BoilerNav

CS 30700 Design Document

Team 13:

Ivan Ni

Hunter Striegel

Jacob Goodwin

Nathaniel Dang

Roman Marchuk

Index

Purpose

We are designing an exterior and interior navigation system with real-time updates with web and (potentially) hardware support. There is currently no way to navigate within the interior of buildings on the Purdue campus. Software such as Google Maps, Apple Maps, etc. only navigate a user to a specific building, they do not navigate within buildings. This app will be a one-stop shop to navigate anywhere on the Purdue campus. It will scale along with the size of the user base and serve the needs of any person on-campus who requires navigational assistance across the campus. This app will help anyone who needs to navigate to their classes, any visitors to the Purdue campus, and just generally serve to help people. Additionally, this app will also help navigate users to things such as bus stops or bike racks and provide features such as personal routes and schedule based routing.

Functional Requirements

General:

1) As a user, I would like to go through a tutorial for using various parts of the app

Outdoor Navigation UI:

2) As a user, I would like to view an outdoor map of Purdue's campus
3) As a user, I would like to pan, zoom, and scroll around in the outdoor map.
4) As a user, I would like to select Purdue's buildings to view their indoor floor plans.
5) As a user, I would like to view information on bike routes.
6) As a user, I would like to view information on bike racks.
7) As a user, I would like to view information on parking lots.

8) As a user, I would like to view information on bus routes and stops.
9) As a user, I would like to select a building to travel to on Purdue's campus.
10) As a user, I would like to search for a building to travel to on Purdue's campus.
11) As a user, I would like to have route durations to the nearest minute.
12) As a user, I would like to be rerouted if I stray off the path.
13) As a user, I would like to receive real-time GPS updates.

Indoor Navigation UI:

14) As a user, I would like to select which floor I am on for indoor navigation.
15) As a user, I would like to select start/destination routes on the current floor for indoor navigation.
16) As a user, I would like to search for start/destination rooms on the current floor for indoor navigation.
17) As a user, I would like to see the duration of routes for indoor navigation.
18) As a user, I would like to have multiple routes for indoor navigation if routes are of comparable time.
19) As a user, I would like to be rerouted if I stray off the path for indoor navigation.

User Accounts and Features (Outdoor & Indoor UI):

20) As a user, I would like the option to create an account.
21) As a user, I would like the option to login to my account.
22) As a logged in user, I would like the option to update my account information (username, password, etc.).
23) As a logged in user, I would like the option to delete my account and associated data.
24) As a logged in user, I would like to create personal routes for outdoor navigation.

25) As a logged in user, I would like to create personal routes for indoor navigation.

26) As a logged in user, I would like to save frequently used destinations.

27) As a logged in user, I would like to select from my favorite locations and be routed from my current location to the selected location.

28) As a logged in user, I would like to add personal route times to outdoor routes.

29) As a logged in user, I would like the option to make my personal routes private.

30) As a logged in user, I would like to share my personal routes with other users.

31) As a logged in user, I would like to submit a schedule and receive outdoor routing based on the schedule.

32) As a logged in user, I would like to send rooms and locations in buildings to developers/administrators to help build indoor maps.

33) As a logged in user, I would like to be able to flag if a part of the indoor and outdoor maps are inaccurate/vandalized.

34) As a logged in user, I would like to submit feedback (delays, accidents, construction. etc) on the given routes.

35) As a logged in user, I would like to create favorite locations to be routed to.

Security/Feedback/Accuracy/Visualizations/Other:

36) As an elevated user/administrator, I would like to review requests for map updates from logged in users.

37) As an elevated user/administrator, I would like an option to ban certain users from flaggings/sending map info if their information is intentionally inaccurate/spam.

38) As an elevated user/administrator, I would like the option to view and rollback to previous versions of the map.

39) As a user, I would like to view visualizations of the most popular routes on Purdue's campus.
40) As a user, I would like to view visualizations of the most popular rooms in a selected building on Purdue's campus.
41) As a user, I would like to view a real time heat map of route congestion.
42) As a user, I would like to see distances in metric and imperial (choice).
43) As a user, I would like to book rooms through Purdue's websites.
44) As a user, I would like the route info to be sent to my email once I book a room.
45) As a user, I would like to be routed through buildings when the weather is bad.

