

CS496 Documentation

Brett Bonner and Ryland Mata
Client: Abagail Best

Last Updated: 2/26/2025

Contents

1 Client Information	3
2 Project Description	3
2.1 Overview	3
2.2 Key Features	3
2.3 Why this Project is Interesting	4
2.4 Areas of CS required	4
3 Requirements	5
3.1 User Stories	5
3.2 Non-Functional Requirements	7
4 Iteration Planning	8
5 Design and UI	9
5.1 Architecture	9
5.2 Technology	10
5.3 Data	11
5.4 UI	11
6 Iteration 1	14
6.1 Activities	14
6.2 Retrospective and Reflection	14
6.3 Planning For Next Sprint	14
6.4 Testing	15
7 Iteration 2	15
7.1 Activities	15
7.2 Retrospective and Reflection	15
7.3 Planning For Next Sprint	15
7.4 Testing	15

1 Client Information

1. Client name: Abby Best
2. Client title: Chief Executive Director
3. Client email address: abby@assistedlivingdirect.org
4. Client employer: Medicaid Compasscare
5. How you know the client: IS Professor's daughter

2 Project Description

To create a user-friendly app that will provide an intuitive way for users to search for and access information on assisted living facilities, including their Medicaid Waiver and Senior Assisted Living Subsidy (SALS) certifications.

2.1 Overview

Currently there is no way for an individual to see all the available assisted care facilities near them. This app aims to fix this problem by combining all the care facilities, including those certified for Medicaid and SALS, in Maryland into one app. The client wants the project to revolutionize the search for assisted living facilities by providing centralized, user friendly platform that offers real-time, on all available options. Currently, our client uses an excel sheet to manage all of the data for the assisted living facilities and she has to manually update it. The data for these facilities is currently being manually updated on their website. However, they aim to create their own application version of the website while also implementing real time updates from each facility. We plan on making an app that will automatically pull data from the other assisted living facilities databases and update in real time.

2.2 Key Features

1. User Authentication: Login/Register with email or Apple ID
2. Find Assisted Living Facilities in Maryland and update in real time: Connect to Database, Filter by certifications, Google Maps API integration
3. Facility Availability: Profiles for 29 Assisted Living Facilities, Filter System
4. Facility Profiles: Cost, gender, Wheelchair accessible, Medicaid or SALS, directions
5. Notifications: notify users of changes in availability
6. Admin Panel: add/edit facilities

2.3 Why this Project is Interesting

After learning that there is no centralized platform or application dedicated to finding assisted living facilities that accept Medicaid or Maryland's Senior Assisted Living Subsidy (SALS), My client saw an opportunity to address this gap. Creating such a resource could significantly benefit Maryland residents, helping them find accessible, affordable assisted living options tailored to their needs.

2.4 Areas of CS required

Web development and Database Management would be required for this project.

3 Requirements

3.1 User Stories

S1: Register (2)

- As a user, I want to register an account on the website so that my user data is stored in the website for future use.

S2: Log in / Log out (1)

- As a user, I want to be able to login to my account on the website so that I can access my saved data

S3: Search for Facilities (5)

- As a user I want to be able to see all assisted living facilities in Maryland so that I can search for assisted living facilities

S4: Facility Filters (3)

- As a user, I want to be able to filter living facilities by what they offer so that I can pick the one that fits my needs

S5: Location Filters (3)

- As a user, I want to be able to filter through living facilities based on location so that I can pick the one that is in the area of my desire

S6: Information for Facility (3)

- As a user, I want to be able to see all information for a facility I click on so that I can pick the facility best for me

S7: Notifications (3)

- As a user, I want to receive a notification when there is a change in availability in a facility so that I can see if a facility opened up

S8: Notification Preferences (1)

- As a user, I want to be able to choose how I receive message notifications so that I can receive alerts via email, text, or both

S9: Facility Listings (8)

- As an admin, I want to be able to add/delete facilities and update any information so that I can create new listening and keep information up to date

S10: Upload Image For Facility (1)

- As an admin, I want to be able to upload image of facility so that I can search for assisted living facilities

S11: Update Facility Images (2)

- As an admin, I want to be able to change the image for facilities so that facilities can make updates to their images for any changes

