Jack of All Travel

Milestone 1 Report

Jessica Fang, Jessica Liu, Rohin Aulakh, Nicole Stuart

https://github.com/CMPT-276-SPRING-2025/final-project-16-moons

API Selection and Usage

Google Calendar API

- **Description:** The Google Calendar API will allow calendar functionality to be integrated inside our web application. It will support event creation, reminders and notifications, and sharing across devices as key features.

- **Usage:** The API will be used to allow users to schedule events through an interactive calendar in order to manage their limited travel time effectively. Setting reminders and establishing a schedule are very important while on vacation, our web app will allow ease of use to a virtual calendar that makes viewing and scheduling plans easy and accessible to those joining you on vacation.

Google Maps API

- **Description:** The Google Maps API provides location based services through a map interface. Key features include navigation, route planning, estimated travel times and finding nearby attractions and restaurants.

- **Usage:** In our project, the API will be used to allow users to search for anything available on the Google Maps system that is nearby to them. A user could search for walking/transit routes around town or find dining options, accommodations and services nearby to them. The virtual map will allow users to become familiar with their destination before arriving and allow them to make informed decisions either on the go or while planning.

Feature Descriptions

Google Calendar Features

1. **Notifications for Upcoming Events**
    - Description: Users can choose to be notified for upcoming travel related events such as flights, reservations, bookings, or check-ins. Notifications can be customized based on time and urgency to ensure you never miss anything.
    - Benefit: Helps users stay on top of their schedule and keep them organized. Rather than switching between hotel, tour, and airline apps, users can receive all wanted notifications through one calendar interface. Allows users to never miss any events they may have been looking forward to, whilst still enjoying their vacation.
2. **Sharing Calendar with Friends**
    - Description: Users can share their calendars with their travel companions, allowing easy collaboration and communication between all travellers. A shared calendar makes event coordination easier and keeps everyone on the same page.

- Benefit: Rather than individually messaging everyone before leaving or dealing with data and poor signal while abroad, Jack of All Travel's calendar functionality adds the ability to add all wanted trip details in one place. This is good for reducing confusion between travelling companions and saving all users from worrying about messaging or calling others in order to set up plans. As they can all access and see a universal calendar.

3. **Adding Locations & Flights**
    - Description: Directly communicating with the Google Maps API, users can save locations, transit info and flight information to their calendar. These could include addresses, transit schedules and possible travel times.
    - Benefit: Allows users to set it and forget it, instead of repeatedly searching for directions or locations, just save the info into your calendar and access it there when you need it. Combines with the sharing feature to make directions or searched locations easily accessible between multiple people, minimizing confusion. If the user's group was looking for a pizza place to eat dinner at, they could easily search for it through the Google Maps integration and add its address and the time they want to be there into the calendar for others to see.

