

CWU Campus Navigation App

Software Requirement Specification

Group 2:

Austin M Sill, Chakar Baloch, Devlin Hamill,
Eli Holter, Krai Pongrapeeporn, Vandan Amin

Department of Computer Science, Central Washington University
CS481 - Capstone Project

Dr. Donald Davendra

March 19, 2025

Contents

1	Introduction	2
1.1	Purpose	2
1.2	Scope	2
1.3	Glossary	3
2	General Requirements	3
2.1	Functions	3
2.2	External Interfaces	3
2.3	Hardware Interfaces	4
2.4	Software Interfaces	4
3	Use Cases	4
3.1	Use Case 1: Sign Up / Login	5
3.2	Use Case 2: Edit Profile	7
3.3	Use Case 3: Create Event	8
3.4	Use Case 4: Edit Event	9
3.5	Use Case 5: Remove Event	11
3.6	Use Case 6: Change Preferences	12
3.7	Use Case 7: View Directions	13
3.8	Use Case 8: View Map	14
3.9	Use Case 9: View Favorites	15
3.10	Use Case 10: Modify Favorites	16
3.11	Use Case 11: View Location Information	17
3.12	Use Case 12: Search Locations	18
3.13	Use Case 13: Navigate to Locations	19
3.14	Use Case 14: View Calendar	20
3.15	Use Case 15: View Weekly Schedule	20
3.16	Use Case 16: View Daily Schedule	22
3.17	Use Case 17: Modify Class	22
4	Traceability Matrix	24

1 Introduction

1.1 Purpose

The purpose of this project is to create a schedule-based campus navigational aid android app for the Central Washington University (CWU) campus. This app can be used by students to map out where they need to go to attend classes per their schedule and can be used to navigate to other locations like stores and sports courts.

1.2 Scope

The application will include the following functionalities:

- An interactable graphical user interface (GUI).
- A map of the CWU campus.
- User profile functionality.
- Sign-in and sign-up pages.
- The user's schedule.
- A favorite locations page.
- Map scheduling.
- Notifications for upcoming classes.
- App-generated directions/pathing used by the user to find the buildings for the classes in their schedule.
- Modification support such as adding, editing, and deleting specific parts of the schedule.
- A daily schedule that displays the users' classes, buildings, and times.

The following features will not be included in the application:

- A map of the entire world.
- Directions to places outside the CWU campus.
- Address-based navigation.
- Google Street View integration.
- Visualization of buildings outside the CWU campus.
- Support for time zones outside Ellensburg's time zone.

The application will be available only on Android platforms.

1.3 Glossary

- Event: A singular period of time that represents scheduled recurring classes or singular events.
- Application Programming Interface (API): A set of functions and commands provided by a software provider that allows connection and interaction with an external software component.
- Database: This will serve as shorthand for Firebase Cloud Firestore database.
- Inactivity: When something is not being interacted with by the user.
- Class: Internally, classes will be represented as recurring scheduled events.

2 General Requirements

2.1 Functions

The app will be written for and runnable on Android devices, supporting Android 5 (API version 21) or later.

2.2 External Interfaces

- **Google Maps API:** The app will use the Google Maps API SDK for Android to show the campus map.
 - API cost: \$7 per 1000 map loads, \$200 free credits per month.
 - All collective installs of the app must be limited to no more than 28,571 map loads per month to avoid going over the free allowance.
- **Google Routes API:** The app will use the Google Routes API SDK for Android to show the path to a destination.
 - API cost: Free usage caps at 10,000 call a month\$10 per 1000 map loads, \$200 free credits per month.
 - All collective installs of the app must be limited to no more than 20,000 route calls per month to avoid going over the free allowance.
- **Firebase Database:** The app will use the free spark plan from Firebase for to store user information and handle sessions.
 - 1 GB of storage.
 - 10 GB/month of data transfer.
 - 100 simultaneous connections.
 - Limited to a single database instance.
- **Global Positioning System (GPS):** The app will utilize the device's GPS functionality to find the user's position. This will be used to:
 - Locate the user on the map and represent it with an icon.
 - Calculate routes from the user to their destination.

2.3 Hardware Interfaces

- **Standard Device Components:**

- Touch screen.
- GPS antenna: The app will use the GPS components of the device to locate the user.
- Internet Connection must be available to retrieve schedule data.

2.4 Software Interfaces

- **GPS API:** This API will allow the app to interface with the GPS system, and will be used to acquire the location of the user.

3 Use Cases

Overview: The student interacts with the Class Scheduling System to manage their class schedules, edit their profile, and navigate to classes. Below are detailed use case specifications for each scenario.

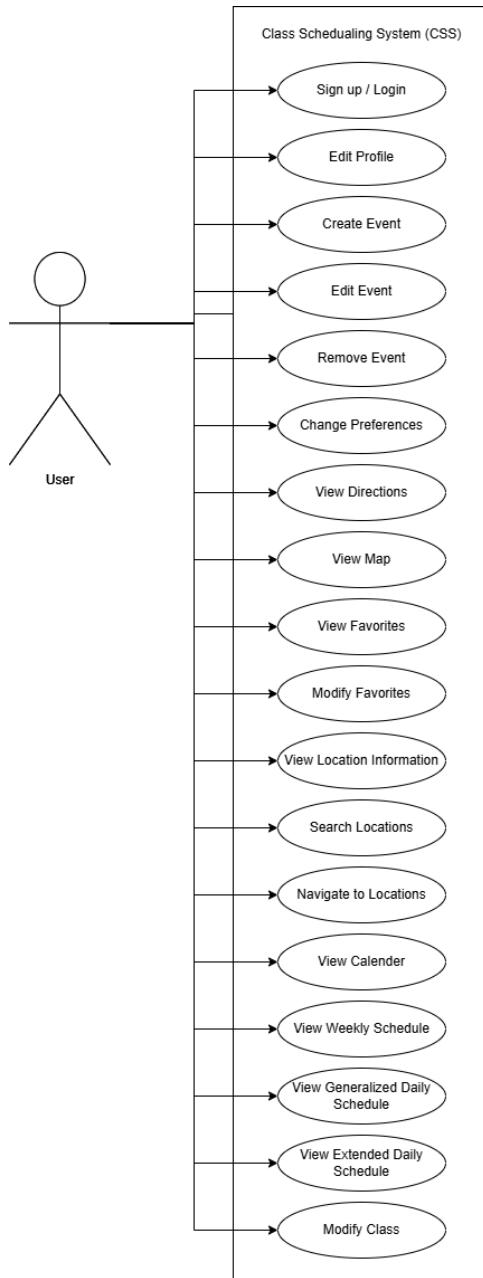


Figure 1: Use Case Diagram

3.1 Use Case 1: Sign Up / Login

Description: The user creates an account (sign up) or logs into an existing account (login).

- **Functional Requirements:**

- 3.1.1 Users can create an account within the app.
- 3.1.2 Users can log in if they have an account.
- 3.1.3 Supports a “log in with Google” feature.
- 3.1.4 Enables saving and retrieving schedules, favorite locations, and settings via Firebase Cloud Firestore.

3.1.5 User data can be retrieved on any device running the app that the user logs into.

- **User Interface:**

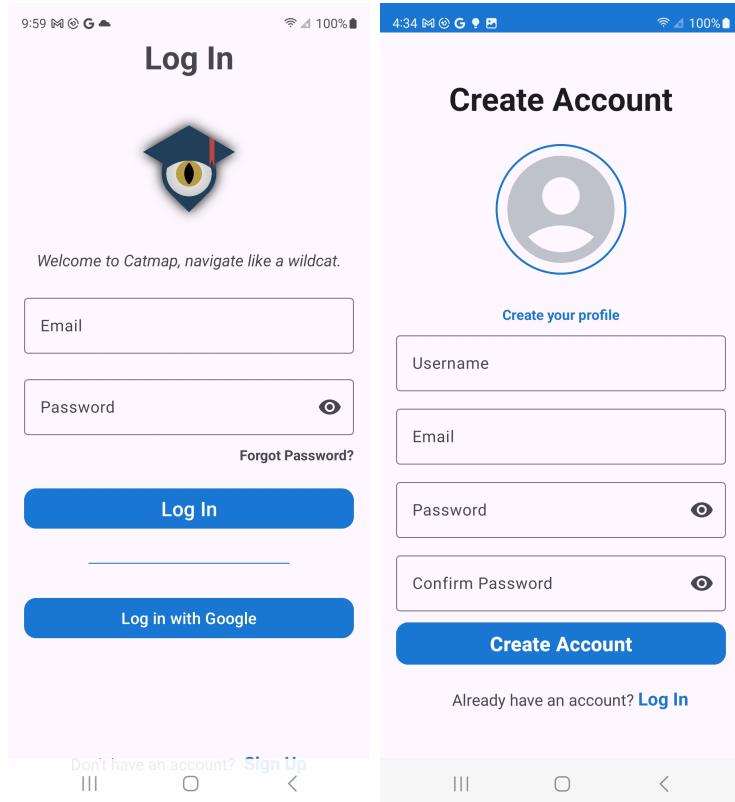


Figure 2: User Interface Mockup: Sign Up / Login

- **Software Interfaces:** Refer to the Appendix, Figure 19.

3.2 Use Case 2: Edit Profile

Description: The user updates personal information like profile picture, name, email address, and password, and their changes are reflected wherever they are logged in.

- **Functional Requirements:**

- 3.2.1 Users can change their profile picture.
- 3.2.2 Users can change their name.
- 3.2.3 Users can change their password.
- 3.2.4 Profile changes will be stored in and retrieved from a Firebase Cloud Firestore database.