Hardware (if time allows):

46) As a user, I would like to get my real-time altitude in order to detect what floor I am on indoors.
47) As a user, I would like to get the direction I am facing using a gyroscope.
48) As a user, I would like to receive a more refined real-time position using a GPS and WIFI positioning for indoor and outdoor navigation

Non-Functional Requirements

Usability:

- Application should be available on web browsers
- Application should have intuitive ui
- Application should be easily usable as an outdoor and indoor navigation tool
- Application should provide real time outdoor location data
- Application should provide real time interior location data (if time allows)
- Application should provide ~10 or less meter location accuracy (if time allows)

Security:

- Application should be secure and be protected against map vandalism.

Performance:

- Application should provide duration of routes to the nearest minute.
- Application should support requests from multiple users (1000+) and have reasonable response time (< 5 ms)

Scalability:

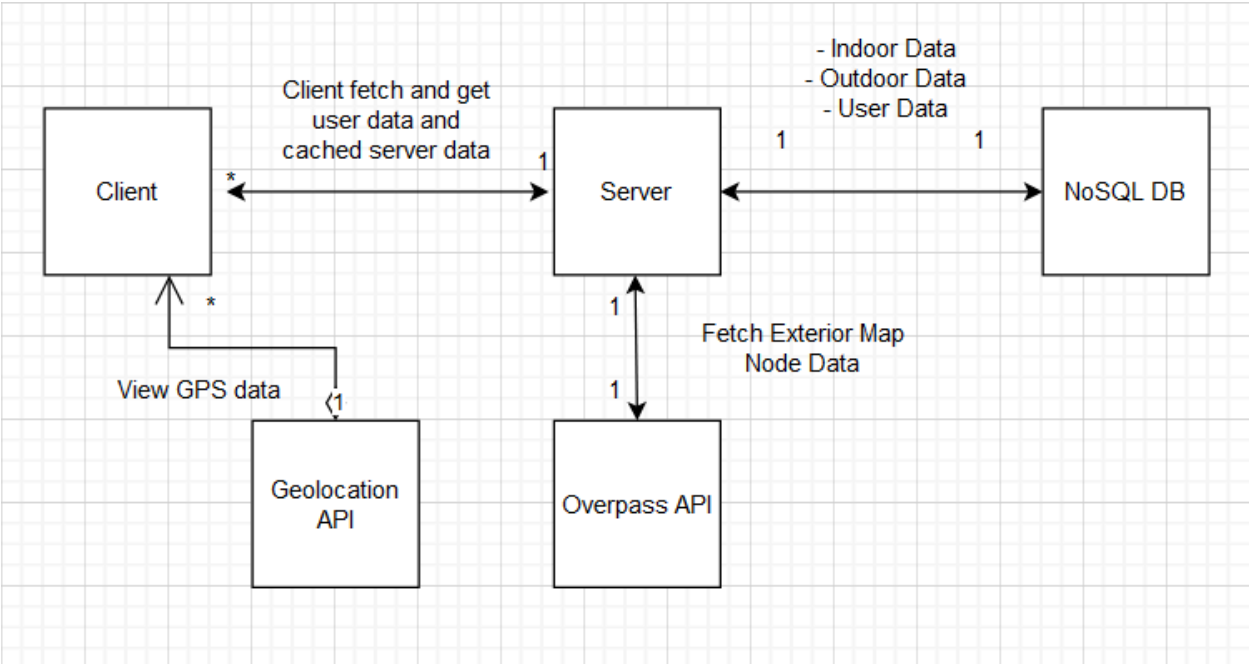
- Application should allow users to mark locations and rooms in buildings, which will help scale the app.

Deployment:

- Application will be hosted on Heroku, Render, or another service and should be available to users 24/7.

