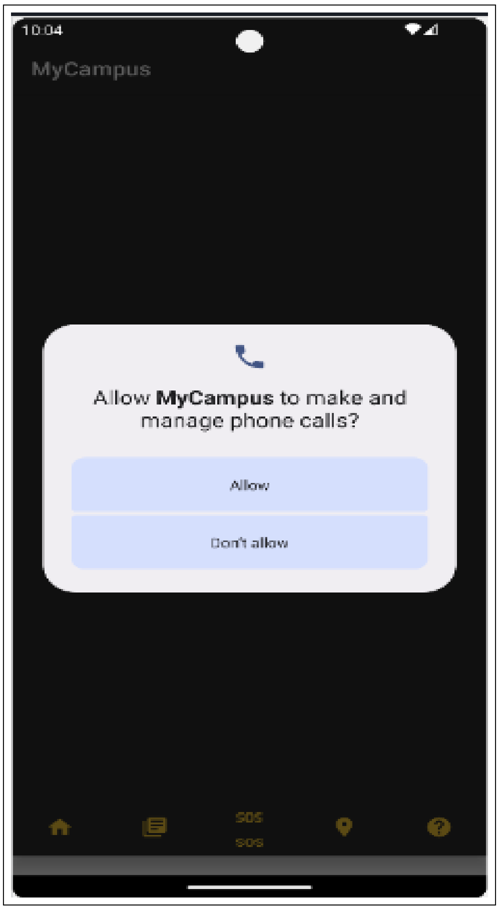


Figure 7.7: **SOS - Emergency call button**

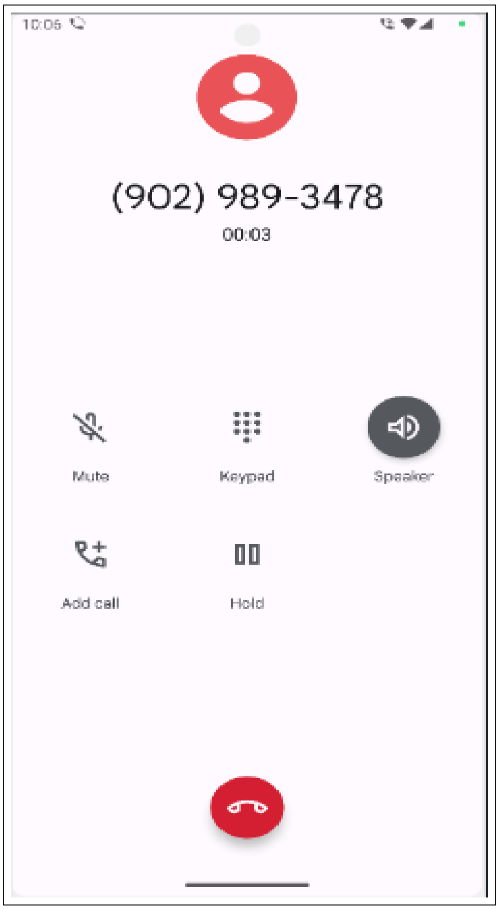


10:04

MyCampus

Allow **MyCampus** to make and manage phone calls?

