Section 1 - Project Description

1.1 Project

Meet In The Middle

1.2 Description

The Meet in the Middle App is a web application which will help a group of users find mutually accessible locations.

Meet In The Middle (MITM) is an app designed for families and friend groups, and is intended to simplify event planning and social coordination. Through the app, users will create profiles, add friends to their profile, and set their preferences. To make planning easier, the app will suggest meeting places based on distance, traffic, and travel preferences. The app will help users and groups plan outings with event itineraries. Additional features of the MITM app include real-time traffic updates, event notifications, and privacy controls to ensure a seamless and enjoyable social planning experience.

1.3 Revision History

Date	Comment	Author
10/8/2023	Version 0.1 – Initial Draft	Craig Walkup

Contents

- Section 1 - Project Description
  - 1.1 Project
  - 1.2 Description
  - 1.3 Revision History
- Section 2 - Overview
  - 2.1 Purpose
  - 2.2 Scope
  - 2.3 Requirements
    - 2.3.1 Estimates
    - 2.3.2 Traceability Matrix
- Section 3 - System Architecture
- Section 4 - Data Dictionary
- Section 5 - Software Domain Design
  - 5.1 Software Application Domain Chart
  - 5.2 Software Application Domain
    - 5.2.1 Web Page
      - 5.2.1.1 Login
      - 5.2.1.2 Dashboard
      - 5.2.1.3 Events
        - 5.2.1.3.1 Itinerary Editor
          - 5.2.1.3.1.1 Itinerary Map
      - 5.2.1.4 Plans

5.2.1.6 Logout

5.2.1.7 Location Search Results

5.2.1.7.1 Location Recommendations Map

5.2.2 Express Server

5.2.2.1 Route 1

5.2.2.2 Route 2

5.2.3 MongoDB Atlas

Section 6 – Data Design

6.1 Persistent/Static Data

6.1.1 Dataset

6.1.2 Static Data

6.1.3 Persisted data

6.2 Transient/Dynamic Data

6.3 External Interface Data

6.4 Transformation of Data

Section 7 - User Interface Design

7.1 User Interface Design Overview

7.2 User Interface Navigation Flow

7.3 Use Cases / User Function Description

Section 8 - Other Interfaces

8.1 Interface X

Section 9 - Extra Design Features / Outstanding Issues

Section 10 – References

Section 11 – Glossary

Section 2 - Overview

2.1 Purpose

This document is intended to be used by the MITM development team and by project maintainers.

2.2 Scope

This document describes every component and integration of the Meet in the Middle app.

The primary benefits of the Meet in the Middle (MITM) app is to facilitate easy social planning and event coordination among family, friends, and groups. We aim to streamline the process of selecting meeting destinations based on user preferences, distance, and traffic, while the goals include enhancing social connectivity and making event planning more efficient and enjoyable.

2.3 Requirements

Your mileage may vary -- we typically break down the requirements to provide a ballpark estimate.

2.3.1 Estimates

#	Description	Weeks Est.
1	UX/UI	3
2	Frontend Development	4
3	Backend Development	4
	TOTAL:	11

2.3.2 Traceability Matrix

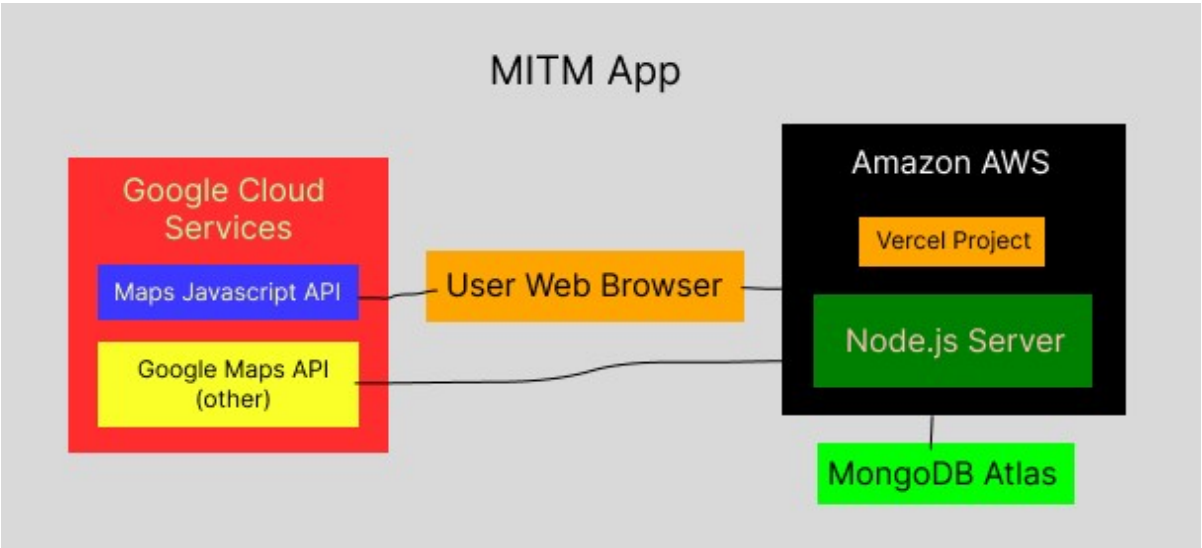
Cross reference this document with your requirements document and link where you satisfy each requirement

SRS Requirement	SDD Module
Req 2	2
Req 3	3.1, 3.2, 3.3

Section 3 - System Architecture

The MITM will use the Express web framework with Amazon AWS hosting and MongoDB Atlas integrations.