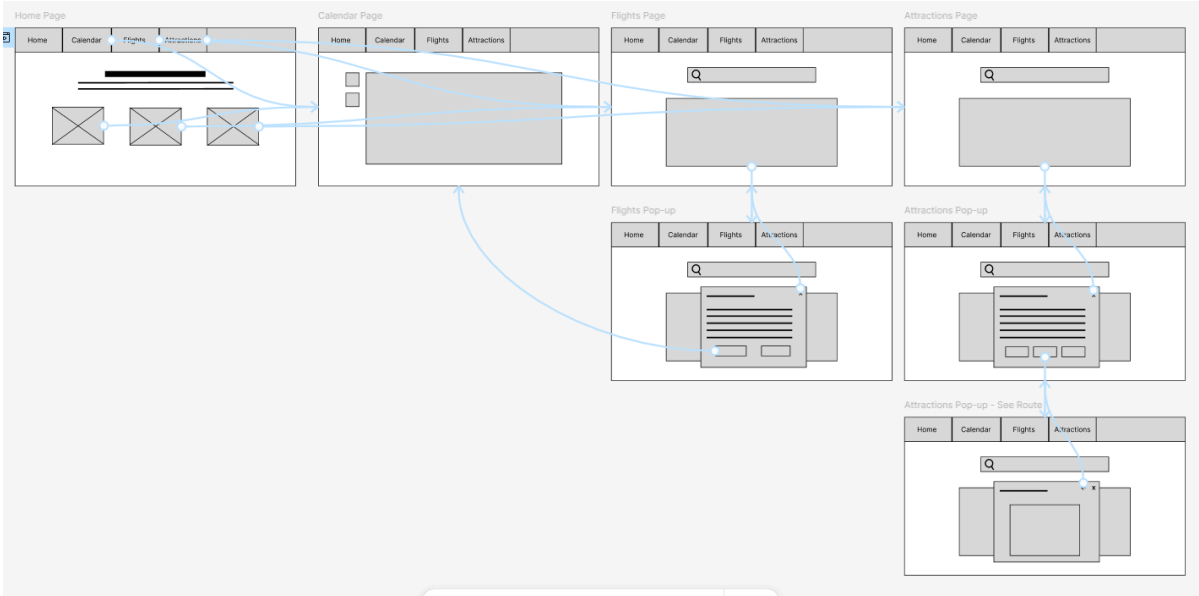
Google Maps Features

1. **Flight and Transit Information**
    - Description: Users can search for relevant flight, travel, routing and other transit options between destinations. The Jack of All Travel app will display departure and arrival times, estimated travel durations, potential ticket prices and redirection to booking links to finalize the purchase.
    - Benefit: Instead of comparing between numerous airline, train, bus, and travel agencies for the correct travel option. Users will be able to find all relevant travel options inside the app allowing for quick comparisons and a more efficient time planning flights and transit.
2. **Hotel and Accommodations**
    - Description: Compare hotel and other accommodations based on a multitude of factors including, pricing, location, ratings and amenities. Expanded filters allow users to find the correct option for them based on their stay.
    - Benefit: Similar to finding the correct flight, travellers often must switch between numerous websites comparing hotels manually. Our web app serves to streamline this process and allow users to swiftly find, decide and book lodging options without the need to switch between different sites.
3. **Recommended Dining and Services**
    - Description: Based on a user search or location data, recommended restaurants and dining options nearby can be shown. Users are able to filter the results depending on what they want to eat or they can alter their search to find another service entirely that is nearby to them.
    - Benefit: Finding good places to eat can be very time consuming, especially when in unfamiliar areas. Rather than going to the nearest fast food chain or ending up at a poor restaurant. Users can search for what they want and decide based on

filters and reviews. This feature is also useful for other services such as grocery stores, malls, or attractions, making it easy to find quality places nearby to the user when requested.