Figure 7.8: **Asking permission to make calls**



(902) 989-3478

00:03

Figure 7.9: **Calling SOS emergency contact**

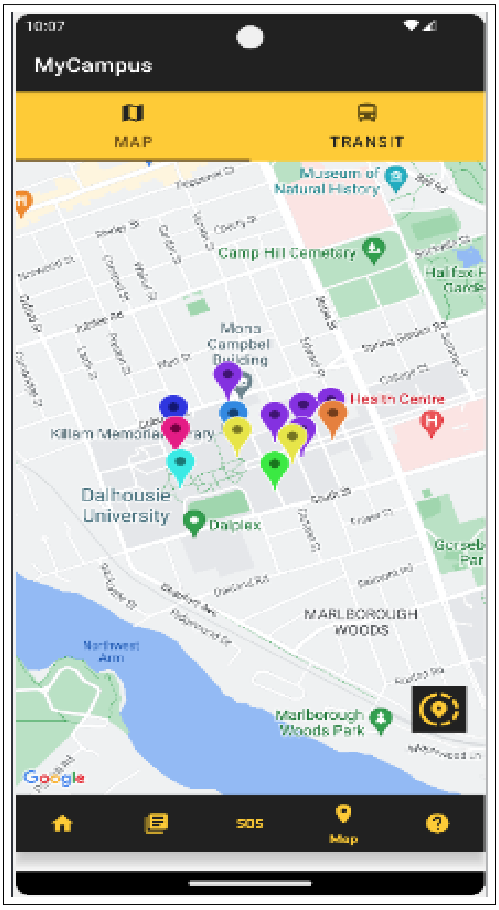


Figure 7.10: **Location Screen - Campus Map with pinned locations**

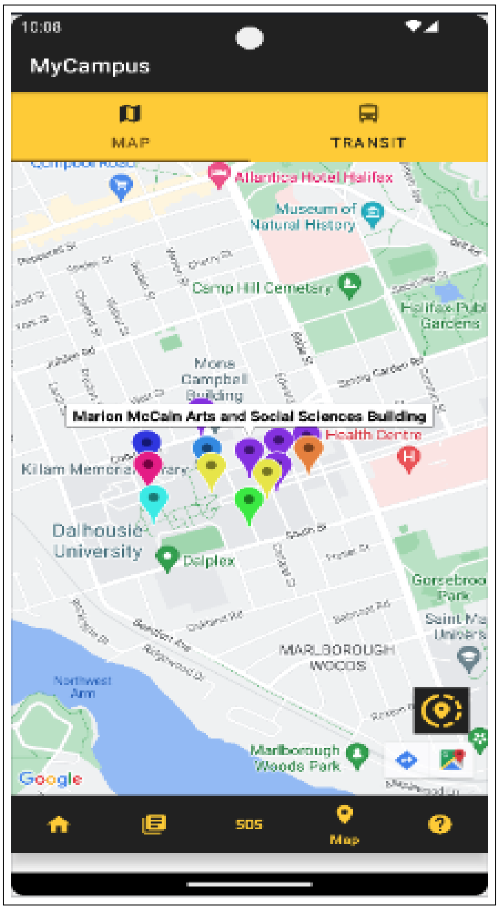
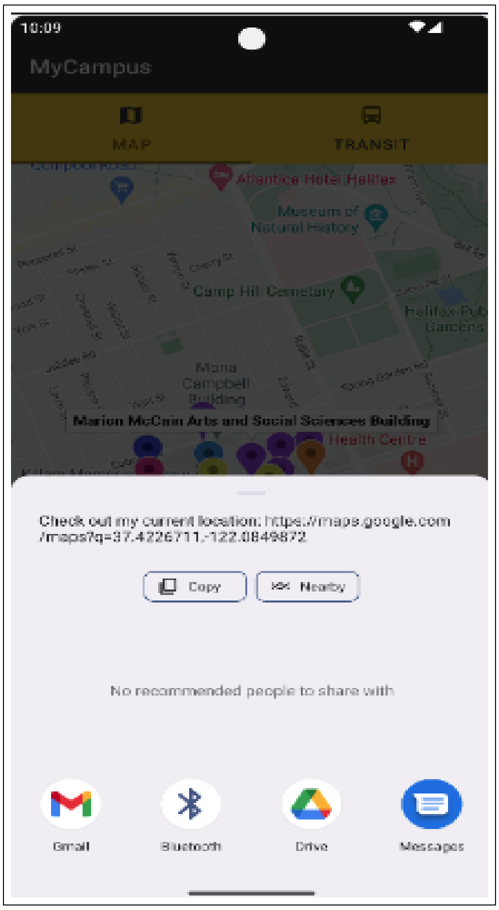