Describe/include a figure of the overall system architecture (and where this module fits in)

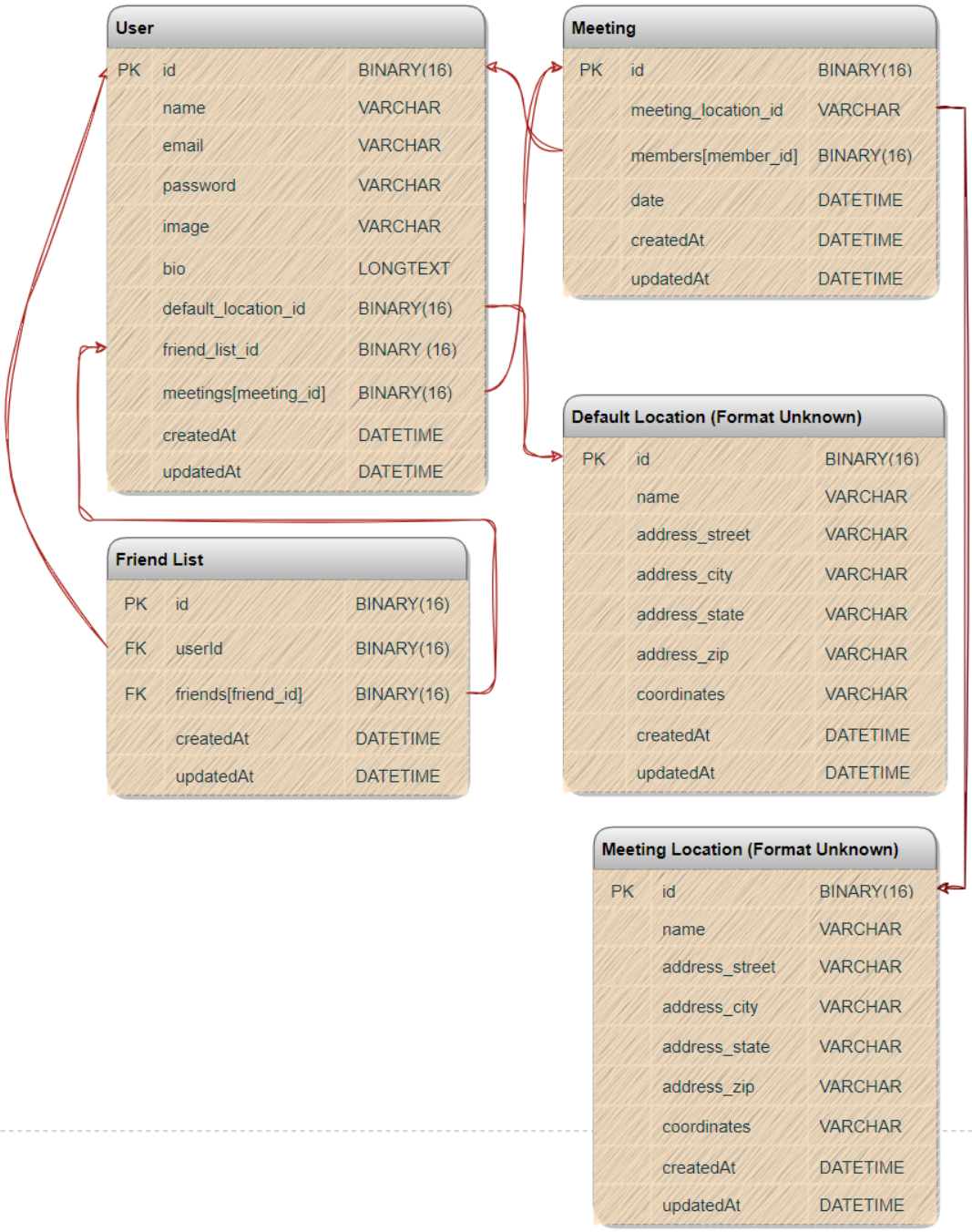


The site will be deployed through Vercel. Express will handle the requests and MongoDB atlas will provide storage services to the server application. The web browser will use Maps Javascript API and the Node.js server (which is deployed via Amazon AWS)