- **User Interface:**

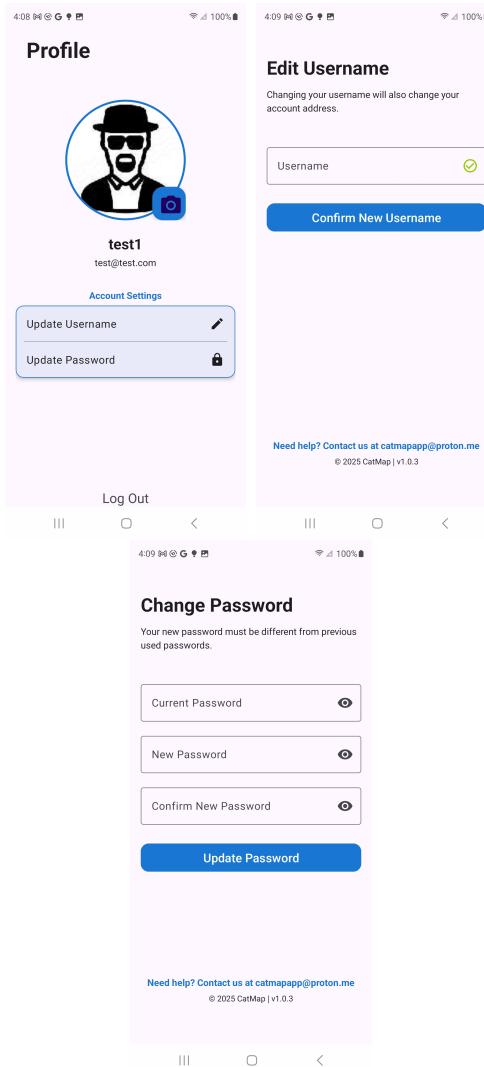


Figure 3: User Interface Mockup: Edit Profile

- **Software Interfaces:** Refer to the Appendix, Figure 20.

3.3 Use Case 3: Create Event

Description: The user adds individual classes (courses) to create an event.

- **Functional Requirements:**

- 3.3.1 A valid date on the calendar needs to be selected prior to adding the event.
- 3.3.2 A valid building on CWU campus needs to be selected prior to adding the event.
- 3.3.3 The event title, such as a subject name or store name, needs to be typed out prior to adding the event.
- 3.3.4 After all info is validated, the event should be added to the user's schedule.
- 3.3.5 An error message should display if any info selected or typed in is invalid.

- **User Interface:**

- **Software Interfaces:** Refer to the Appendix, Figure 21.

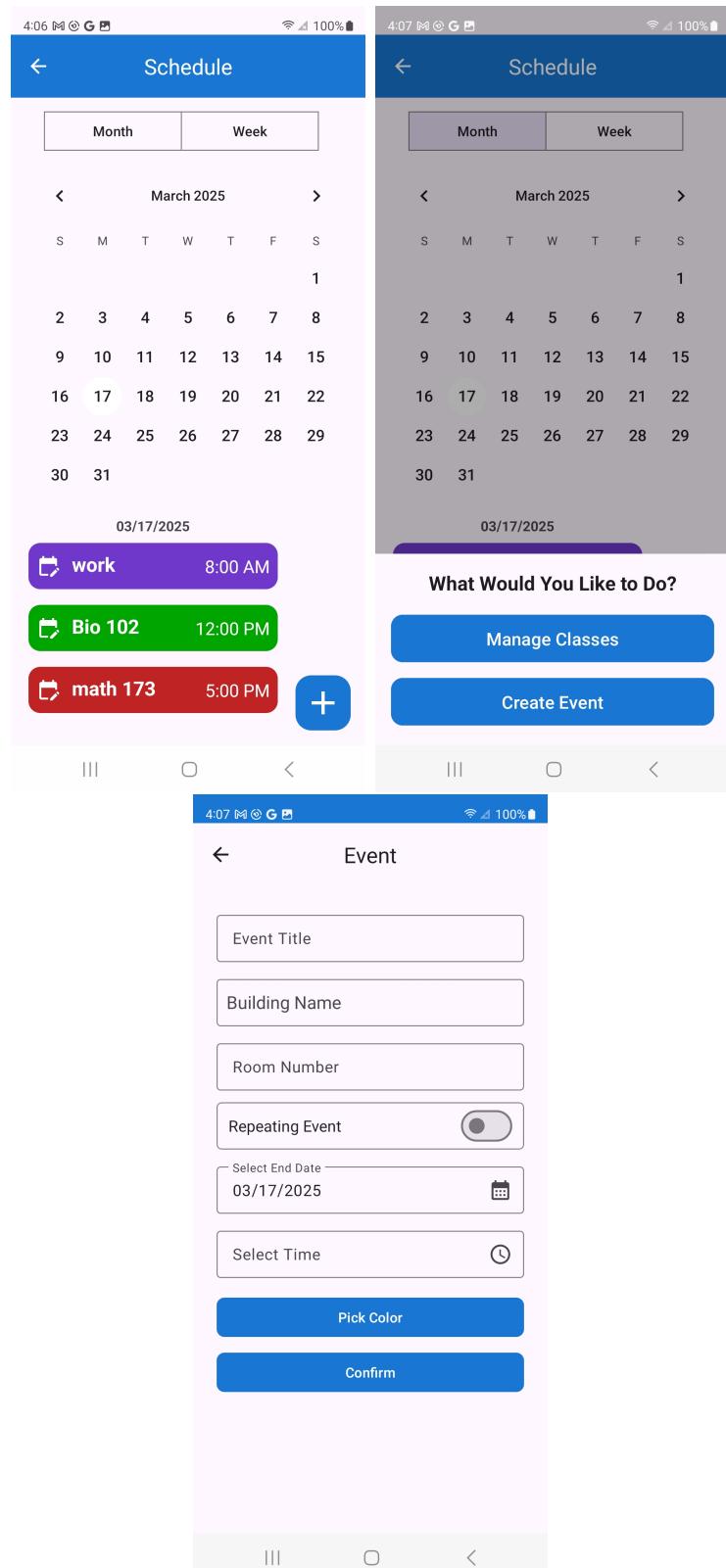


Figure 4: User Interface Mockup: Create Event

3.4 Use Case 4: Edit Event

Description: The user modifies an existing event within the schedule.

- **Functional Requirements:**

- 3.4.1 The selected event needs to be on the schedule.
- 3.4.2 The date should be changeable.
- 3.4.3 The name of the event should be changeable.
- 3.4.4 The building of the event should be changeable.
- 3.4.5 All info being edited needs to be validated prior to changing the existing event.
- 3.4.6 Entering or selecting invalid info should display an error message.

- **User Interface:**

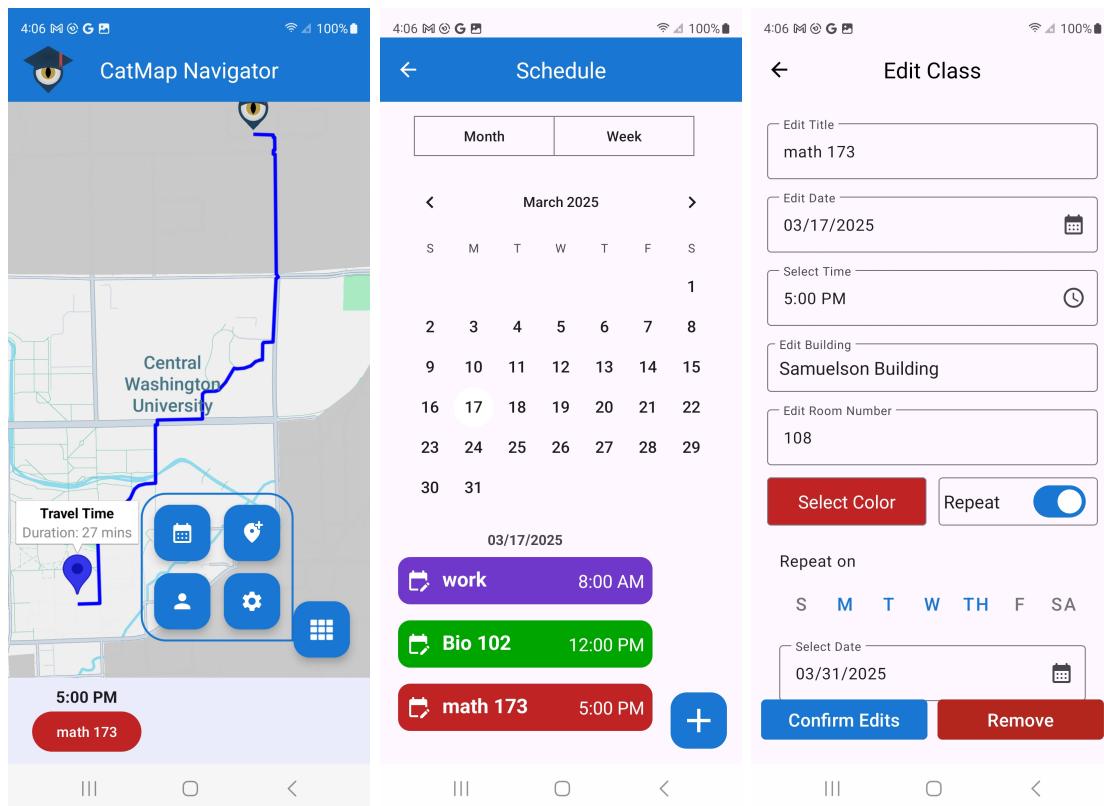


Figure 5: User Interface Mockup: Edit Event

- **Software Interfaces:** Refer to the Appendix, Figure 21.

3.5 Use Case 5: Remove Event

Description: Allows the user to delete a specific event from their schedule.

- **Functional Requirements:**

- 3.5.1 The selected event needs to be on the user's schedule or saved onto the favorites tab.
- 3.5.2 The user should click the desired event they wish to remove.
- 3.5.3 After selecting the event, the user should click the remove button to remove it.