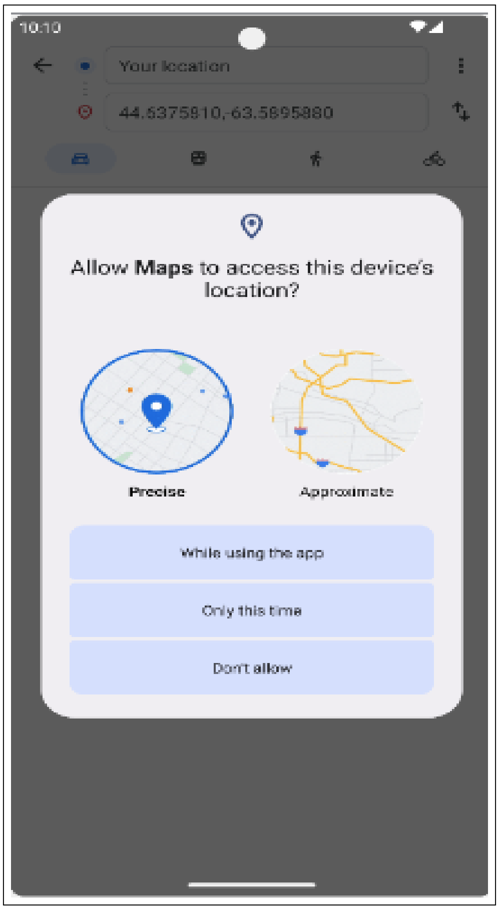
Figure 7.11: **Displaying the name of the location**



Check out my current location: https://maps.google.com/maps?q=37.4226711,-122.0849872

No recommended people to share with

Figure 7.12: **Sharing the location**



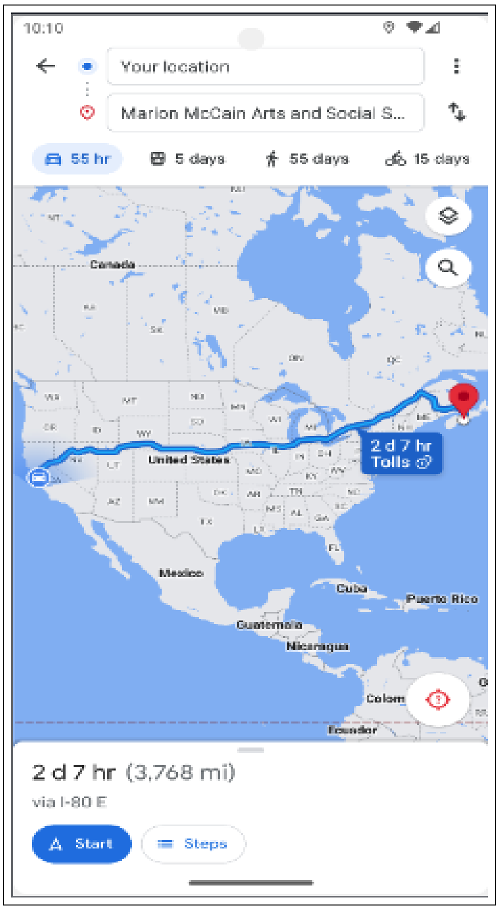
10:10

Your location

44.6375810,-63.5995880

Allow **Maps** to access this device's location?

Figure 7.13: **Asking permission for using Maps**



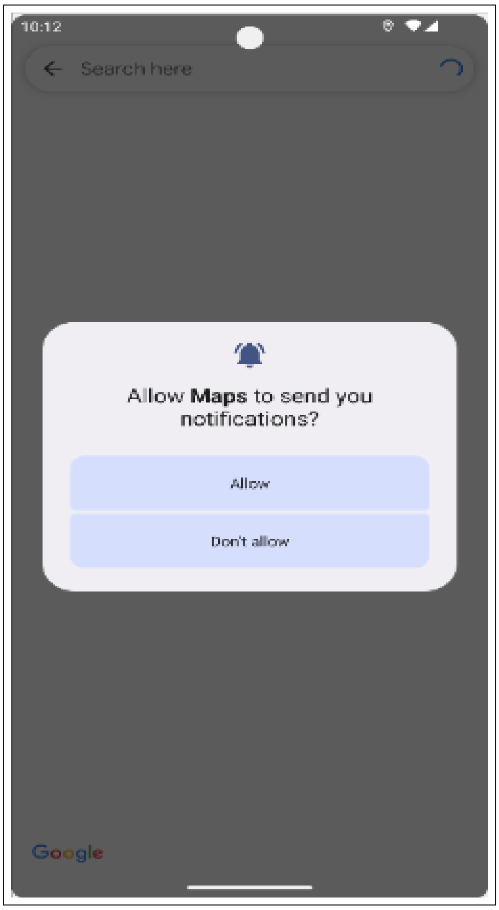
Your location

Marion McCain Arts and Social S...