Section 4 - Data Dictionary

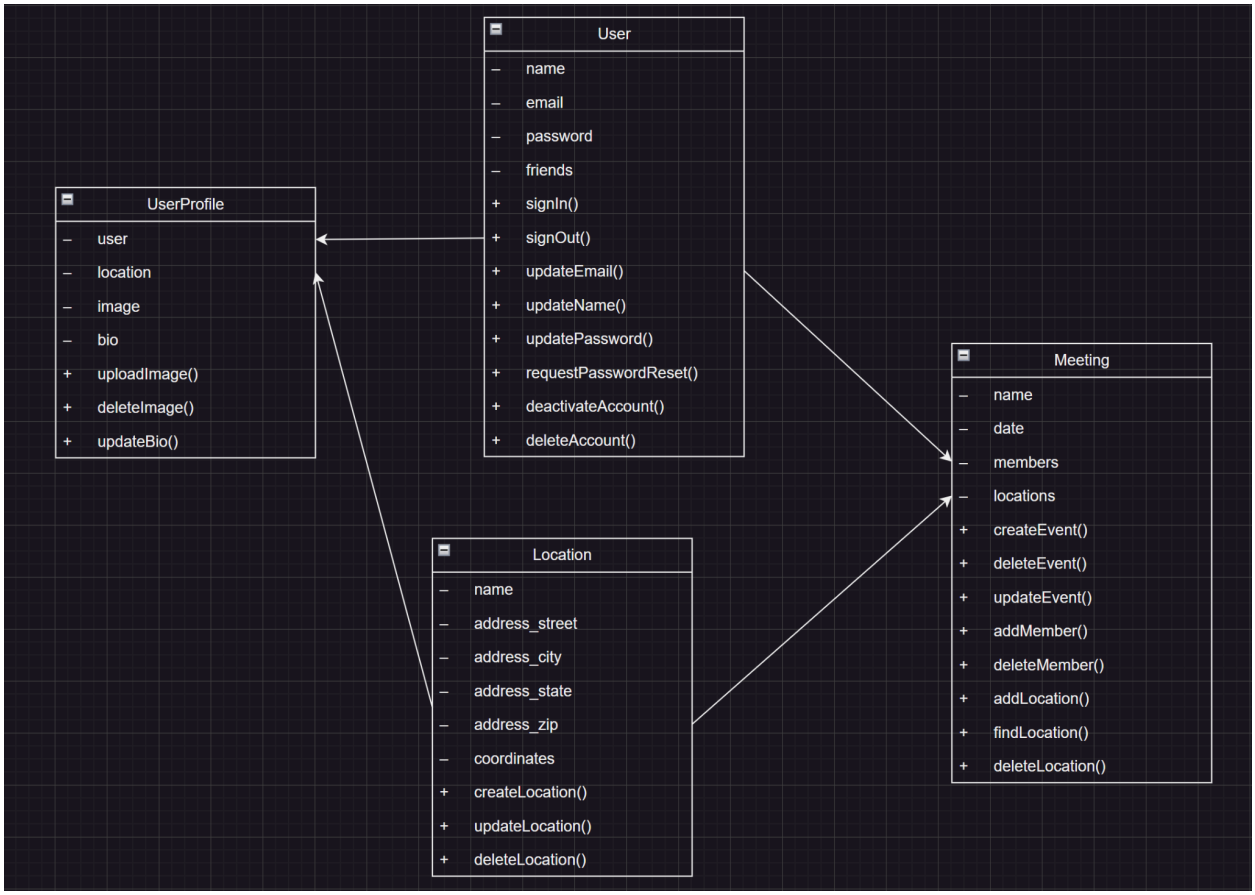
Endpoints and Data Fields:

All tables and data are further explained in Section 6. Each of the following data fields will contain an api endpoint in the backend of the application.



Section 5 - Software Domain Design

5.1 Software Application Domain Chart



5.2 Software Application Domain

A Comprehensive high level description of each domain (package/object wherever it is better to start) within the scope of this module (or within the greater scope of the project if applicable)

The app will prepare location search results, profile pages, and event pages. It will be powered by Amazon AWS services, Next.js, Google Maps APIs, and MongoDB Atlas.

The MITM app is composed of:

- routes; the API
- page templates, static pages, and other data

Navigation Bar

The MITM website may feature a navigation bar.

User Account Services

User authentication protects user data.

Login and Registration

The server will have a static registration form and login form. The login form shall include username and password fields. The registration form shall include a location field in addition to a username and password field.
Upon receipt of a registration form:

1. If the form is successfully processed the information in the form is posted to the backend at which point a User is created.
2. If there was an error in processing the user we are presented with an error and asked to fill in the form again with accurate info.

When an unauthorized user tries to access MITM functions, they will be redirected to the login form. The login page will feature a login field, password field, submit button, and “create account” button. Upon submission of the form, if the credentials are valid, the user email and password is sent to the backend and a JWT is returned. On subsequent calls to the backend, the JWT is used as authentication.

The backend is built using a REST API. Therefore, no logout button is needed on the frontend. Logout will occur when the cached JWT expires.

Alternatively, if the user signs up using their Google Email, then the backend will connect to Google API and acquire user email and provide authentication using OAuth2.