Mid-Fidelity Prototype

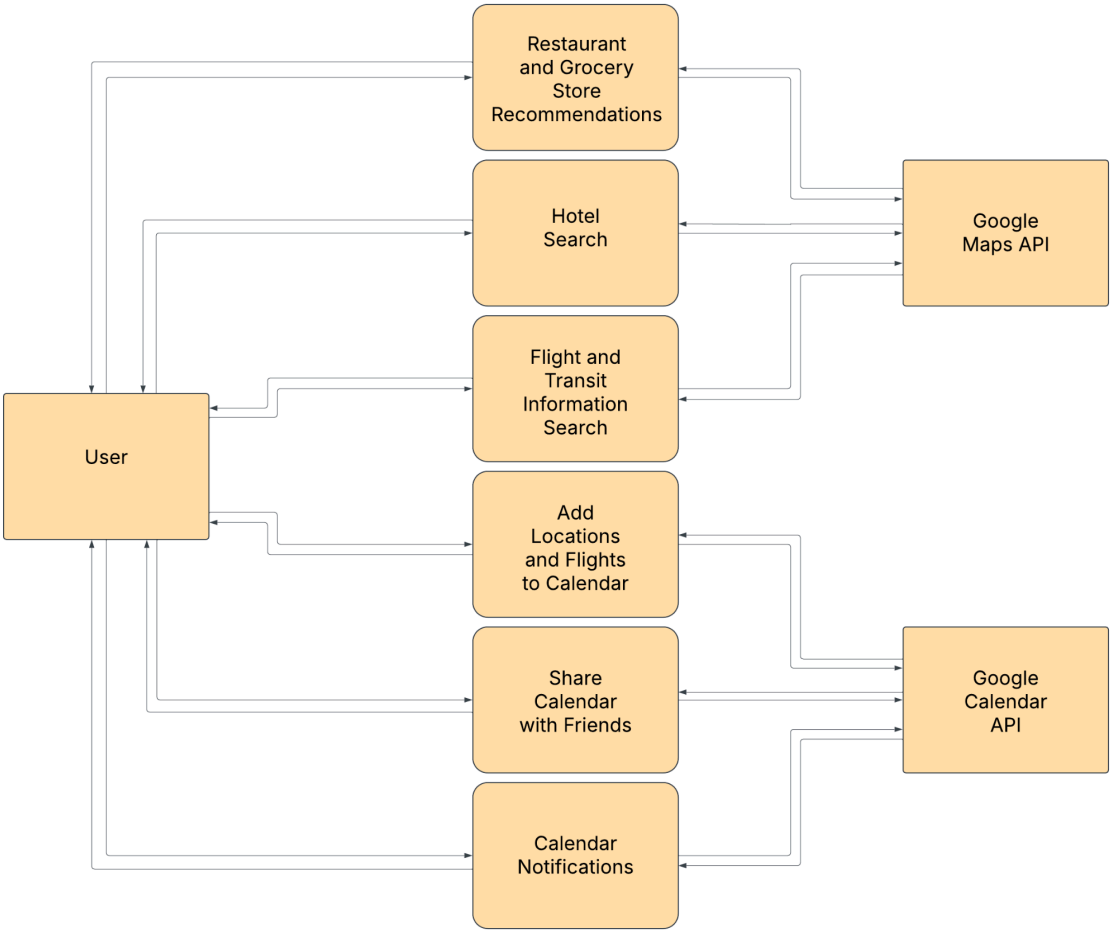
https://www.figma.com/design/HxSbE8efqiUVN3gEGQrFxp/Jack-of-All-Travel?m=auto&t=451dNF4Ga6J6Fktf-1



Data Flow Diagrams

Level 0:

Level 1:

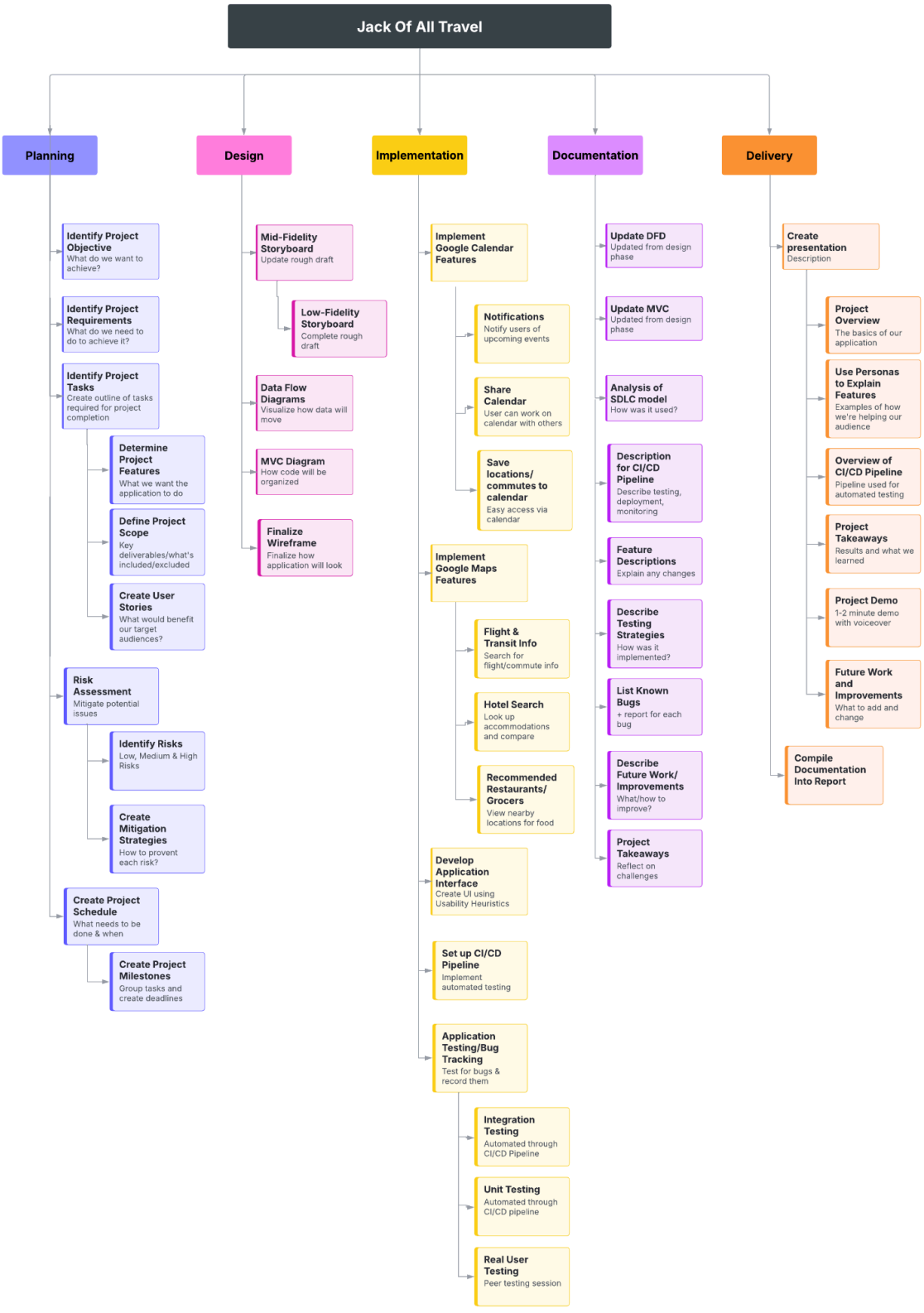


SDLC Model Selection

Agile (Kanban) and Prototype Model

We have chosen a combination of the agile, specifically kanban, and prototype model that is centered on the agile model but takes some aspects of the prototype model. Kanban was chosen as the group is familiar with the model from using it for the mini project, and works well for a project with a hard deadline as it emphasizes teamwork and working quickly. Kanban also provides a great visual for progress, making it easy to see tasks that haven't been started, tasks that need to be worked on, and tasks that are finished along with the team member assigned to each task. The prototype model was chosen since it focuses on making an initial prototype, then fixing issues and building upon it after gathering feedback.

Work Breakdown Structure - [Link to Project]



Project Schedule and Milestones

Class Milestone 0

Due February 7
(Jan 6 - Feb 7)

- Complete group contract
- Set up project repo
- Research APIs
- Complete project proposal report

Class Milestone 1

Due March 7
(Feb 7 - Mar 7)

- Complete planning report
- Planning phase should be complete
- Design phase should be complete
- Complete video presentation

Prototype #1 Complete

Complete by March 20
(Mar 7 - Mar 20)

- Implement Google Calendar Features
- Interface should be started and interactive with Google Calendar
- CI/CD pipeline is set up
- Automated tests written and ready to be implemented

Class Milestone 1.5

Due March 21
(Mar 7 - Mar 21)

- Google Calendar features should be nearing completion
- Start Google Maps features
- CI/CD pipeline fully configured for Google Calendar testing
- Begin debugging issues in Prototype #1

Prototype #2 Complete

Complete by April 6
(Mar 21 - Apr 6)