2 d 7 hr (3,768 mi)

via I-80 E

Figure 7.14: **Displaying the distance between the locations**

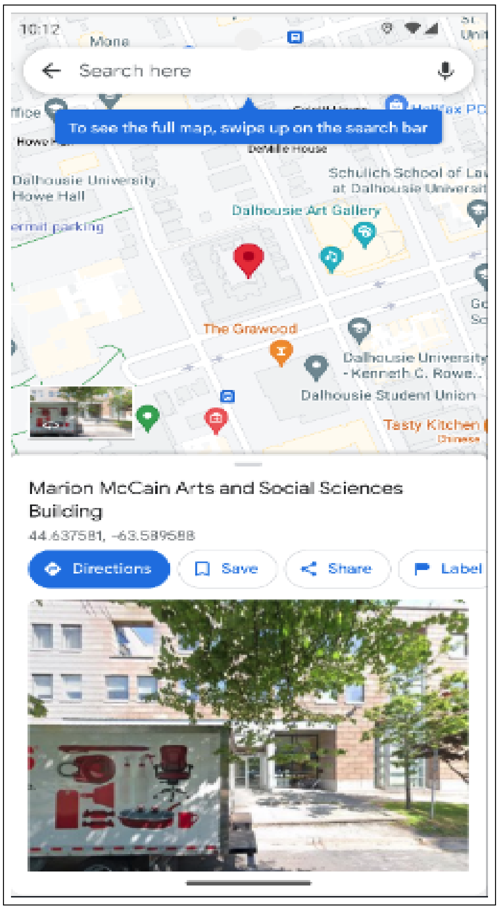


10:12

Search here

Allow **Maps** to send you notifications?

Figure 7.15: **Asking permission to send notifications**

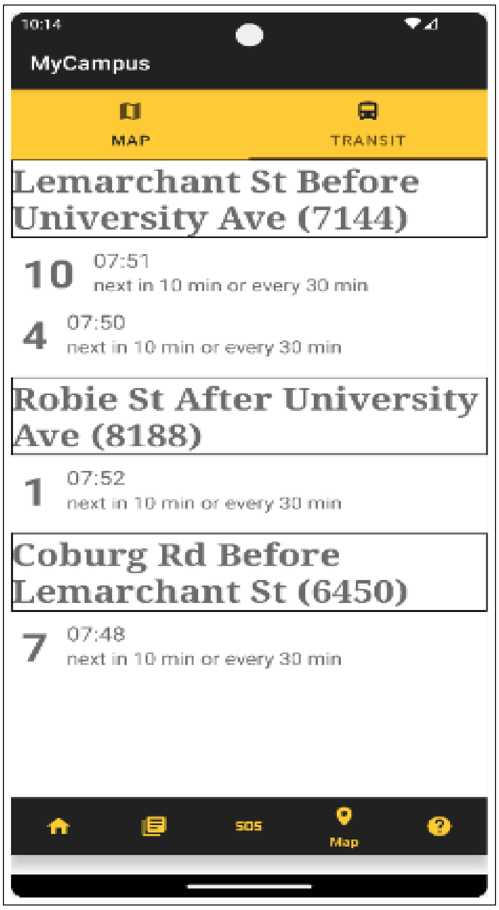


To see the full map, swipe up on the search bar

Marion McCain Arts and Social Sciences Building

44.637581, -63.589588

Figure 7.16: **Location specific details**



Lemarchant St Before University Ave (7144)

10 07:51 next in 10 min or every 30 min

4 07:50 next in 10 min or every 30 min

Robie St After University Ave (8188)