Profiles

When the MITM app receives a registration form, a profile will be created for the user. The MITM app will feature an “edit profile” form. The form is filled in with user information upon retrieval (see x.x.x.x, “Login and Registration”). The user will submit the form to update or edit their profile information.

Friends

The MITM app will serve a static “Search for Friends” form. The form fields will help the user narrow their search. The MITM app will search for users which match the keywords in the form, and serve the dynamic “Send Friend Request” form in response. The friend search feature must not use private user profile information.

The dynamic “Send Friend Request” form will be sent in response to a “Search for Friends” form. It will contain a list of friends. The user will select one or multiple friends. Upon the “Send Friend Request” form, friend requests will be added to the selected users’ profiles with a hook back to the active user.

The MITM app will serve a dynamic “View Friend Requests” form. It will contain a list of users which have sent a friend request to the active user. The active user will designate requests as accepted or rejected. Each request contains a user id. When a request is accepted, the encapsulated user id is added to the friend’s list of the active user. When the MITM receives the “View Friend Requests” form, all accepted and rejected requests are removed from the user’s file.

Events

Events drive the meetup process. They connect users and are used to perform searches. Meet in the Middle events will help creators plan events and meetups. Each event will have an event coordinator, attendees, and an event itinerary.

New Events

Creators will be able to fill out a new event form to create an event page. The event form will include a selection of friends. The server will add the event information to the database and send the event invitations to the recipients.

Event Retrieval

Attendees will be able to retrieve an itinerary for the event. Each item in the itinerary represents a group activity and will provide a time, location, and description. The creator will be able to add and remove events from the itinerary.

Activity Retrieval

The user will retrieve additional details about an activity from the itinerary list.

Itinerary Map

The map will visualize the event travel paths for the attendee.

The event page will provide both the location of a user and the friend that the user needs to meet with.

Event Management

The creator will add items to the event's itinerary, send additional event invitations, and perform a search from the event's management page. Upon submission of the "add an activity" form, a new activity will be added to the event.

Group Search

The search feature will be accessible from the event management page, and will help the creator find the best locations and times for the event's attendees. The creator will be able to use the search to select locations and times for new activities.

The Google Maps API will be used to find locations and routes, and the Maps Javascript API will be used for rendering.

Google Cloud location services can be used in our application at no cost to the development team. In addition to the Google Cloud trial credit, the Google Cloud free tier offers $200 in credit, monthly. Google Cloud services are billed according to different factors, including region. The "Places API (New)" charges a rate of $20 per one thousand API calls. To find more precise costs, from the "APIs and Services -> Library" tab, navigate to the "Places API" or the "Places API (New)" product details. Both of these pages contain links to detailed pricing schemes.

Dashboard (Profile)

The dashboard will present users with the most important information and links to other features. The user will be able to validate and review their profile information from the dashboard. Users will be able to find the most useful information at a glance, including upcoming events, and access friend and event functions.

MongoDB Atlas

MongoDB Atlas is an online database storage system. The MITM web app will use MongoDB to store all database information. Collections are analogous to a relation, and a document to an entry. The Mongoose package from NodeJs creates the database schema and uses CRUD operations to manage the database. See Section 4 for database structure and Section 6 for database information.

Section 6 – Data Design

Describe the data contained in databases and other shared structures between domains or within the scope of the overall project architecture.

6.1 Persistent/Static Data

6.1.1 Dataset

The dataset of this application is based around a User. After the user establishes their credentials via a JWT they will have access to the following data.

6.1.2 Static Data

This web app uses Mongoose as a NodeJs package. The models created via Mongoose will not change throughout the life cycle of the application. Models will include User Profile Model, Meeting Model, Location Model, and User Relationship Model.

6.1.3 Persisted data

User Profile, location, and recommendations will all change as the app is updated.

🧑User Profile:

Each User will contain a unique identifier that is generated by Mongoose. A unique email will also be associated with the user. A password and some metadata will also be stored within the User field.

🧑User Profile Information:

User profile stores more detailed information about the User including a short bio. Currently “User Profile” and “Profile Information” are two separate fields but this may change according to the needs of the app.

📍Location:

Location of a recommended meeting is stored in the database. Location contains name, address, coordinate info and the unique Google address that is provided by Google API. Location will also store a meeting id that will link to a specific meeting.

🤝Meeting:

The meeting field is linked to a specific location via a meeting id.

6.2 Transient/Dynamic Data

Not all data is persistent. The following data does not need to be stored in a database but will be acquired dynamically

🪙Javascript Web Token

JWT will be cached on the device and used as a credential every time a request is posted to the backend.

📲Messaging

There is the future possibility of building messaging into the application. If this feature is added the data will be cached on the device and not stored in the database.

6.3 External Interface Data

Data is gathered from external API's to provide additional features to the User.

🗺️Google Maps API

This API displays an interactive map for the user to view their location and the location of friends.

🛂Google Auth via Passport

Passport is a NodeJs package that will be used to exchange authentication information between the front end and back end of the application. When a user first signs up for the application their Google email address is acquired from Passport.