- **User Interface:**

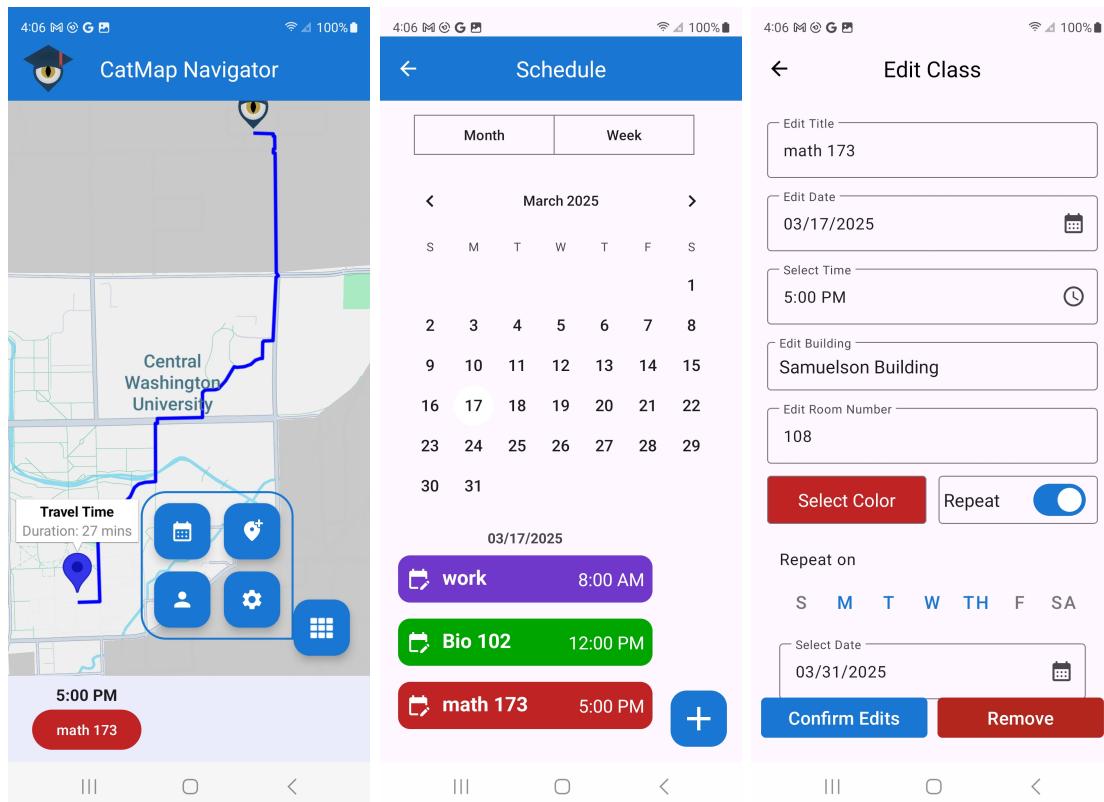


Figure 6: User Interface Mockup: Remove Event

- **Software Interfaces:** Refer to the Appendix, Figure 21.

3.6 Use Case 6: Change Preferences

Description: The user will be able to configure their preferences such as light or dark mode, notification settings, etc.

- **Functional Requirements:**

- 3.6.1 The miscellaneous icon needs to be created prior to accessing the preference settings.
- 3.6.2 The settings icon needs to be created prior to accessing the preference settings.
- 3.6.3 After clicking on the miscellaneous icon, four icons should appear above the map for a settings icon, location icon, profile icon, and schedule icon.
- 3.6.4 When the settings icon is clicked, a new window should appear showing the user preference settings.
- 3.6.5 After accessing the preference settings, the app should allow the user to enable dark mode through the window.

- **User Interface:**

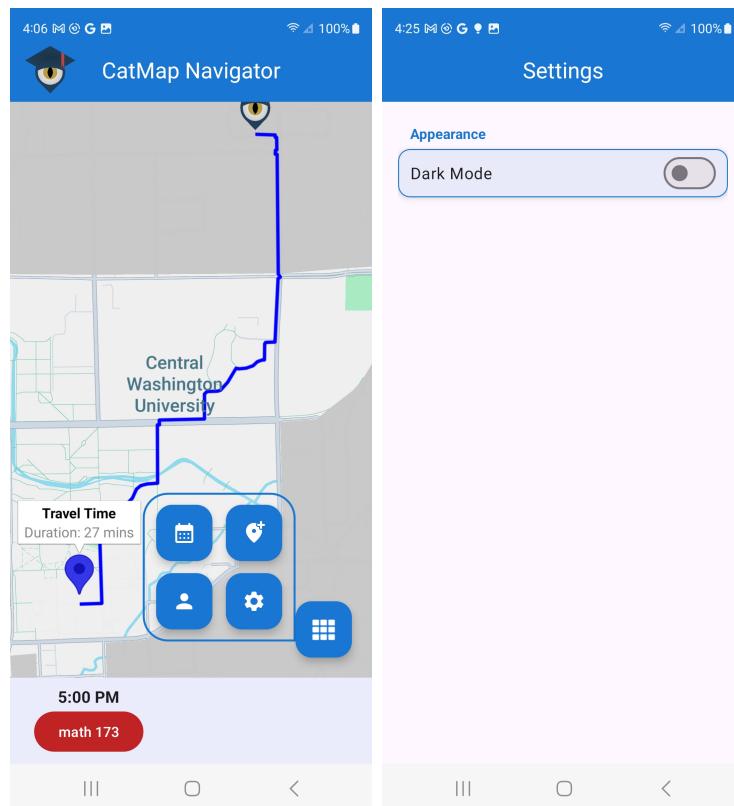


Figure 7: User Interface Mockup: Change Preferences

- **Software Interfaces:** Refer to the Appendix, Figure 22.

3.7 Use Case 7: View Directions

Description: The user clicks/taps on a scheduled class link (or building name) to go to the class.

- **Functional Requirements:**

- 3.7.1 The map needs to be visualized before directions can be created.
- 3.7.2 Based on upcoming events, a path should start to be created showing the shortest distance to reach the upcoming event.
- 3.7.3 A line should be drawn on the map showing the shortest distance.
- 3.7.4 The user should be able to map out a line for specific events to navigate to favorites or upcoming events.

- **User Interface:**



Figure 8: User Interface Mockup: View Directions

- **Software Interfaces:** Refer to the Appendix, Figure 19.

3.8 Use Case 8: View Map

Description: The user will be able to view the CWU campus map with relevant roads and buildings depicted to the user.

- **Functional Requirements:**

- 3.8.1 The campus map will include a top-down view of the campus, and the user's location.
- 3.8.2 The campus map will NOT include buildings outside of the campus.
- 3.8.3 The campus interactive map will be used to help determine what is and isn't on campus.

- **User Interface:**

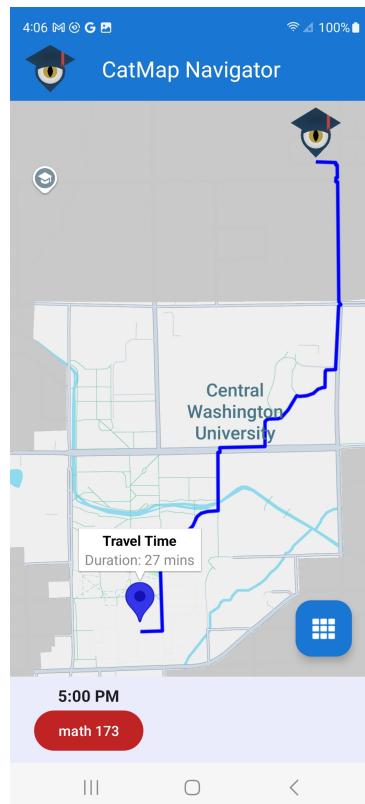


Figure 9: User Interface Mockup: View Map

- **Software Interfaces:** Refer to the Appendix, Figure 19.

3.9 Use Case 9: View Favorites

Description: Allows the user to see their selected favorite locations within the campus.

- **Functional Requirements:**

- 3.9.1 The miscellaneous icon should be created prior to accessing the favorites window.
- 3.9.2 After clicking on the miscellaneous icon, four icons should appear above the map for a settings icon, location icon, profile icon, and schedule icon.
- 3.9.3 After clicking on the location icon, a new window will appear with a list of locations and a favorites tab.
- 3.9.4 The app will depict the user's favorite locations or search for a new one if they desire.

- **User Interface:**

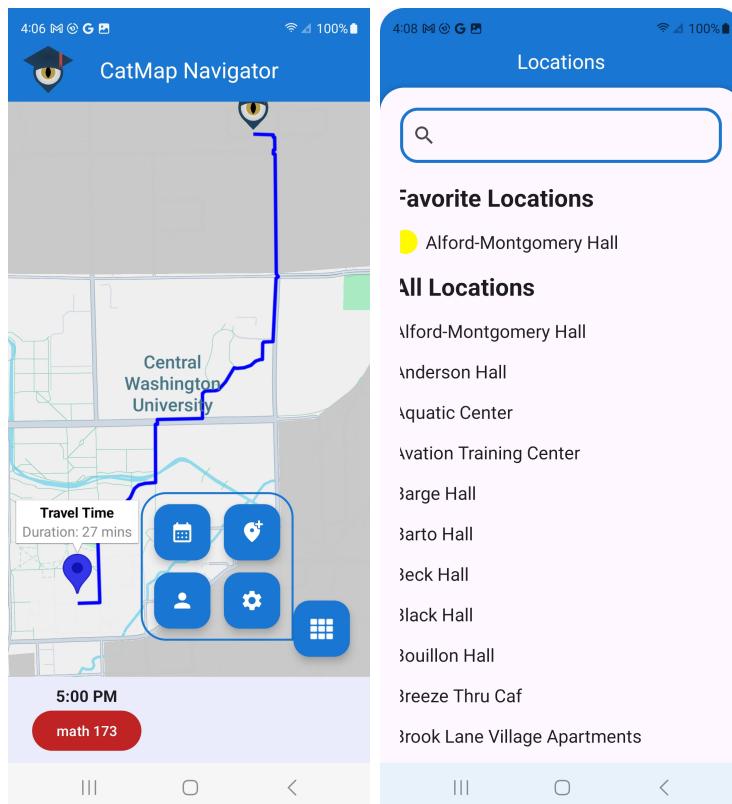


Figure 10: User Interface Mockup: View Favorites

- **Software Interfaces:** Refer to the Appendix, Figure 23.