S12: Reset Password (1)

- As a user, I want to be able to reset my password so that If I forgot it, I can log back in

S13: Google Maps API (2)

- As a user, I want to be able to see where a facility is located on a map so that I can see if its close by

S14: Contact Admin (2)

- As a user, I want to be able to create a message to contact an admin so that they can answer my questions

S15: Read User Message (2)

- As an admin, I want to be able to read Users messages so that I can help answer their questions

S16: Delete (2)

- As an admin, I want to be able to delete Users messages so that when I'm done helping the user I can clear the inbox

S17: Saved Facilities (2)

- As a user, I want to be able to save a posting for a facility so that I can keep it stored for future reference

S18: Removed Saved Facility (1)

- As a user, I want to be able to remove facilities that I have previously saved so that I can remove it from my list and not see it when I don't need it

S19: Recently Viewed (2)

- As a user, I want to be able to have a saved list of recently viewed facilities so that I can easily go back to facilities I previously viewed

S20: Pictures (1)

- As a user, I want to be able to see a picture of the facility so that I can see if the facility is nice

S21: Directions (1)

- As a user, I want to be able to have directions to any of the assisted living facilities so that I can see how far the drive is

S22: Contact Information (1)

- As a user, I want to be able to have access to email and phone number of facility so that I can contact the correct people if I have more questions

S23: Comparing Facilities (1)

- As a user, I want to be able to view information of multiple facilities at the same time so that I can compare them between each other

S24: Featured Facilities (1)

- As a user, I want to be able to view the most popular facilities so that I can see what's popular among everyone else

S25: Writing Reviews (2)

- As a user, I want to be able to write a review for a Facility so that I can share my opinion with others for reference

S26: Deleting Reviews (1)

- As a user, I want to be able to delete a review I made for a Facility so that I can remove it in case I no longer want it posted

S27: Reading Reviews (1)

- As a user, I want to be able to view reviews for a Facility so that I can judge how good a facility is based on others experience

S28: Profile Management (5)

- As a user, I want to be able to manage my account so that I can view it, make updated changes, and delete it

S29: Sorting (2)

- As a user, I want to be able to sort my searches based on price, proximity, etc. so that I can compare and contrast my filtered facilities

3.2 Non-Functional Requirements

NFR UI

- The Website shall have the same design and layout as the client's website

NFR Images

- All images shall be uploaded in JPEG format and not PNG

4 Iteration Planning

Sprint	Features
Iteration 1	Search For Facilities (S3) Google Maps API (S13) Information for Facility (S6)
Iteration 2	Facility Listings (S9) Upload Image for Facility (S10) Update Facility Images (S11) Saved Facilities (S17) Register (S1) Log In/Out (S2) Profile Management (S28)
Iteration 3	Notifications (S7) Notification Preferences (S8) Writing Reviews (S25) Deleting Reviews (S26) Reading Reviews (S27)
Iteration 4	Recently Viewed (S19) Pictures (S20) Directions (S21) Contact Information (S22) Comparing Facilities (S23) Featured Facilities (S24) Facility Listings (S9)
Iteration 5	Reset Password (S12) Read User Message (S15) Delete (S16) Removed Saved Facility (S18) Sorting (S29)

Table 1: Iterations Table, Total Story Points: 62

5 Design and UI

5.1 Architecture

We plan on implementing a Web Model-View-Controller (MVC) architecture. The model will be responsible for interacting with the database to perform CRUD operations. Moreover, the model will respond to the controller when accessing the database is necessary because the controller can't access the database. The view is responsible for handling the front-end of the website, rendering the UI and interacting with the user. For this we will be using React as React's components dynamically display data from the back-end and captures user input. The controller will serve as an intermediary between the Model and the View. The controller will communicate with the model to update and retrieve data from the database.