6.4 Transformation of Data

Describe any data transformation that happens between design elements.

🧑User Data

User data will be obtained from the authentication service and modified to fit the standard format of the database.

📍Location Data

Data received from Google API is modified into a standard format across the platform.

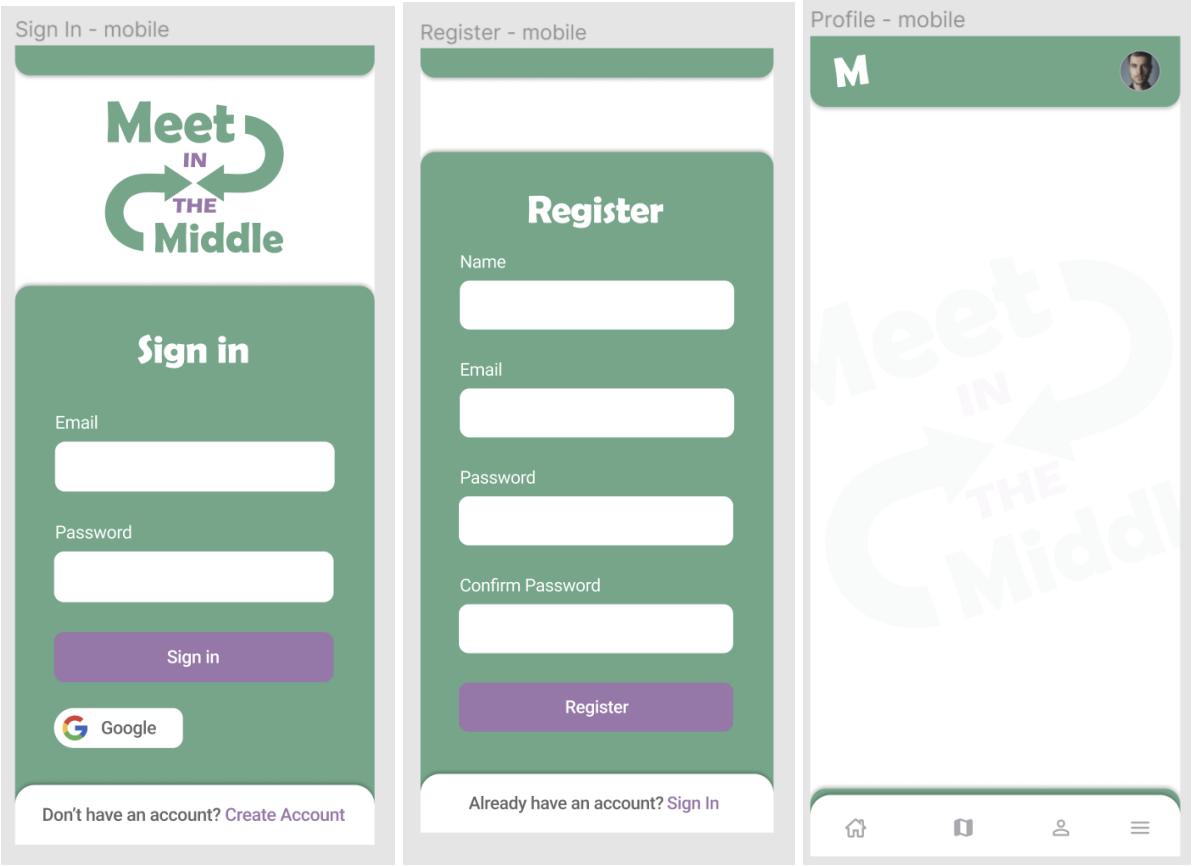
🤔Recommendations

The location algorithm uses both user data and preferences to create recommendations. Recommendations are then stored in a meeting and sent to the front end.