3.10 Use Case 10: Modify Favorites

Description: Allows the user to add non-class event locations to a favorites tab to be revisited later on.

- **Functional Requirements:**

- 3.10.1 The miscellaneous icon should be created prior to accessing the favorites window.
- 3.10.2 After clicking on the miscellaneous icon, four icons should appear above the map for a settings icon, location icon, profile icon, and schedule icon.
- 3.10.3 After clicking on the location icon, a new window will appear with a list of locations and a favorites tab.
- 3.10.4 The app will depict the user's favorite locations or allow them to search for a new one.
- 3.10.5 After searching for a location, the user will be able to add the selected location to their favorites via an "Add to Favorites" button.
- 3.10.6 The app will also allow the user to remove a favorite location.

- **User Interface:**

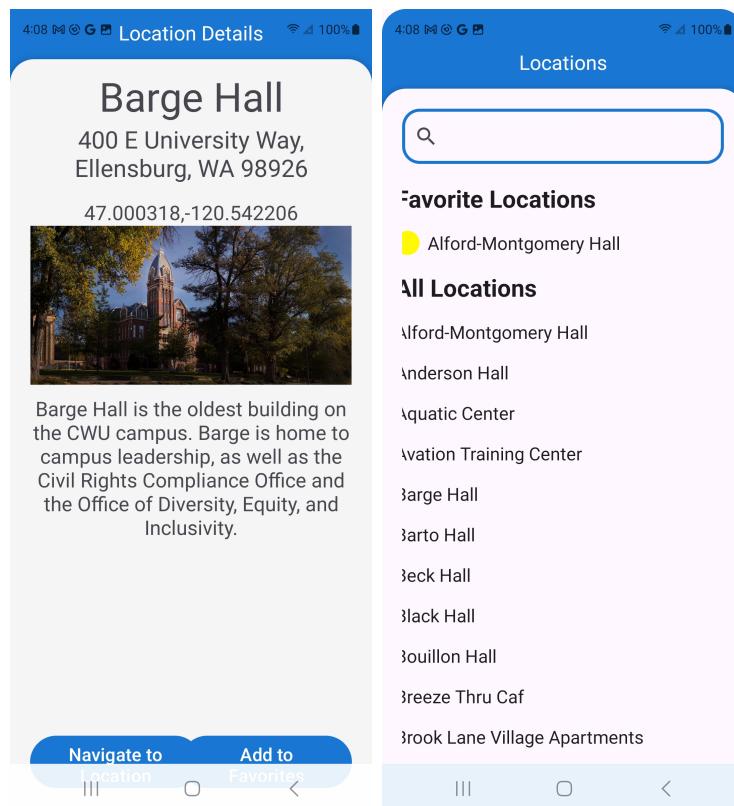


Figure 11: User Interface Mockup: Modify Favorites

- **Software Interfaces:** Refer to the Appendix, Figure 23.

3.11 Use Case 11: View Location Information

Description: Allows the user to view information on specific buildings around the campus such as addresses and room numbers.

- **Functional Requirements:**

- 3.11.1 The user will be able to click a location from the locations menu to see more information about it.
- 3.11.2 The user will be able to view a picture of the location.
- 3.11.3 The user will be able to read a description of the location.
- 3.11.4 The user will be able to see opening and closing times of the location.
- 3.11.5 The user will be able to click a button to begin navigation to the selected location.
- 3.11.6 The user will be able to click a button to add the location to the favorites list.

- **User Interface:**

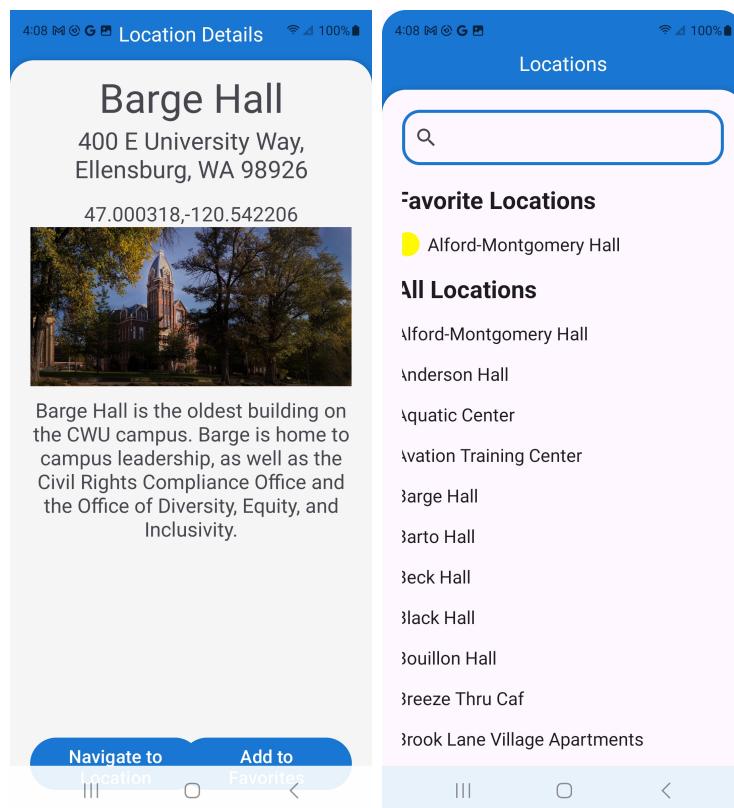


Figure 12: User Interface Mockup: View Location Information

- **Software Interfaces:** Refer to the Appendix, Figure 23.

3.12 Use Case 12: Search Locations

Description: Allows the user to search for a specific building or location around the campus based on the building name.

- **Functional Requirements:**

- 3.12.1 The user will be able to search for a specific location using the search bar.
- 3.12.2 Clicking the search bar will take the user to the locations page, autofocus into the search bar, and display the keyboard.
- 3.12.3 As the user types a location name, the search feature will present the user with locations that match the location name so far.

- **User Interface:**

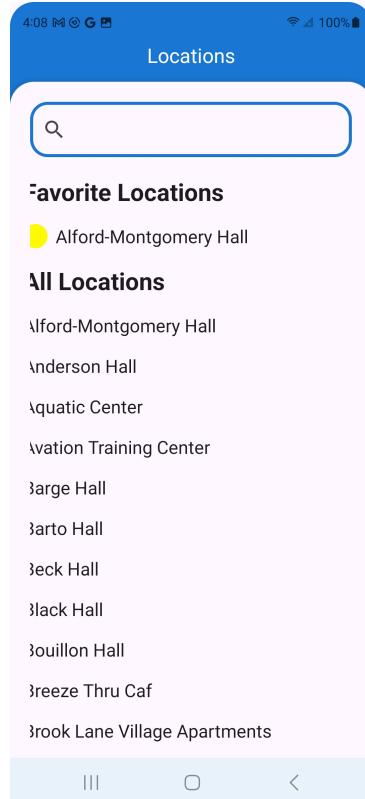


Figure 13: User Interface Mockup: Search Locations

- **Software Interfaces:** Refer to the Appendix, Figure 23.

3.13 Use Case 13: Navigate to Locations

Description: Allows the user to navigate to a selected location within the application.

- **Functional Requirements:**

- 3.13.1 A path will be drawn from the user's location to the destination.
- 3.13.2 A pin icon will represent the user's location.
- 3.13.3 A pin with a circular picture of the destination will represent the destination.
- 3.13.4 The path will be made of connected segments that lead the user to the destination.
- 3.13.5 Paths will be drawn from the user to the next scheduled event, or if the user wants to go to a specific unscheduled location, the path will be drawn from the user to a desired location and then finally to the next scheduled event.

- **User Interface:**

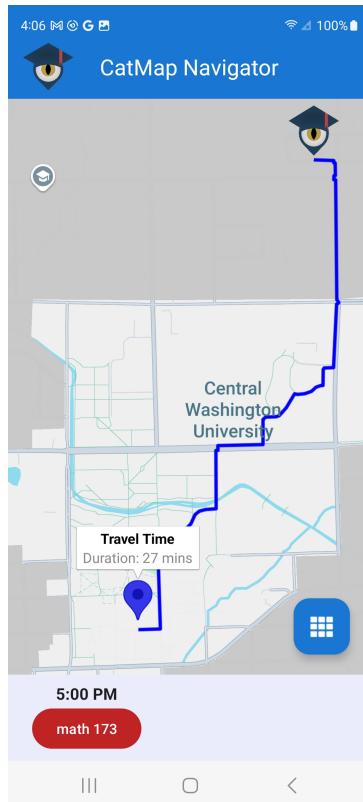


Figure 14: User Interface Mockup: Navigate to Locations

- **Software Interfaces:** Refer to the Appendix, Figure 19.

3.14 Use Case 14: View Calendar

Description: Allows the user to view the calendar and the schedule associated with a specific day on that calendar.

- **Functional Requirements:**

- 3.14.1 A calendar of the current month will be presented when the user clicks the schedule icon on the home screen.
- 3.14.2 There will be arrows at the top left and right to allow the user to change the month displayed.
- 3.14.3 When the user clicks on a day within the calendar, events scheduled for that day will be shown below the calendar with their name and start time.
- 3.14.4 The date and day of the week will be shown when the user clicks a day.