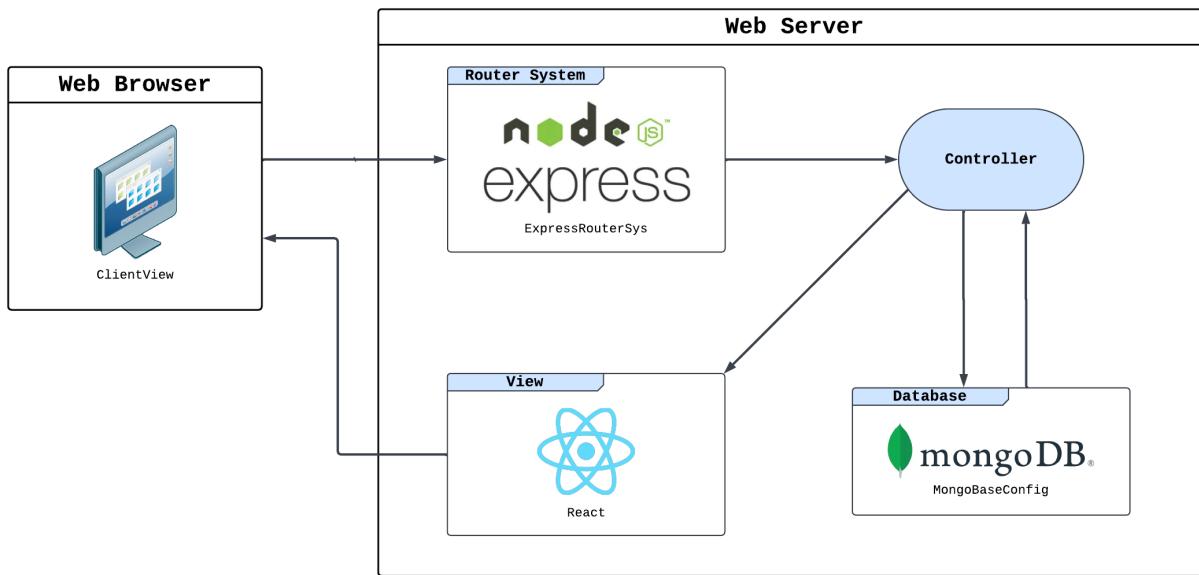


Figure 1: Web MVC Architecture for Medicaid Compasscare Web App

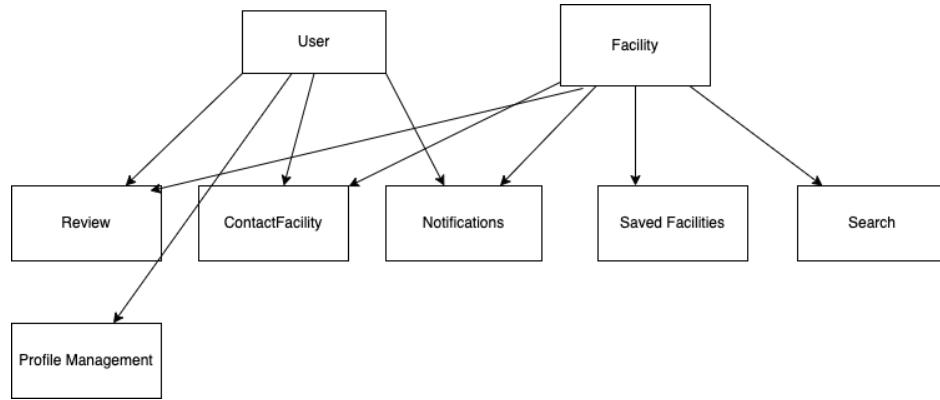


Figure 2: Simplified Class Diagram

We will incorporate a full MERN stack for this project. Thus, the main modules include React for the frontend, Express.js in tandem with Node.js for the backend, and MongoDB will serve as our database.

5.2 Technology

React will be used to build the user interface, leveraging its component-based architecture for reusable and dynamic UI elements. Thus, the primary language for the entire stack, including both the frontend and backend, will be JavaScript. The backend of our project will run on Node.js and Express.js as they will help us in building the API endpoints and managing the server-side routing. To properly compliment our frontend and backend we will use MongoDB as it offers seamless integration with the MERN stack and efficient handling of semi-structured data. We will also be using Jest as our testing framework. Jest allows us to have reliable unit and integration testing for both frontend and backend logic.

As far as libraries go, we will be using an ODM (object data modeling) library for MongoDB that provides schema-based validation and middleware support known as Cors. Axios will also be utilized for making HTTP requests from the frontend to the backend API.