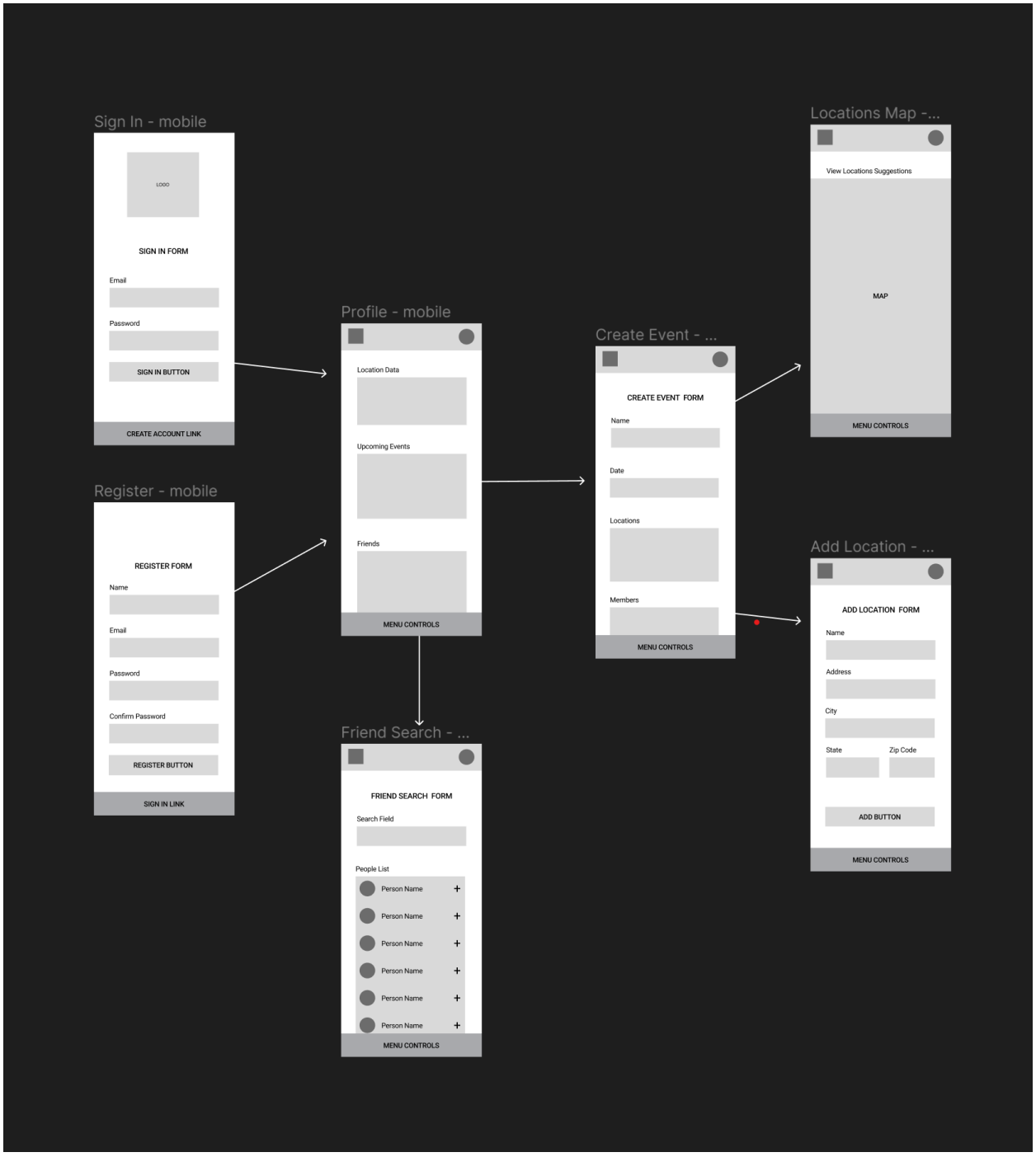
Section 7 - User Interface Design

7.1 User Interface Design Overview

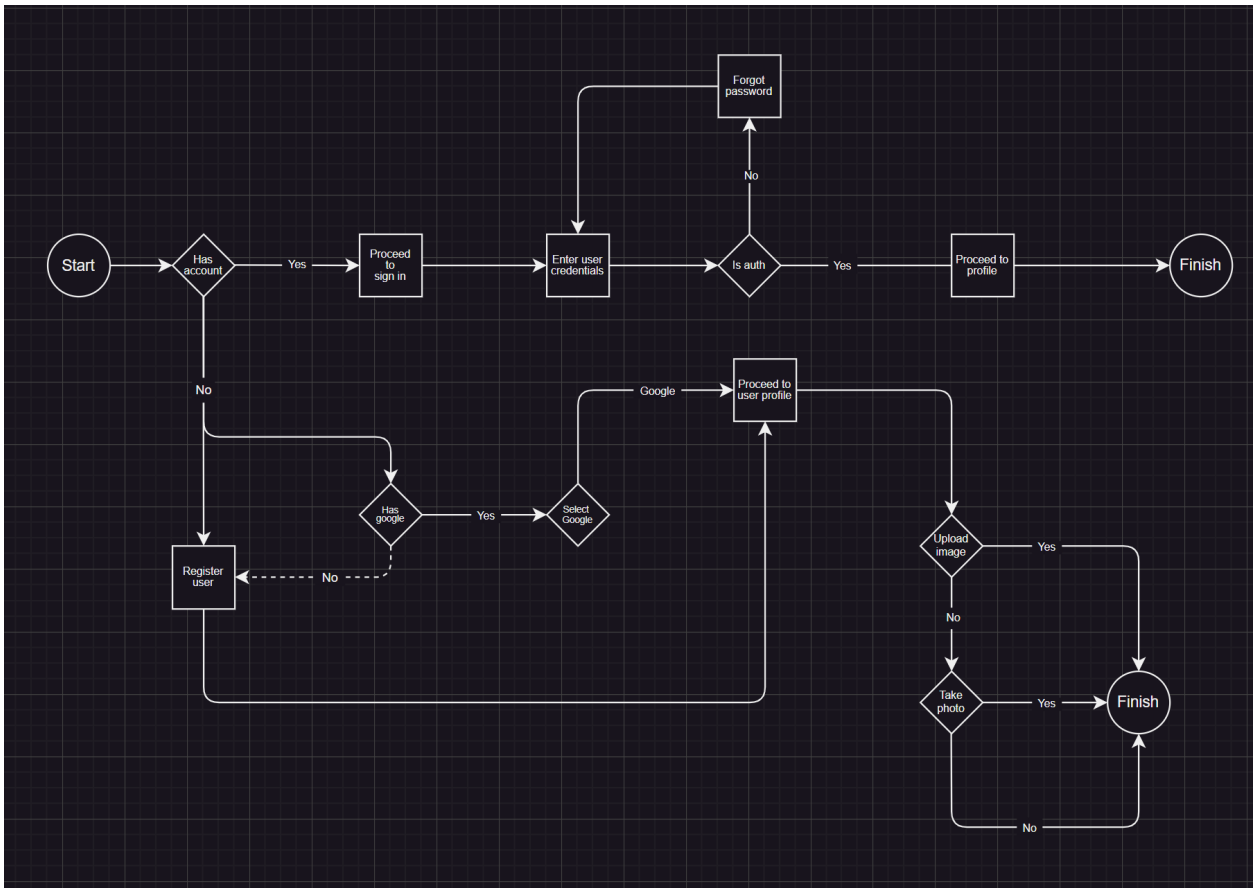
The user interface design consists of a simple color scheme with three primary colors following the 60-30-10 rule to provide a stylistic and clean experience. The main design is centered around the logo with soft accents and a rounded appearance. The main pages will utilize a menu bar at the bottom of the application for ease of access and control. The links at the bottom navigation will consist of a home link, a maps link, a person search link, and a more link that will provide a menu for main navigation. This will include the ability to view the map, handle events, handle friends, and handle their location. There will be a header at the top with the profile image and logo. The profile image will also serve as a secondary navigation menu with the profile controls and logout link.