1 07:52 next in 10 min or every 30 min

Coburg Rd Before Lemarchant St (6450)

7 07:48 next in 10 min or every 30 min

Figure 7.17: **Displaying Transit Data - Bus Timing Information**

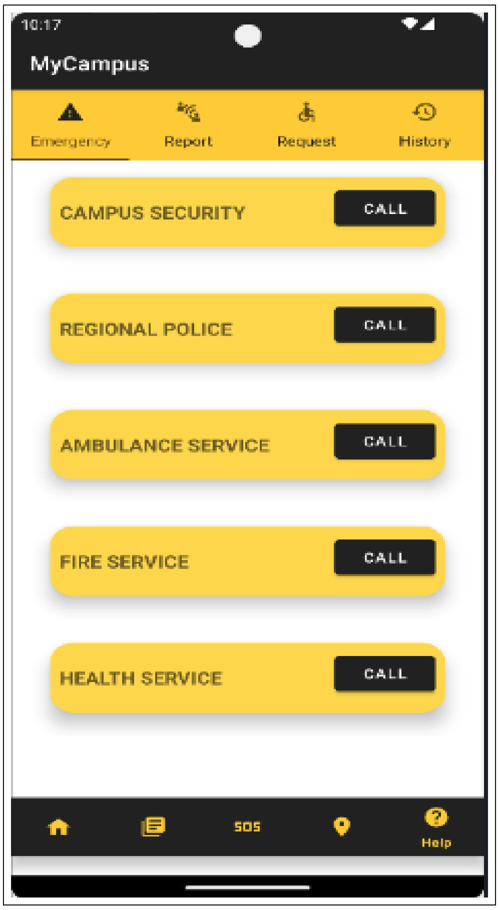


Figure 7.18: **List of Emergency contacts with a call option**