Design Outline

High Level Overview:



We decided to go with a client-server-database model because the app requires access to part of a large amount of data. The server can send only what the client requests so the client doesn't need to store a large amount of unnecessary data. Also, clients need to be able to make suggested edits to the data to update the database for other users, even if the original client goes offline and is unable to share suggested edits to future clients.

Database: MongoDB database that contains user accounts, user emails, passwords, saved routes, saved destinations, and other associated data. Database will also store indoor floor data for buildings and outdoor data.

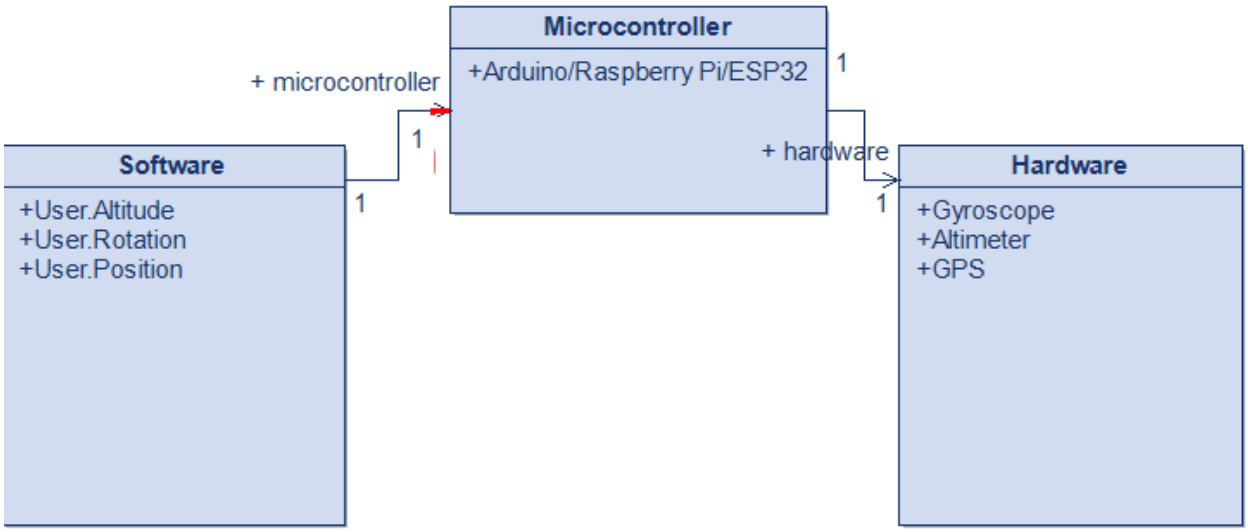
Client: Connects to the server in order to retrieve important information about the app for full functionality. Concretely, this includes navigational data, account information, saved routes, saved destinations, etc.

Server: Processes requests from client and supports create, read, update, and delete operations for our database.

Geolocation API: Provides real time GPS updates for the user's current outdoor location.

Overpass API: The server will fetch map information from Overpass, process it, and then store resulting data into the database.

Hardware Features (if time allows):



Software: Receives the raw hardware data and transforms it into useful data for real-time indoor navigational utility.

Microcontroller: Translates the hardware measurements into software data for real-time updates. This microcontroller is intended to be connected by-wire (depending on the microcontroller brand).

Hardware: Will handle the measurements by a gyroscope and altimeter to compile rotational and elevational data and give those measurements to the microcontroller for further processing.

Design Issues

Functional Issues:

1) What information should be associated with user accounts?
    - Option 1: Email, Password
    - Option 2: Email, Password, Full Name
    - Option 3: Email, Password, Full Name, Major (optional)
    - Option 4: Email, Password, Full Name, Major (optional), Device ID (optional)
    - Option 5: Email, Password, Full Name, Major (optional), Device ID (optional), Last-Location (optional)
    - Option 6: Email, Password, Full Name, Major, Device ID (optional), Last-Location (optional), Gender (optional)

    Choice: Option 4

    Justification: We choose option 4 because we want as much user info as possible for the best user experience and to personalize the application as best as possible, but not be too invasive on privacy where the application ends up storing excessive data that the app does not need. We would also like some user information to be optional so users will only be required to provide minimal data, but can also choose to provide more data to access additional features and have a more complete profile. With the Major data, we can show users where their department is, as well as navigation through some buildings that they may have access to where others don't. The email and password serve as ways for them to log in to their account.