- **User Interface:**

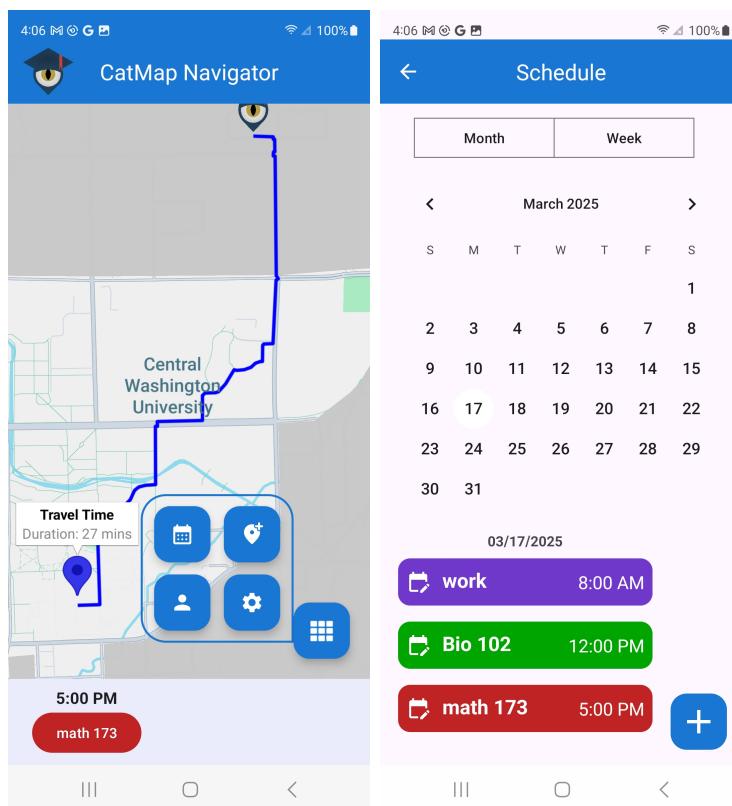


Figure 15: User Interface Mockup: View Calendar

- **Software Interfaces:** Refer to the Appendix, Figure 21.

3.15 Use Case 15: View Weekly Schedule

Description: The user will be able to view scheduled events for the current week.

- **Functional Requirements:**

- 3.15.1 The current week's schedule will be shown when the user clicks "View Weekly Schedule" in the schedule page.
- 3.15.2 The weekly schedule will list the seven days of the current week and their dates vertically.
- 3.15.3 Scheduled events for each day will be listed below the dates.
- 3.15.4 The user will be able to see more information about each event by clicking them.

- **User Interface:**

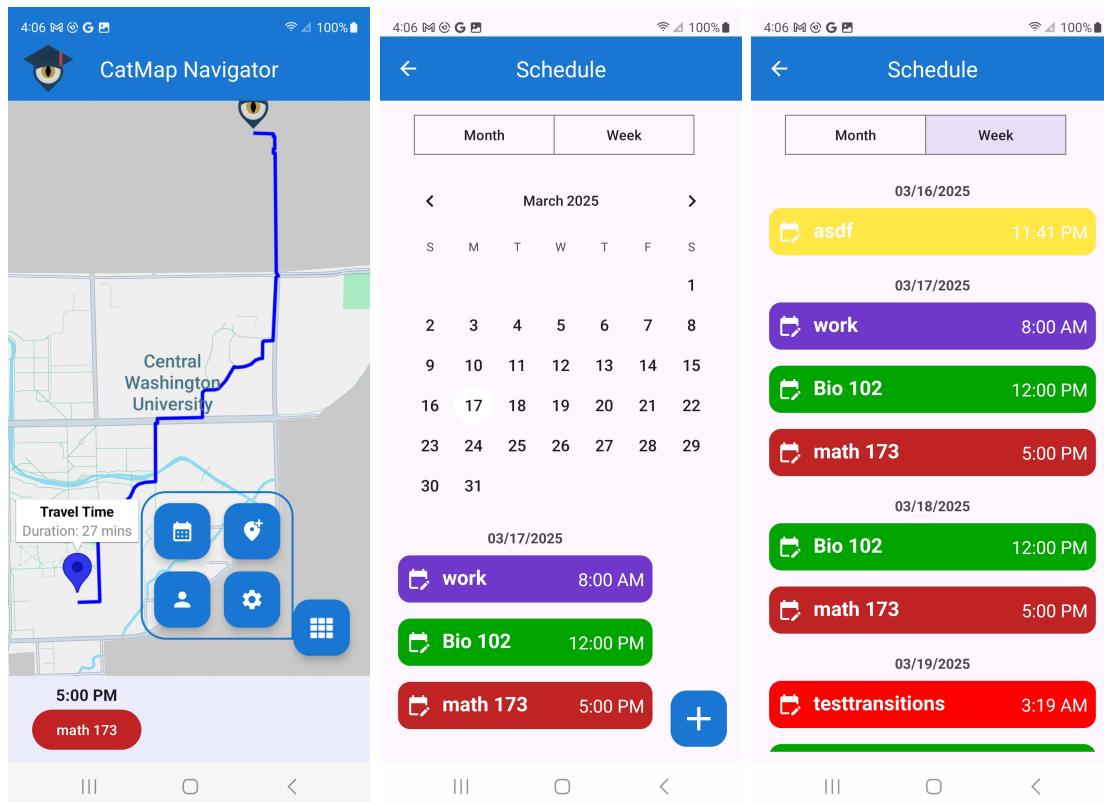


Figure 16: User Interface Mockup: View Weekly Schedule

- **Software Interfaces:** Refer to the Appendix, Figure 21.

3.16 Use Case 16: View Daily Schedule

Description: Shows a generalized version of the daily schedule showing events, times, and when they are about to start.

- **Functional Requirements:**

- 3.16.1 The daily schedule will show the user the name and time period of upcoming events in queue.
- 3.16.2 The daily schedule will display the events scheduled for the current day as rounded rectangles containing the event name and their starting time.
- 3.16.3 The queue will be horizontally scrollable, allowing the user to see all events of the current day.

- **User Interface:**



Figure 17: User Interface Mockup: View Generalized Daily Schedule

- **Software Interfaces:** Refer to the Appendix, Figure 24.

3.17 Use Case 17: Modify Class

Description: Allows the user to modify a class in their schedule.

- **Functional Requirements:**

- 3.17.1 A valid date on the calendar needs to be selected prior to adding the event.

3.17.2 A valid building on CWU campus needs to be selected prior to adding the event.

3.17.3 The event title, such as a subject name, store name, etc., needs to be typed out prior to adding the event.

3.17.4 After all info is validated, the event should be added to the user's schedule.

3.17.5 An error message should display if any info selected or typed in is invalid.

- **User Interface:**

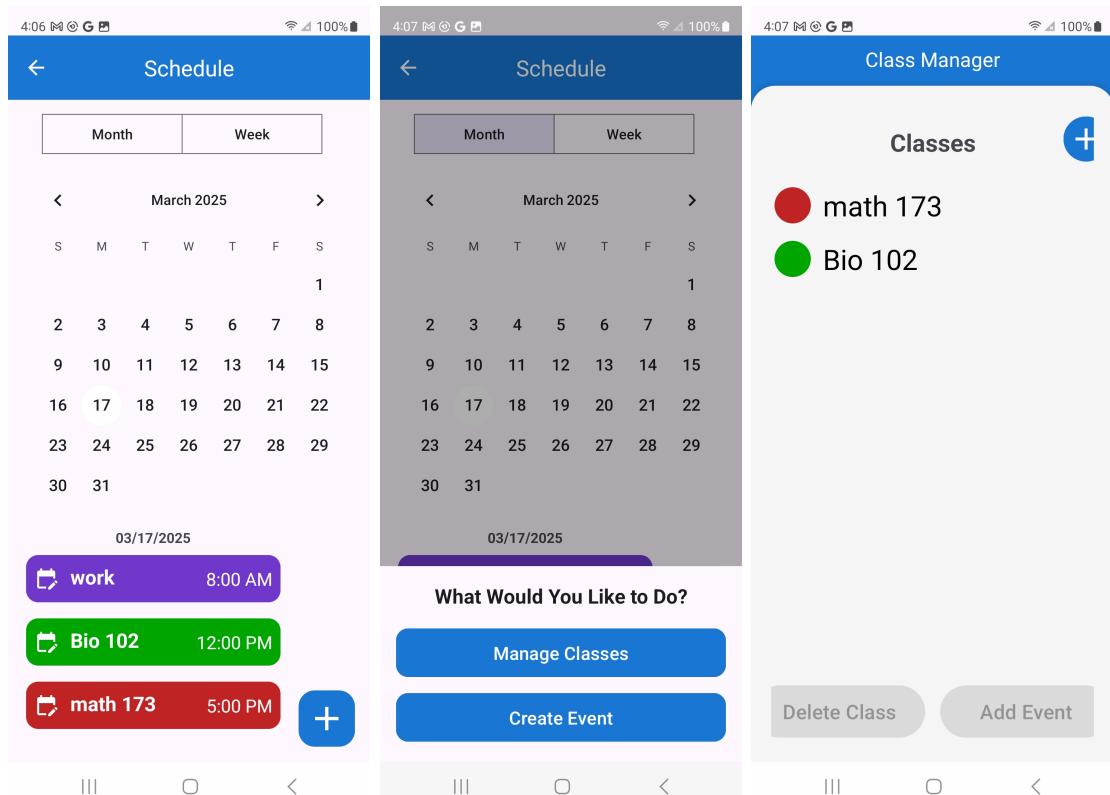


Figure 18: User Interface Mockup: Modify Class

- **Software Interfaces:** Refer to the Appendix, Figure 21.