- Implementation phase complete, both APIs implemented
- Interface is polished and interactive with both APIs
- Tests have been run and bugs recorded
- Debug & record any remaining known bugs
- Documentation & presentation have been started

Documentation Complete

Complete by April 7
(Mar 21 - Apr 7)

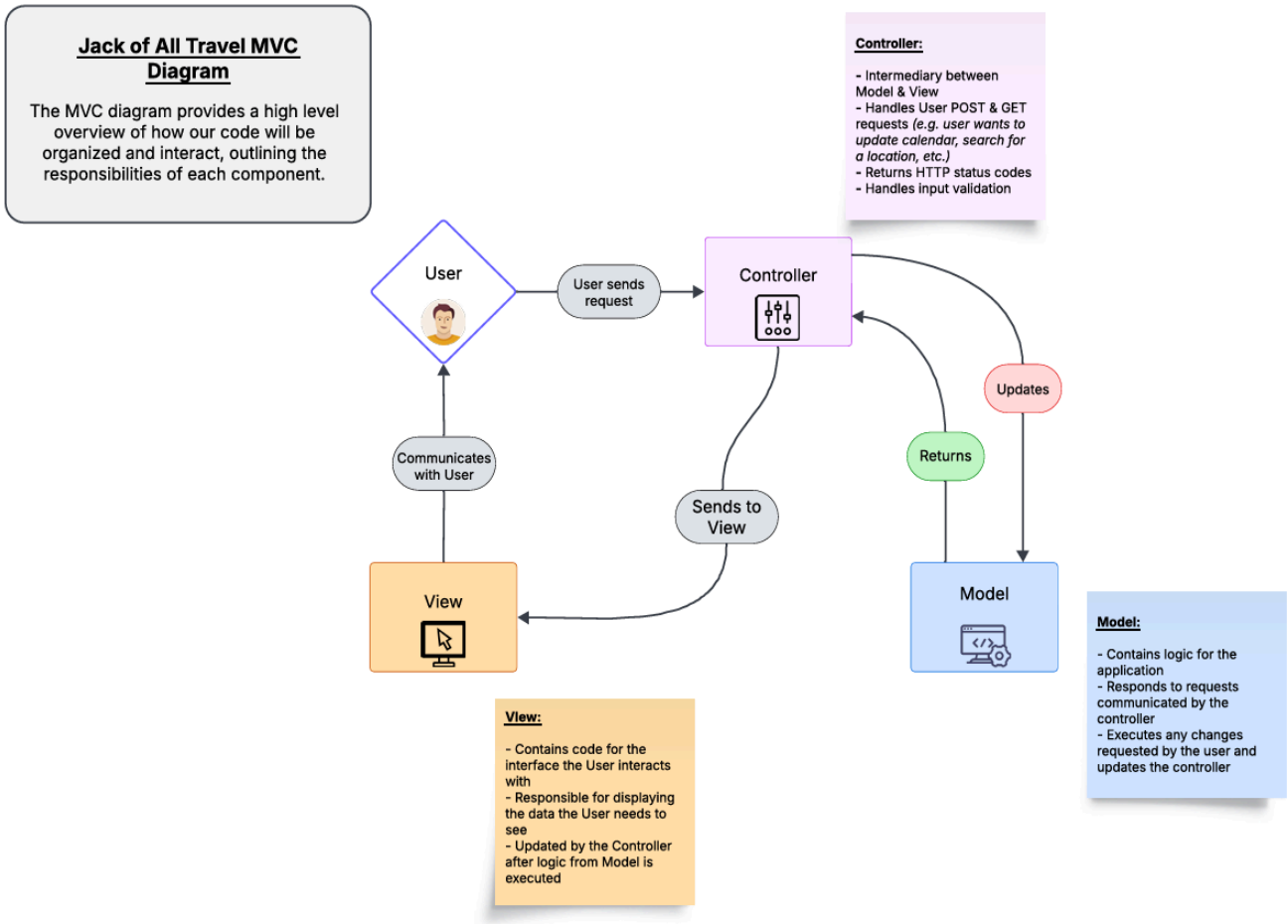
- Documentation phase complete (all project info has been written)
- All known bugs recorded and reports written
- Project presentation is nearing completion/complete