2) How large and extendable should the indoor maps be for users to access?

   - Option 1: A hardcoded subset of campus building interiors
   - Option 2: An empty campus map of interiors which is automatically scalable through user interactions.

   Choice: Option 2

   Justification: We chose option 2 because it will be easier to scale the application along with user traffic, and save on development time as the interiors are automatically built.

Non-Functional Issues:

3) What platform to develop for?

   - Option 1: Web
   - Option 2: Desktop
   - Option 3: Mobile

   Choice: Web

   Justification: We decided to go for the web app because it is more flexible and easier to deploy on different devices. We are also not familiar with technologies required to develop mobile and desktop applications.

4) What framework to use to develop the frontend user interface?

   - Option 1: Angular
   - Option 2: React
   - Option 3: React Native

Choice: Option 2

Justification: We decided on using React to develop our front end user interface since we decided on developing a web app and React is the JavaScript framework that our group is the most familiar with.

6) Fat Client vs Thin Client

- Option 1: Fat Client
- Option 2: Thin Client

Choice: Thin Client

Justification: Because our clients do not need the fastest response time, we will implement thin clients, allowing much more computationally intensive algorithms for the navigation.

7) What database should we use to store our data?

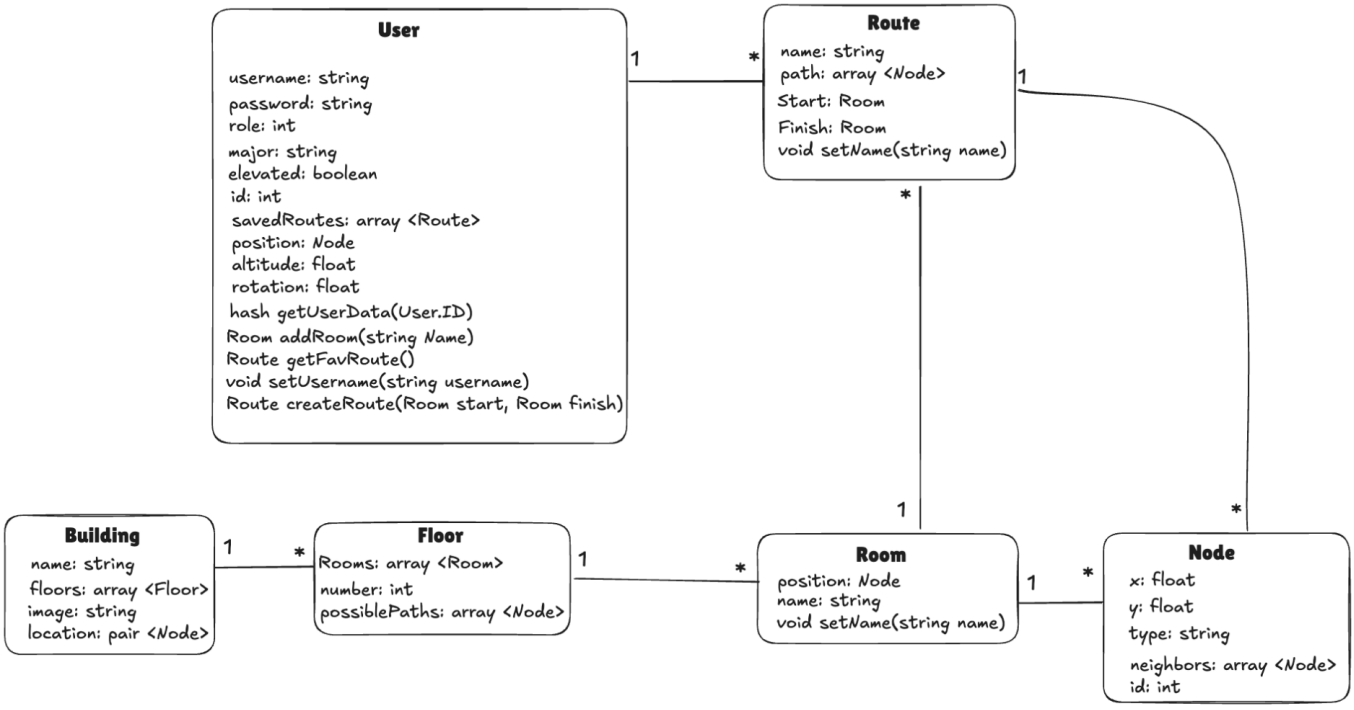
- Option 1: SQL Database
- Option 2: NoSQL Database