4 Traceability Matrix

Use Case	Req. Num	Requirement Description	Design	Test	Status
			Comp.	Case	
1. Sign Up / Login	1.1	User can sign up			
	1.2	User can login			
	1.3	User can login with Google account			
	1.4	Saving user account schedule, favorites, etc, with Firestore			
	1.5	User data can be retrieved on any device running the app that user logs into			
	1.6	User created accounts will be stored in and retrieved from the database			
2. Edit Profile	2.1	User can change profile picture			
	2.2	User can change name			
	2.3	User can change e-mail			
	2.4	User can change password			
	2.5	User profile is retrieved from DB			
	2.6	Profile changes will be done on Firestore			
	2.7	Profile changes are done locally, then on the DB			
	2.8	Profile change queries are attempted max 5 times before failure			
	2.9	Profile changes are retrieved from DB			
3. Create Event	3.1	User can only add events if they're not conflicting			
	3.2	Events needs valid dates			
	3.3	Events need a valid building			
	3.4	Events need a title			
	3.5	Event is successfully added			
	3.6	Error message for failing to add events			
	3.7	Adding events interfaces with Google Maps			
3. Edit Event	4.1	Events needs to be on the schedule to edit			
	4.2	Edited event cannot conflict with other events			
	4.3	Event dates can be edited			
	4.4	Event names can be edited			
	4.5	Event buildings can be edited			
	4.6	Edited event details need to be validated			
	4.7	Display error message for invalid event edits			
5. Remove Event	4.8	Editing events interfaces with Google Maps			
	5.1	Events needs to be on the schedule or favorites to be removed			
	5.2	User can remove events by clicking			
6. Change Preferences	5.3	Events have a removal button			
	6.1	Misc. icon is loaded before accessing prefs. settings			
	6.2	Settings icon is loaded before accessing prefs. settings			
	6.3	Clicking misc. icon pulls up schedule, locations, and settings icon			
	6.4	Prefs. setting can be accessed via settings icon			
	6.5	Acessibility options can be turned on via settings			
	6.6	Acessibility options are turned on using a switch or slider			
7. View Directions	6.7	Notification settings are changed through a separate window			
	7.1	Map needs to be loaded before paths			
	7.2	Upcoming events always show a path			
	7.3	Event paths is a dashed line			
	7.4	Users can create custom paths			
	7.5	Building info is retrieved from Google Maps			
	7.6	Building visualization is done with Google Maps			
	7.7	Directions are created from Google Maps			
8. View Map	7.8	Paths are created from Google Maps			
	8.1	Campus Map shows topdown view			
	8.2	Map shows user location			
	8.3	Map restricts to Campus area			
	8.4	Map determines Campus area			
	8.5	Campus buildings are stored from Google Maps			
9. View Favorites	8.6	Google Maps helps reconstruct campus visually			
	9.1	Misc. icon is accessed before favorites window			
	9.2	Clicking Misc. icon shows settings, location, and schedule icon			
	9.3	Clicking Locations icon pops up Location window			
	9.4	The app will depict the users' favorite locations or search for a new one if they desire.			
	9.5	CPU Campus building is stored from Google Map			

10. Modify Favorites	10.1	Misc. icon is loaded before favorites window			
	10.2	Clicking Misc. icon shows settings, location, and schedule icon			
	10.3	Clicking Locations icon pops up Locations window			
	10.4	The app will depict user fav. locations or search for a new one			
	10.5	User can add a location to favorites via button			
	10.6	User can remove a location from their favorites via button			
	10.7	CPU Campus building is stored from Google Map			
11. View Location Information	11.1	User can view detailed location info			
	11.2	User can view location picture			
	11.3	User can view location description			
	11.4	User can view location opening and closing time			
	11.5	User can view path to location			
	11.6	User can add location to favorites via button			
12. Search Locations	12.1	Users can use search bar for specific locations			
	12.2	Search bar redirects to locations page and displays keyboard for search			
	12.3	Search bar lists locations as users type			
13. Navigate to Location	13.1	A path from user location to location is drawn			
	13.2	A pin icon represents the user's location			
	13.3	Locations will have circular pictures			
	13.4	Paths will be segmented			
	13.5	Paths will be to the latest event, or user custom location then latest event			
	13.6	Paths will be drawn using Google Maps			
	13.7	A node-oriented pathfinding algorithm will be used if Google Maps isn't robust			
	13.8	GPS may be used to retrieve user location			
14. View Calender	14.1	Clicking schedule icon shows current month calender			
	14.2	Calender can switch to other months			
	14.3	Clicking on a day will show scheduled events			
	14.4	Clicking on a day displays date			
	14.5	Google Account events can be imported			
	14.6	Google Account integration will use Google Calender API			
15. View Weekly Schedule	15.1	Weekly schedule is shown			
	15.2	Weekly schedule lists 7 days vertically			
	15.3	Schedule events will be listed			
	15.4	User can click on an event for info			
16. View Generalized Daily Schedule	16.1	Daily timeline will show upcoming and past events			
	16.2	Events are displayed as rectangles			
	16.3	Daily timeline centers on current time with a red vertical bar			
	16.4	Daily timeline shows a span of 5 hours			
	16.5	Daily timeline is scrollable for all 24 hours			
	16.6	Daily timeline centers back after scrolling in 3s			
	16.7	Daily timeline can be expanded and opens extended daily view			
	16.8	Daily timeline scrolling is smooth			
17. View Extended Daily Schedule	17.1	Daily timeline can be expanded and opens extended daily view			
	17.2	Extended daily view contains all info in daily view, and more specifics			
	17.3	Extended daily view displays events with taller rectangles			
	17.4	Extended daily view displays date and is scrollable			
	17.5	Extended timeline scrolling is smooth			
18. Add Class	18.1	Classes cannot have conflicting times			
	18.2	Classes need a valid date			
	18.3	Classes need a valid building			
	18.4	Class info needs to be validated prior to adding			
	18.5	Event will be added after info validation			
	18.6	Error message for inputting invalid info			
	18.7	Building name is retrieved from Google Maps			

Appendix

Figure 19: Sequence Diagram: Sign Up / Login, View Directions, View map, and Navigate to Locations