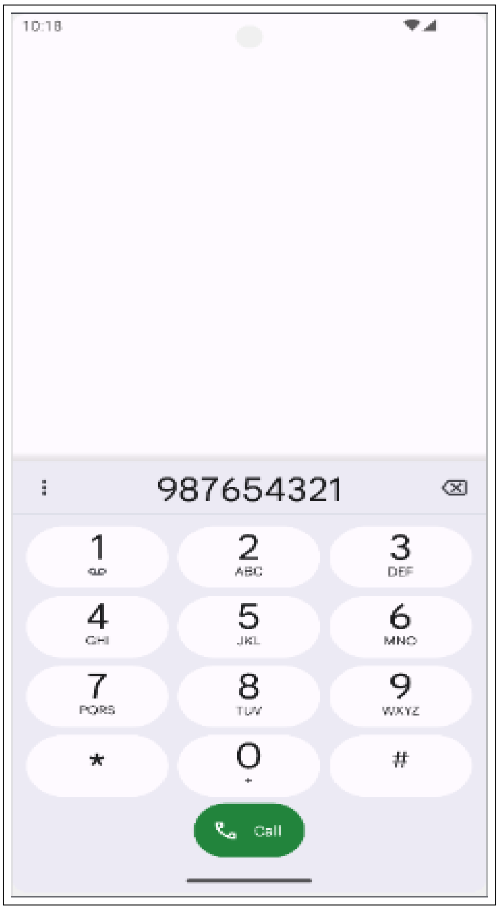


Figure 7.19: **Redirecting to phone dialler with the emergency contact**

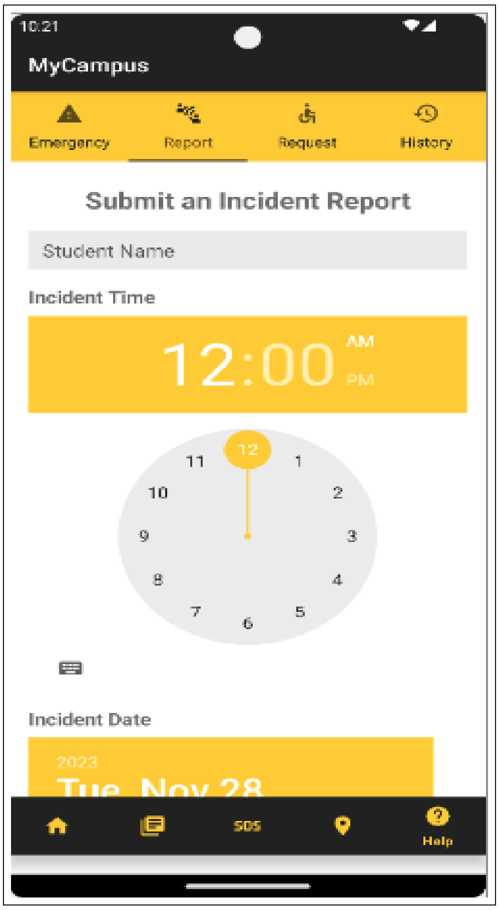
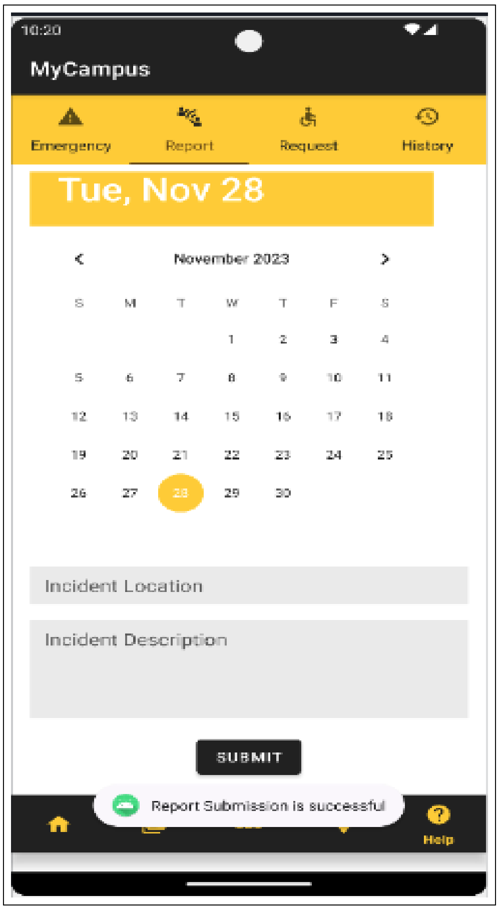
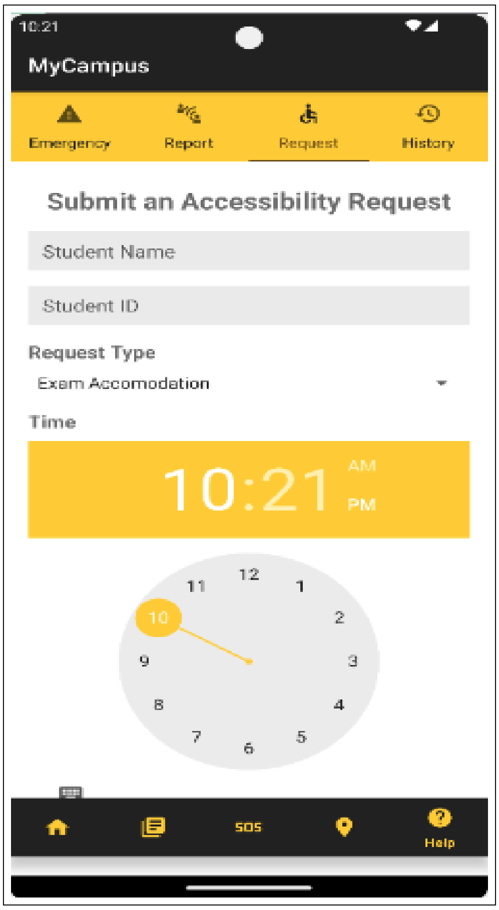