Choice: Option 2

Justification: We decided on using a NoSQL database, MongoDB, because of its ease of use and flexible storage options. Our team also has more experience working with MongoDB than SQL databases.

Design Details

Class Diagram:



User: The user is the central class used to store important client-facing data in order to properly interact with the application. The user contains “savedRoutes” which is a collection of saved routes that the user can reference without having to go through the process of selecting buildings.

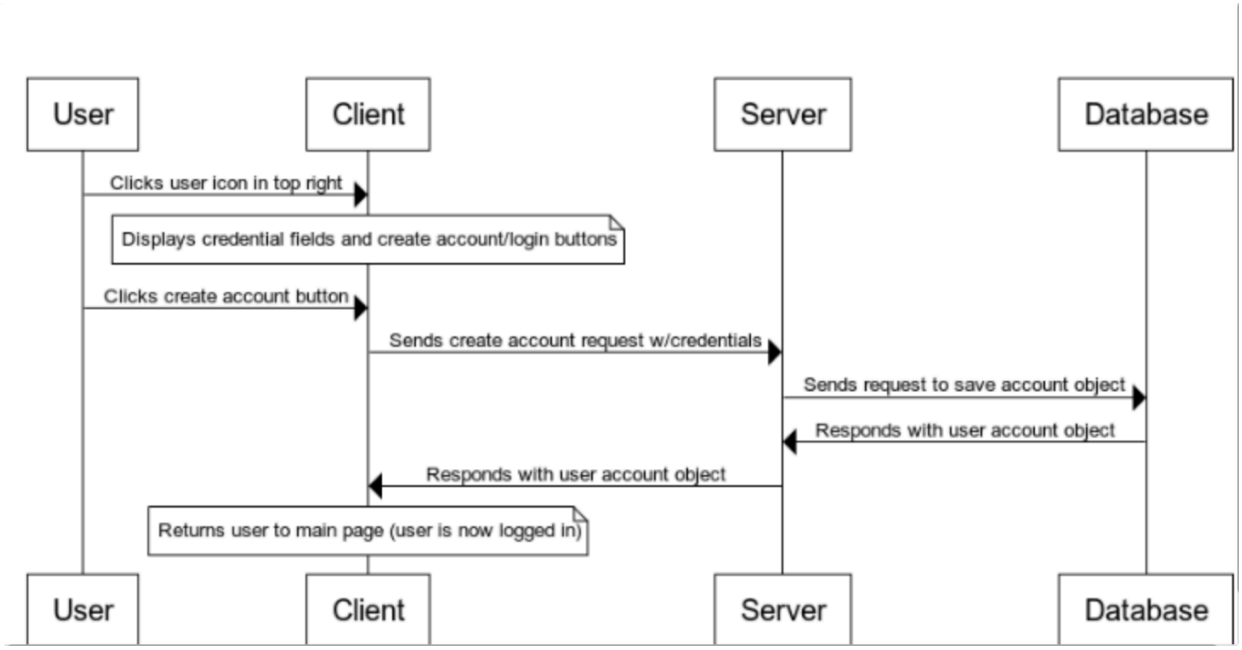
Route: A route is primarily a collection of **Nodes** that store both “start” and “finish” rooms. This class is important for the pathfinding algorithm as reference in order to create the correct path for the user to take.

Room: A room is primarily a node with a name which acts as a reference point for the Route class and is contained in an array of the “**Floor**” class.

Building: A building stores a collection of floors instead of rooms, so that room data is compartmentalized into for easier data manipulation.