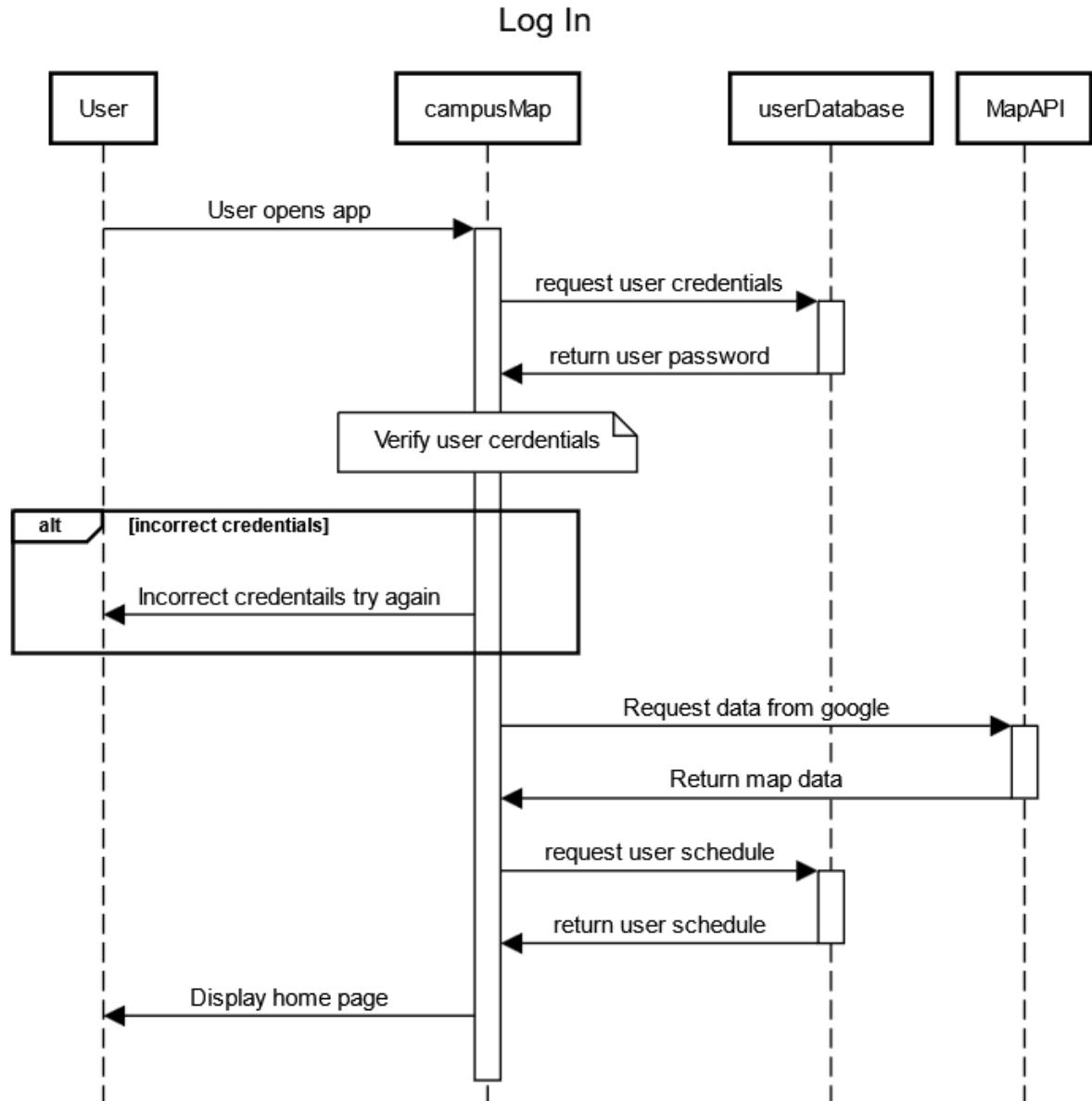


Figure 20: Sequence Diagram: Edit Profile

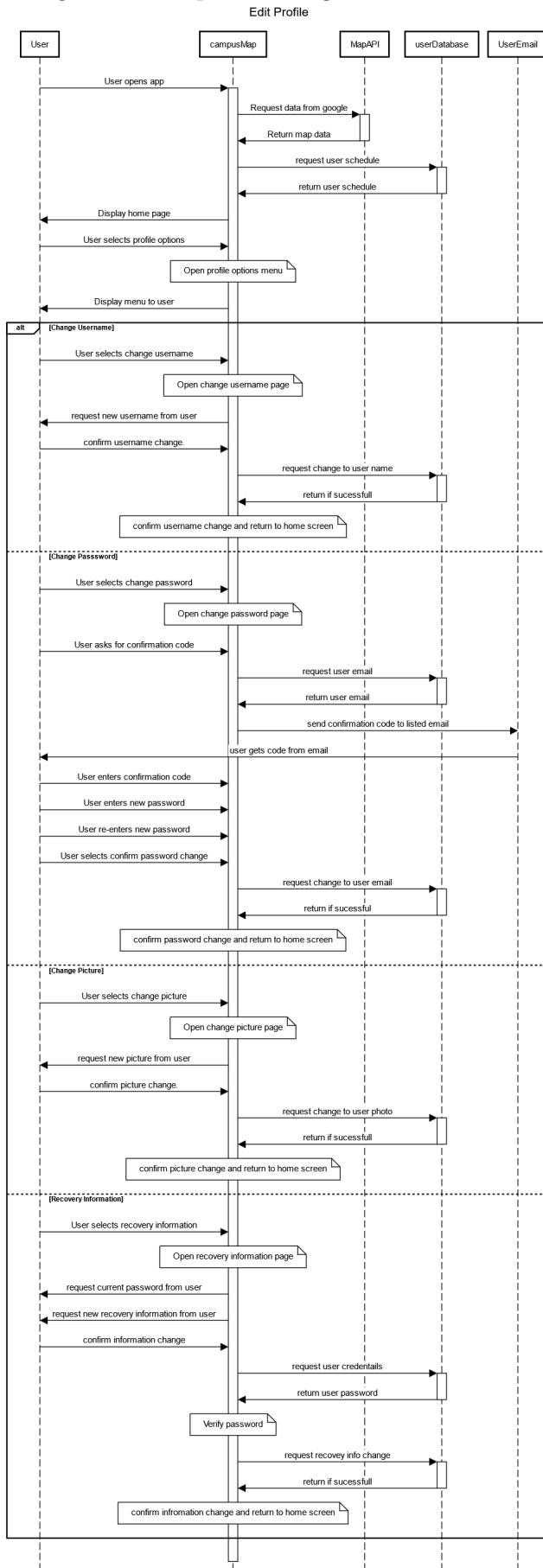


Figure 21: Sequence Diagram: Add Class, Edit Event, Remove Event, and Create Event

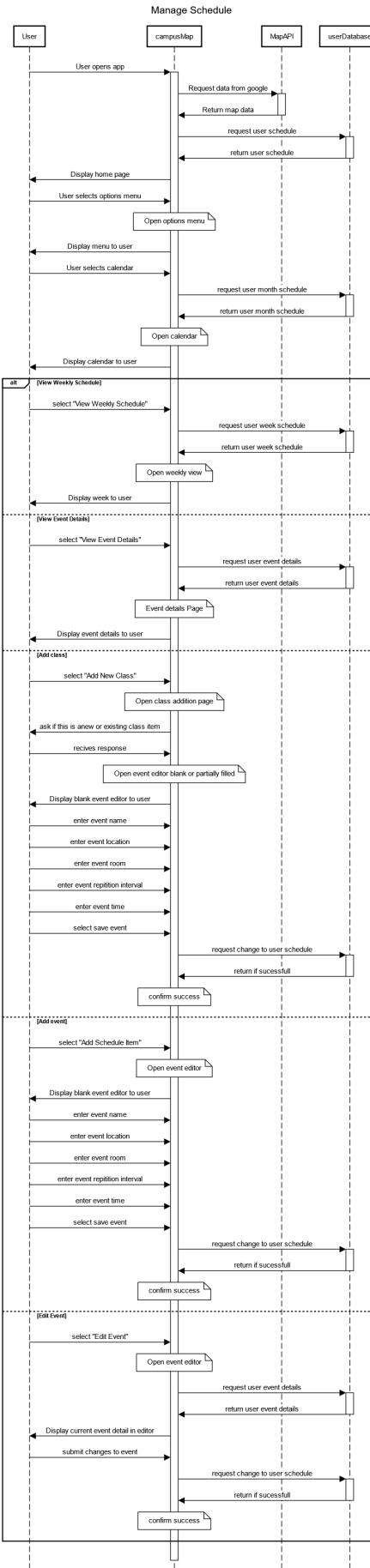


Figure 22: Sequence Diagram: Change Preferences

Change Preferences

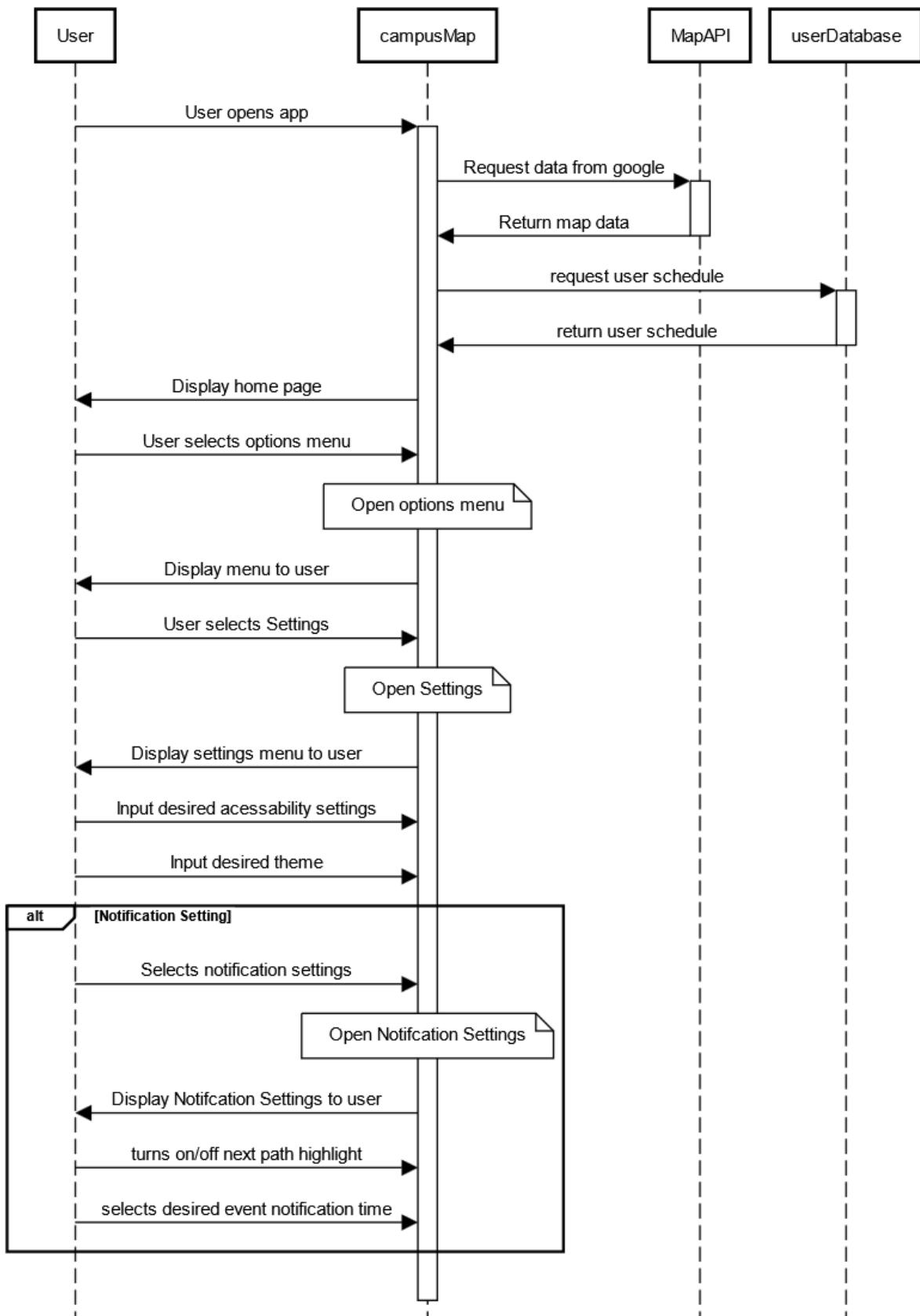


Figure 23: Sequence Diagram: View Favorites, Modify Favorites, View Location Information, Search Locations