Figure 7.20: **Incident report form**



Report Submission is successful

Figure 7.21: **Incident report form submitted successfully**



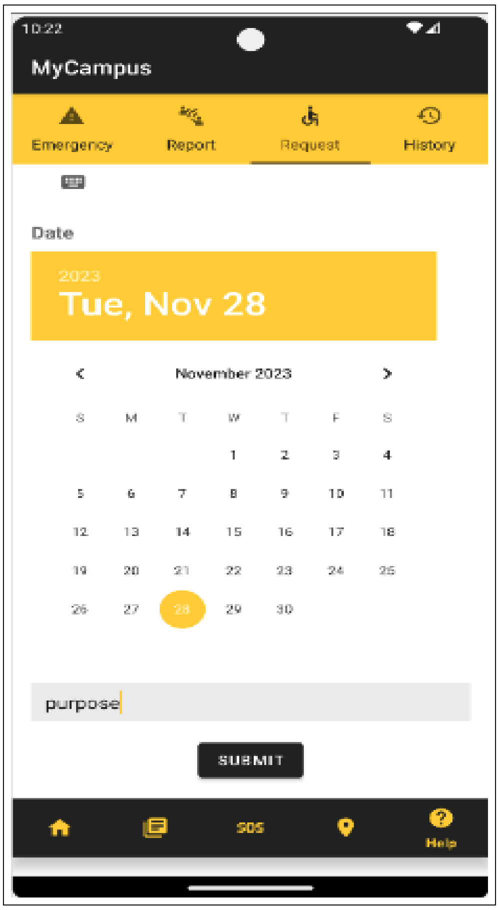
Request Type

Exam Accomodation

Time

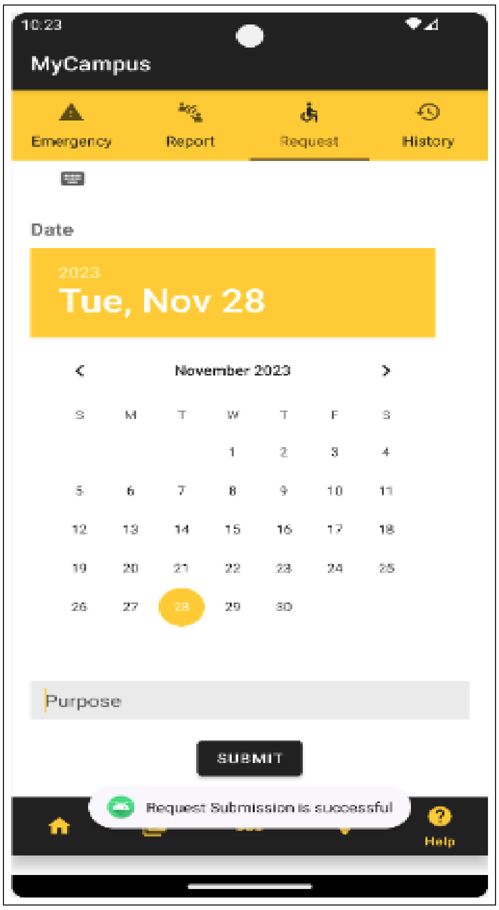
10:21

Figure 7.22: **Accessibility request form**



purpose

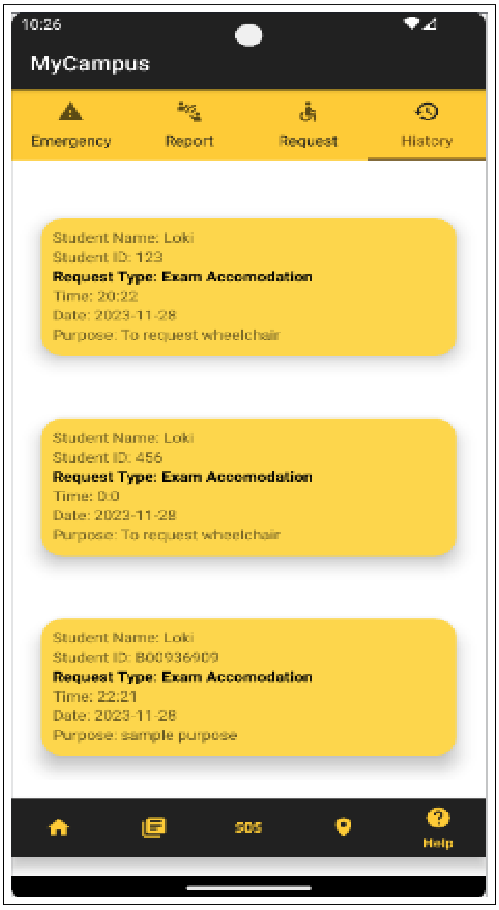
Figure 7.23: **Filling Accessibility request form**