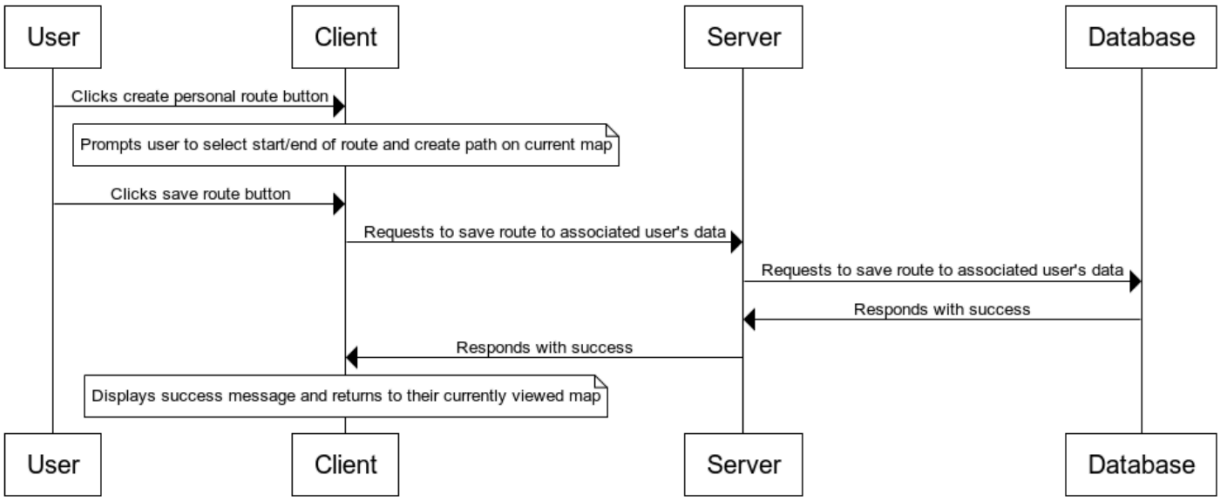
Node: A node is just a point on the building floor map. It stores an x and y component for its place on the map. It also stores the neighbors. These are the nodes that the given node is connected to. This data stores the hallways. Through these nodes, one can create a route to and from each room. There are also types of nodes. There are room nodes, hallway nodes, elevator nodes, and stair nodes. The elevator and stair nodes connect the floors together.

Sequence Diagram for creating account:



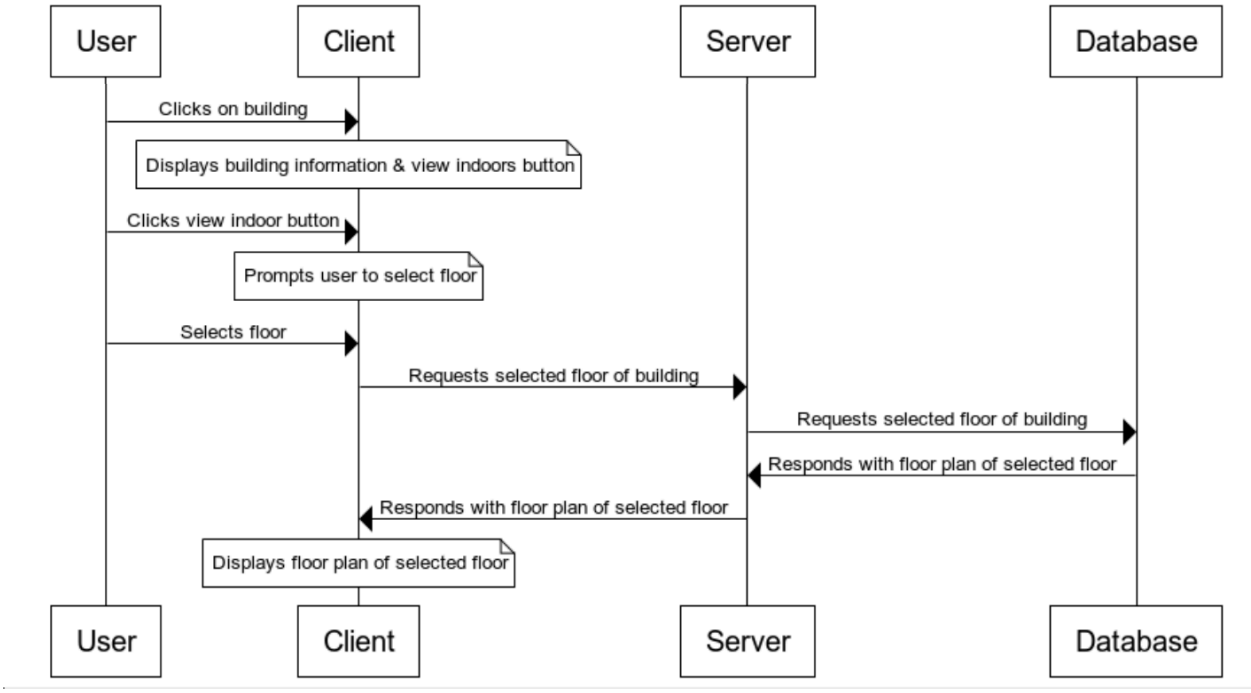
Users can optionally create an account to access additional features of our application such as personal routes, personal route times, schedule based routing, and the ability to help build the application by requesting to add information to our maps. When a user creates an account, they are required to enter an email, password, and full name. They can optionally enter their major and device id, the latter of which will provide additional features. Once the user enters their information and creates an account a request will be sent to the server and the database will store their account information. After the server and database complete and respond to the requests, the user will be returned to the main page and will be able to access the additional feature.