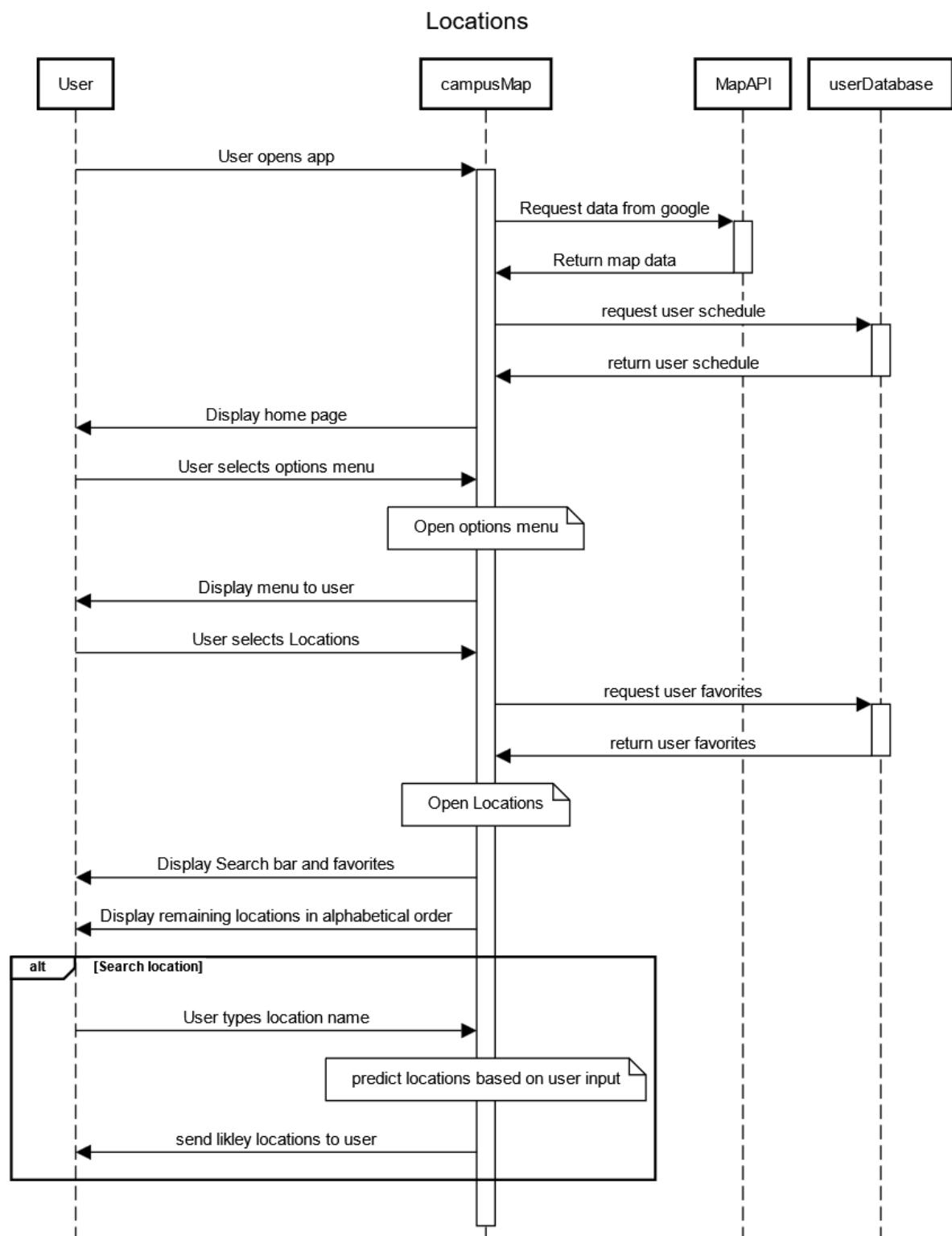


Figure 24: Sequence Diagram: View Generalized Daily Schedule
View Daily Schedule

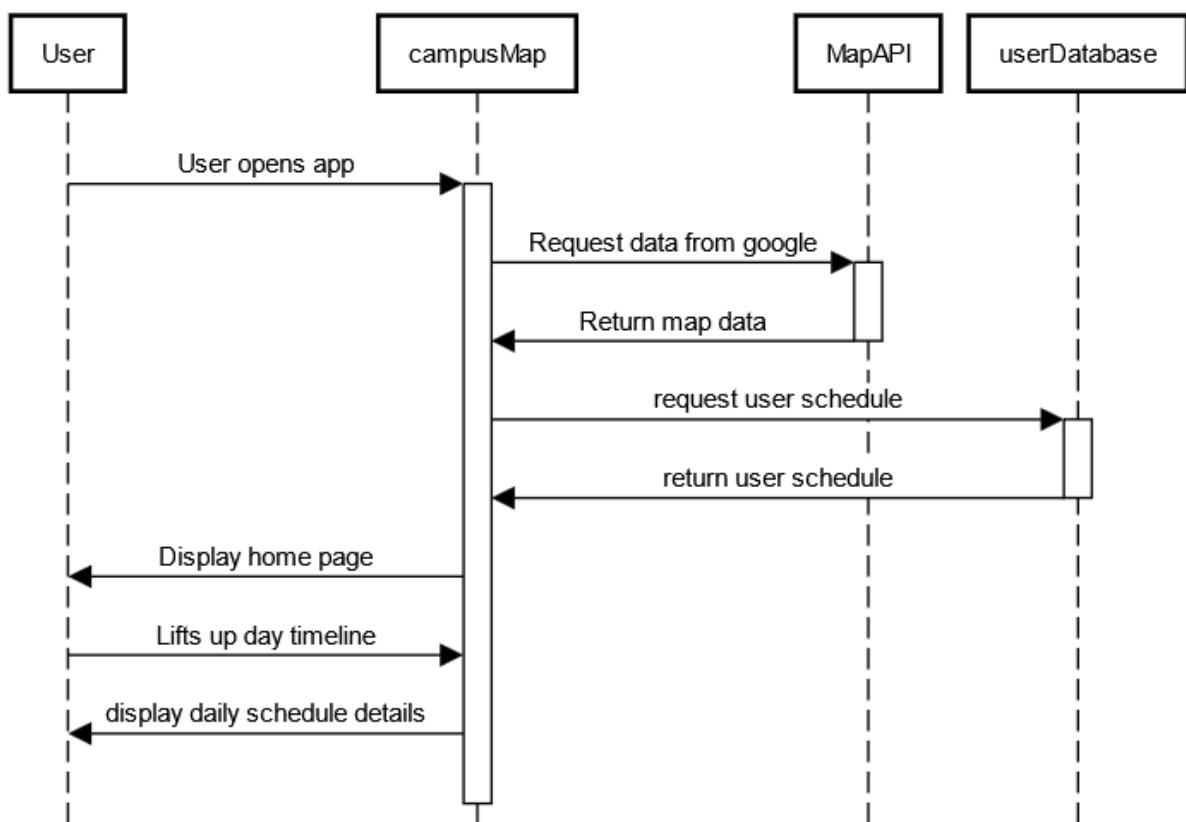


Figure 25: Data Flow Diagram

- DATA FLOW DIAGRAM (DFD) for CWU Campus Navigation App

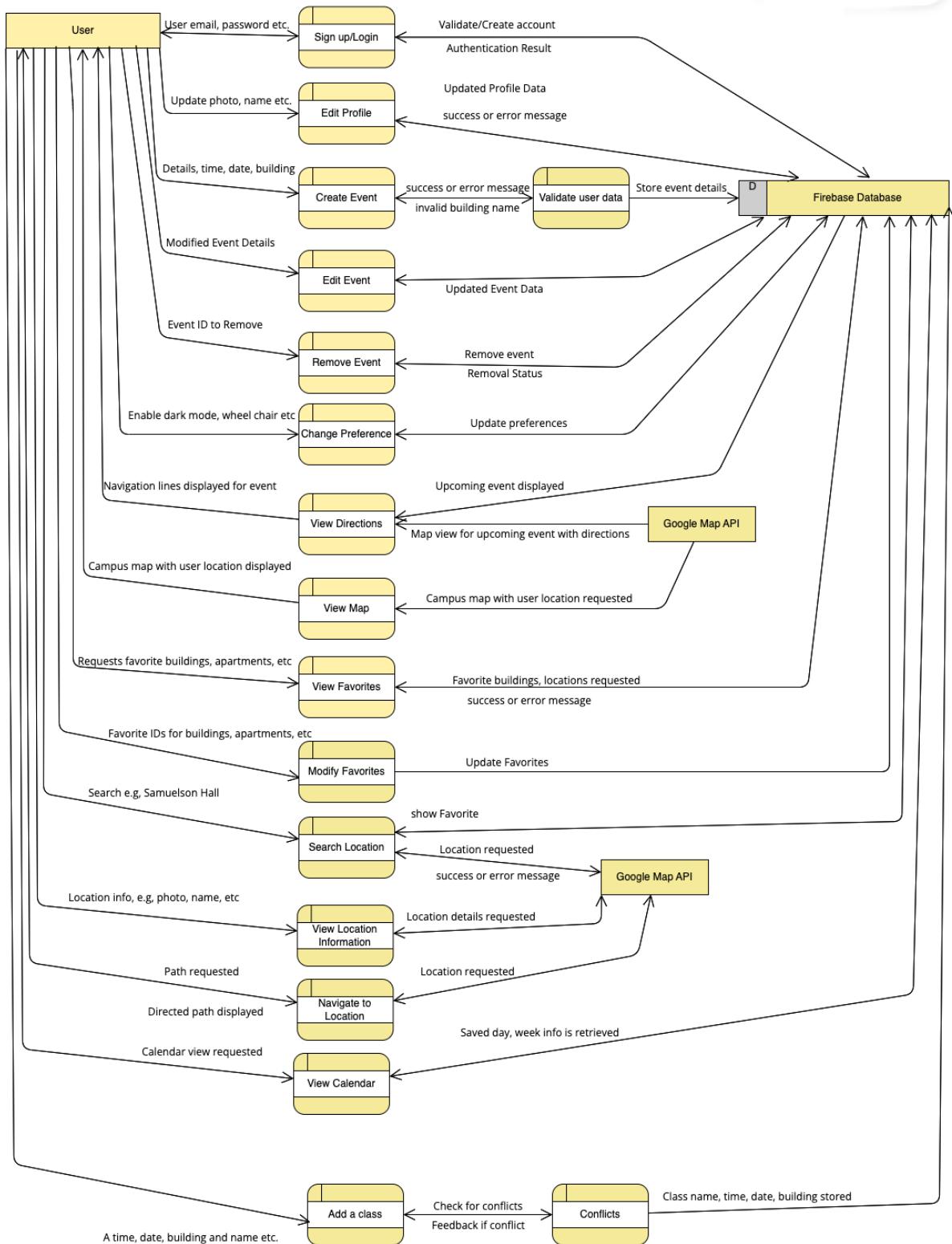


Figure 26: UML Diagram