Class Milestone 2

Due April 8
(Mar 21 - Apr 8)

- Documentation has been compiled into Project Final Report
- Project demo/presentation complete
- Source code is organized & clean
- No remaining errors/warnings in console
- Project README complete

MVC Model



Risk Assessment

Low

- Risk: Difficulties gathering and sorting user feedback
    - Mitigation: Create a survey that separates the sections we are gathering feedback on, ask users to fill in survey when providing feedback
- Risk: Difficulty collaborating on different aspects of the project due to coding differences
    - Mitigation: Require comments code that isn't self explanatory.
- Risk: Outdated information staying cached
    - Mitigation: Automatically fetch data again after a specified time, implement a refresh feature, with a cooldown, that allows user to decide when to re-fetch data
- Risk: Inconsistent design between different pages
    - Mitigation: Decide on a base design before hand (e.g. color palette, font), make sure all team members are following the base
- Risk: Limited customization and accessibility
    - Mitigation: Create a light and dark mode, implement a button/setting to swap between the modes. Create specific accessibility colour themes if time permits.

Medium

- Risk: Exceeding API quotas or limits
    - Mitigation: Cache any fetched data that has high potential of reuse, (e.g. recently searched restaurants, saved routes), set limits per user according to documented limits
- Risk: UI inconsistencies between devices
    - Mitigation: Implement responsive UI to ensure application is easy to understand and use between devices
- Risk: Misinterpretation in user location leading to inaccurate recommendations
    - Mitigation: Allow users to enter an address and get recommendations based on the address
- Risk: Test for features are too shallow, missing less obvious bugs
    - Mitigation: Have other team members test features and report bugs, using the application like the personas created in milestone 0
- Risk: Cached data taking up too much storage on user devices
    - Mitigation: Develop and provide users an option to delete cached data from previous travels

High

- Risk: Unable to complete necessary features
    - Mitigation: Establish all necessary features, making them the highest priority, and assigning each feature to a group member. Set strict deadlines for required features.
- Risk: Feature inconsistencies between different browsers and devices

    - Mitigation: Extensively test completed features using docker containers before deploying to ensure application works across all platforms
- Risk: API failures resulting in undefined behaviour and/or confusing errors
    - Mitigation: Have comprehensive error handling and show an error message, provide solutions if possible (e.g. suggesting refresh if element wasn't loaded properly)
- Risk: Intended necessary feature is more complicated to implement than expected
    - Mitigation: downgrade the feature as necessary, but try to keep the core feature intact, brainstorm alternatives implementable features to reduce impact of downgrade
- Risk: No data sync between devices/browsers
    - Mitigation: Allow users to import and export data

Appendix

Contributions

- Jessica Fang
    - Organized project tasks into a Work Breakdown Structure
    - Created milestones and timeline for project schedule
    - Created MVC diagram
- Jessica Liu
    - Completed the SDLC model selection rational
    - Completed the risk assessment section
- Rohin Aulakh
    - Completed the API Selection and Usage as well as the Feature Descriptions written sections.
    - Created a Kanban board inside the github repo and added the tasks inside the WBS as issues to be completed.
- Nicole Stuart
    - Created a Mid-Fidelity Prototype using Figma.
    - Created Data flow Diagrams - Level 0 & Level 1 using Lucidchart.
    - Presented the Mid-Fidelity Prototype overview in the group video presentation.

Changelog

- March 6
    - Google Calendar API feature 3 changed from "Add locations and bus routes" to "Adding locations and flights".
    - Google Maps API feature 3 changed from "Recommended restaurants and grocery stores" to "Recommended dining and services"

Misc.

- Link to Work Breakdown Structure Project
- Link to Project Schedule

- Link to Model View Controller Diagram Project
- Link to Mid-Fidelity Prototype on Figma