Sequence Diagram for adding personal routes (assumes user is logged in):



Users can create personal routes if they are logged in to their created account. The user must indicate a starting point, destination, and create a path for their personal route. When a user saves the personal route a request will be sent to the server, which will request the database to save the personal route to the user's list of personal routes. The server and database will respond accordingly and the user will receive a message that their request was successful and will be returned to the map they were previously viewing.

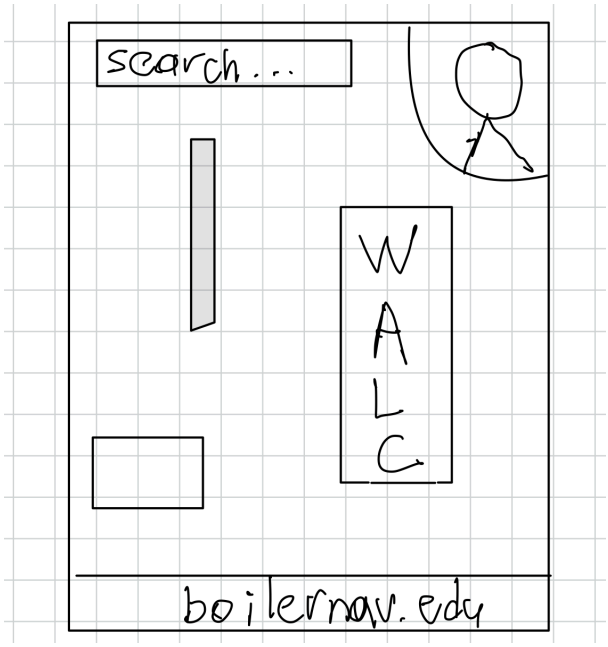
Sequence Diagram for viewing floor of building:



Users can view the indoor floor map of a building by selecting a building and clicking the associated button. They will then be prompted to provide the floor they would like to view. Once the user enters the floor to view, a request will be sent to the server to retrieve the floor map of the selected building from the database. The database and server will respond accordingly and the requested floor will then be displayed to the user.