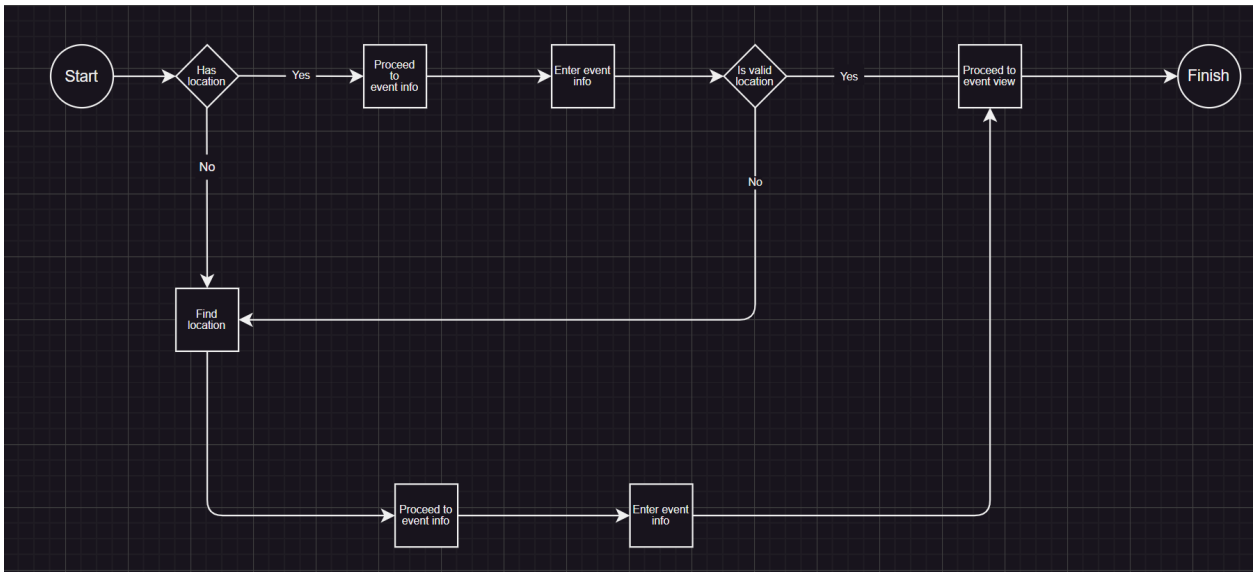
7.2 User Interface Navigation Flow



User authentication and authorization flow chart:

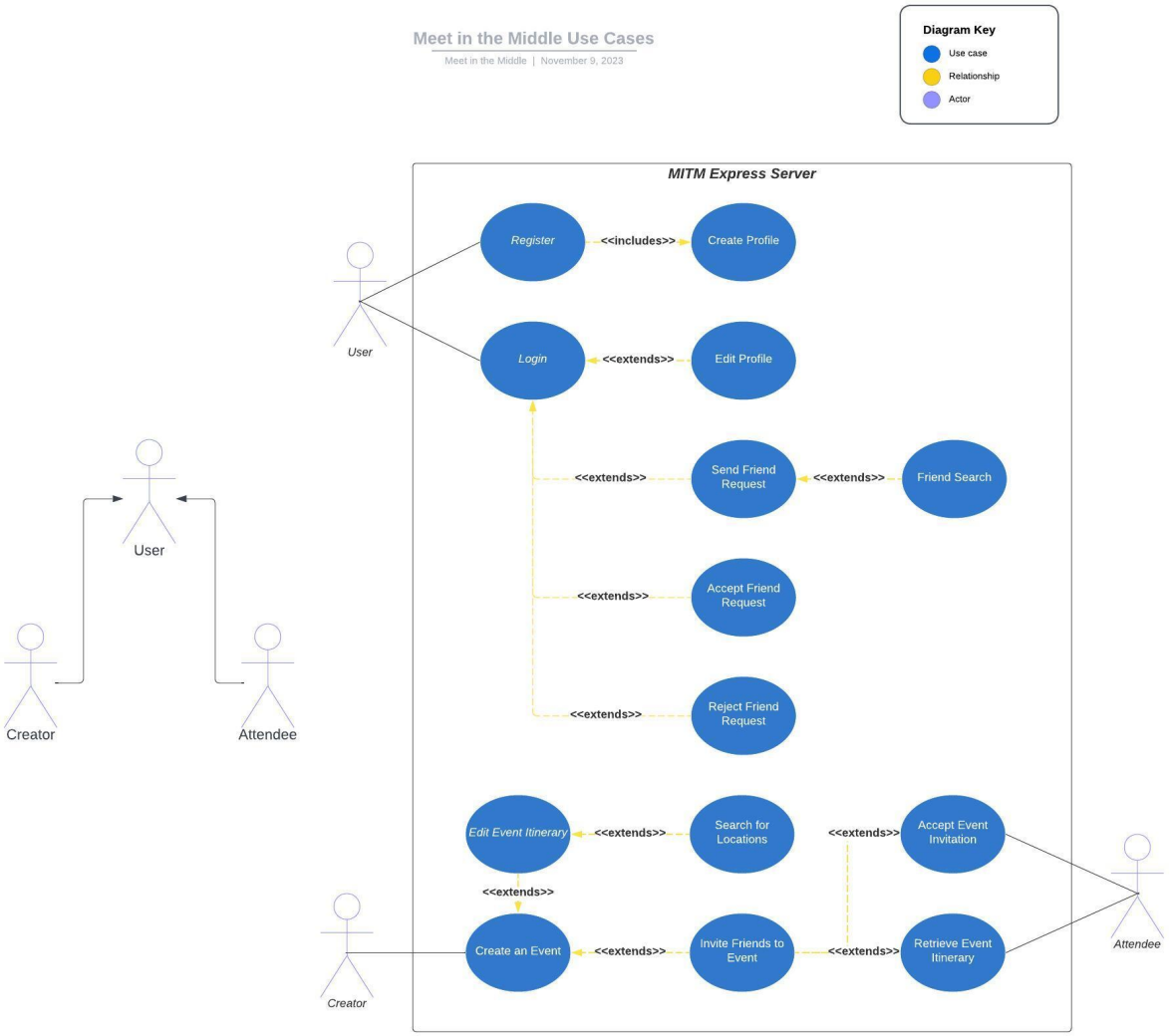


Event creation flow diagram:



7.3 Use Cases / User Function Description

After the initial authorization the user will be taken to their main profile screen to get a quick overview of what is immediately available. The bottom navigation allows the user to go back to the profile overview via the home link, view a map of locations via the maps link, view friends and add friends via the people link, and get access to other navigation opens via the menu link. The main menu will allow users to navigate to the events management where they will be able to add events, modify events, and delete events. User management will allow for users to be searched by name or email and added. Users must be friends to add individuals to event meetings. Forms will be designed and implemented to control the flow of data.



Section 8 - Other Interfaces

Identify any External Interfaces used in the execution of this module, including technology and other pertinent data.

8.1 Interface X

The following is the list of External Interfaces that are known to be necessary for the MITM web app. Additional Interfaces will be added as needed.

🚀Google Cloud Services

Multiple Google cloud Services will be used with the project. This will require both a Google account and a Google API key.

🛂Google Auth via Passport

Authentication is available via Google Account. Passport is used in the NodeJs backend server to communicate and obtain authentication credentials.

🗺️Google Map API

The Google Map API is contacted by both the font and backend. The front end uses the service to acquire and send locations to the backend. The backend also contacts Google Map API to determine meetings.

💿AWS

AWS Amplify allows for full stack deployment of web applications. This is where the application is deployed for use.

Section 9 - Extra Design Features / Outstanding Issues

This is a list of known issues and corrections needed for the development of the MITM web app.

Outstanding Issues:

- A comprehensive description
- Proof of concept
- Completion

Section 10 – References

💻Github MITM Project:
https://github.com/CSCI150-LAB01/Meet_in_the_Middle

Section 11 – Glossary

Glossary of Terms / Acronyms

MITM - Meet In The Middle
The user - An authenticated web browser.