2023
Tue, Nov 28

Request Submission is successful

Figure 7.24: **Submitted Accessibility request form**



Student Name: Loki
Student ID: 123
Request Type: Exam Accomodation
Time: 20:22
Date: 2023-11-28
Purpose: To request wheelchair

Student Name: Loki
Student ID: 456
Request Type: Exam Accomodation
Time: 0:0
Date: 2023-11-28
Purpose: To request wheelchair

Student Name: Loki
Student ID: B00936909
Request Type: Exam Accomodation
Time: 22:21
Date: 2023-11-28
Purpose: sample purpose

Figure 7.25: **Viewing the past history of accessibility requests**

8 Conclusion

Our experience in developing MyCampus was a journey of discovery, marked by the interplay of ambition and practicality. At the outset, our plan reflected a comprehensive vision for a feature-rich application that would redefine the university experience. However, as the development process has been an insightful journey with several key takeaways. The primary accomplishment lies in the successful implementation of various features to enhance user accessibility and streamline processes for students. Throughout this development, I've gained hands-on experience in Android app development, database integration using Room, and effective UI/UX design principles. The process involved creating forms for accessibility requests, displaying historical data, and ensuring a user-friendly interface.

One of the paramount lessons we learned was the dynamic nature of user needs and the importance of adapting to these evolving requirements. Through user feedback and iterative design processes, we discovered the nuanced intricacies of user expectations. This iterative approach not only refined the user experience but also highlighted the need to be responsive and agile in the face of shifting demands.

A pivotal aspect of our technical journey was the understanding and implementing Room Database functionalities for storing and retrieving data related to accessibility requests and historical records. The integration of RecyclerView to display historical accessibility requests provided valuable insights into efficient data presentation. The user interface improvements, including the addition of cards, borders, and proper alignment, contribute to a visually appealing and intuitive design. The inclusion of spacing adjustments between label and value pairs in user profile forms further enhances the application's aesthetics and user experience.

In essence, our experience with MyCampus encapsulates the delicate balance between aiming high and navigating the practical realities of development. We achieved a product that, while not without its challenges, reflects a strategic compromise that prioritizes essential functionalities, ensuring a meaningful and impactful user experience. This journey served as a testament to the resilience of our team, the adaptability required in the face of unforeseen challenges, and the ultimate satisfaction of delivering a product that aligns with the core needs of our users.

References

[1] "What is Design Thinking? — updated 2023," The Interaction Design Foundation, 25-May-2016. [Online]. Available: https://www.interaction-design.org/literature/topics/design-thinking. [Accessed: 22-Oct-2023].

[2] "Flowchart maker & online diagram software," Draw.io. [Online]. Available: http://draw.io. [Accessed: 21-Oct-2023].

[3] B. Fling, Mobile Design and Development: Practical concepts and techniques for creating mobile sites and web apps. O'Reilly Media, 2009.

[4] Atlassian, "The different types of testing in software — Atlassian," Atlassian, 2023. Available: https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing [Accessed Oct. 22, 2023].

[5] "Build local unit tests," Android Developers, 2023. Available: https://developer.android.com/training/testing/local-tests [Accessed: Oct. 22, 2023]

[6] "GitLab," Gitlab.com, 2021. Available: GitLab Documentation [Accessed Oct. 22, 2023].

[7] "Teams — Microsoft Teams," Microsoft.com, 2023. Available: https://www.microsoft.com/en-us/microsoft-teams/log-in [Accessed Oct. 22, 2023].

[8] "Download Android Studio & App Tools - Android Developers," Android Developers, 2022. Available: https://developer.android.com/studio [Accessed Oct. 22, 2023].