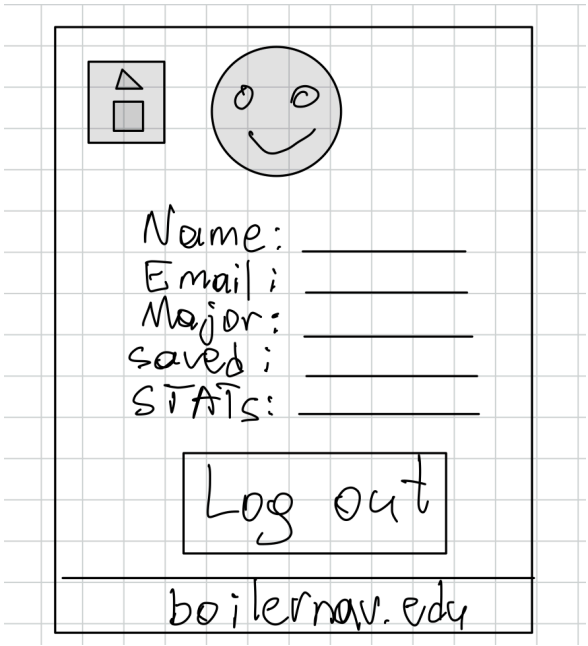
UI Mockups:

Main Page



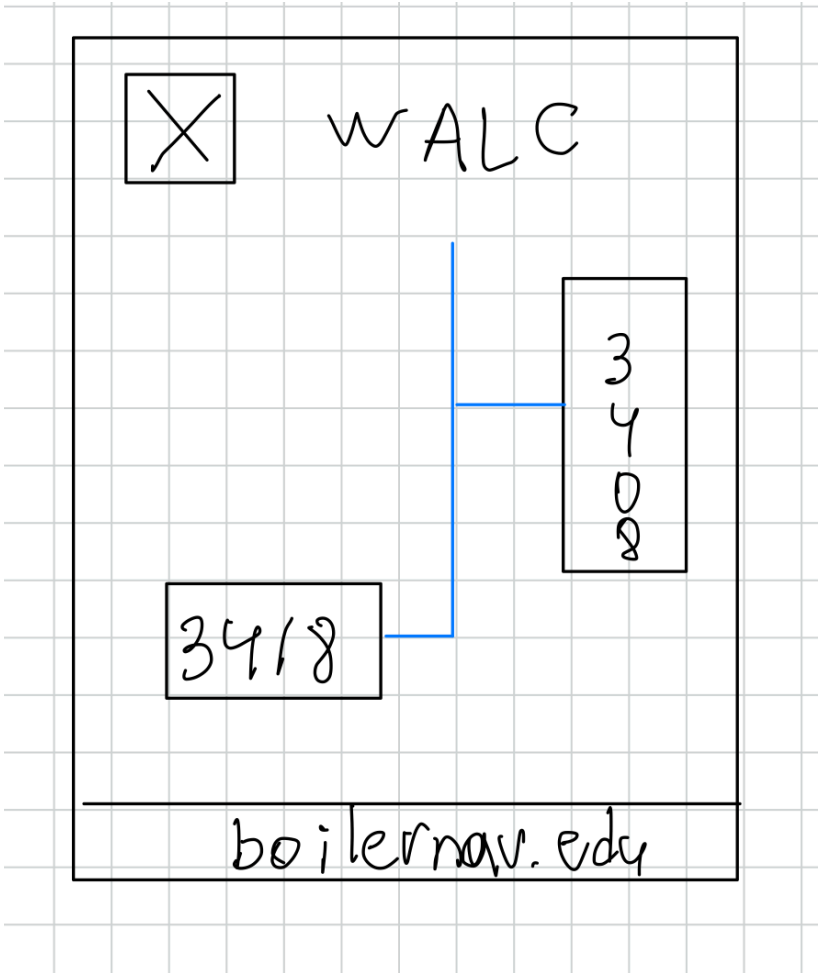
Main page will display the map, search and account buttons as shown above. The buildings with inside navigation available will be marked like WALC in this image.

User Data



If the user clicks the account button, if not logged in, they will be prompted to create an account or log in. Then they will be presented with the screen as above. All of the fields are changeable. Stats and saved items will contain links to the pages where the user will be able to view routes they saved and most visited routes.

Indoor Navigation



When the user clicks on a building from the outside map, they will be shown this screen with an inside layout of the building where he will select the room he is closest to, and then put in the room he is trying to get to. Then he will be presented with a route.