The dataset provided by our client is essentially a large Excel sheet where 29 facilities are connected to it and have access to change it every month. It is extremely unorganized as our client has to manually update their temporary website with this dataset. We will need to clean this before importing it into MongoDB, thus, the data will need to be cleaned and normalized.

5.3 Data

```

1  {
2      "facilityID": "unique_facility_id",
3      "facilityName": "Example Assisted Living Facility",
4      "address": "1234 York Rd, Baltimore, MD",
5      "priceRange": "$2000 - 4000",
6      "capacity" : "10/13",
7      "amenities": ["Handicap Accessible", "Gender Inclusive", "Pet-friendly"],
8      "contactInfo": {
9          "phone": "+1-234-456-7890",
10         "email": "info@example.com"
11     },
12     "images": ["url_to_image1", "url_to_image2"],
13     "lastUpdated": "2025-01-29T15:00:00Z"
14 }
15 {
16     "userID": "unique_user_id",
17     "userName": "Brett",
18     "email": "brett@gmail.com",
19     "password": "123",
20 }
21 [
22     {
23         "reviewID": "unique_review_id",
24         "userID": "unique_user_id",
25         "facilityID": "unique_facility_id",
26         "rating": "8/10",
27         "comment": "Was pretty decent",
28         "datePosted": "12-03-2025",
29     },
30     {
31         "notificationID": "unique_notification_id",
32         "userID": "unique_user_id",
33         "facilityID": "unique_facility_id",
34         "message": "The Best Care Facility has an open room!!",
35         "date": "12-03-2025"
36     }
37 ]

```

Figure 3: Collections and Attributes Data in JSON File

For a NoSQL database like MongoDB, the data structure is typically represented as collections and their associated documents. Thus, each document will be stored in a JSON-like format with key-value pairs, providing flexibility for storing complex and hierarchical data.

5.4 UI

After speaking with our client, they have provided many mockup designs of how they would like their design to follow. These designs were already made prior to us contacting them as they originally intended to create a mobile application. They have since halted this idea in order to work on improving their website. However, they have stated that these designs are still relevant and accurate to how they would like their web application to be designed. These figures act as rough sketches to try and convey their idea of how the main pages should be designed.



Figure 4: Searching For Facilities

They have provided us with sketches for their page where the main listings for assisted living will be posted for users to view. Their prices, names, and icons offer a brief description of what each location offers. As displayed on the right users can filter their search based on handicap accessibility, gender inclusivity, and many more factors important to choosing an assisted living facility.

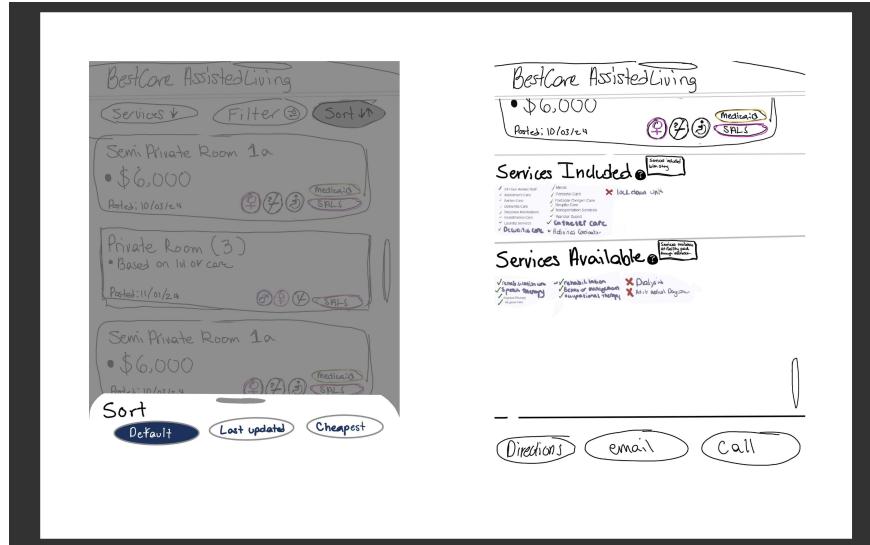


Figure 5: Filters For Facilities

The user can click on each listing in order to learn more about it and contact the facility should they choose to book from there.

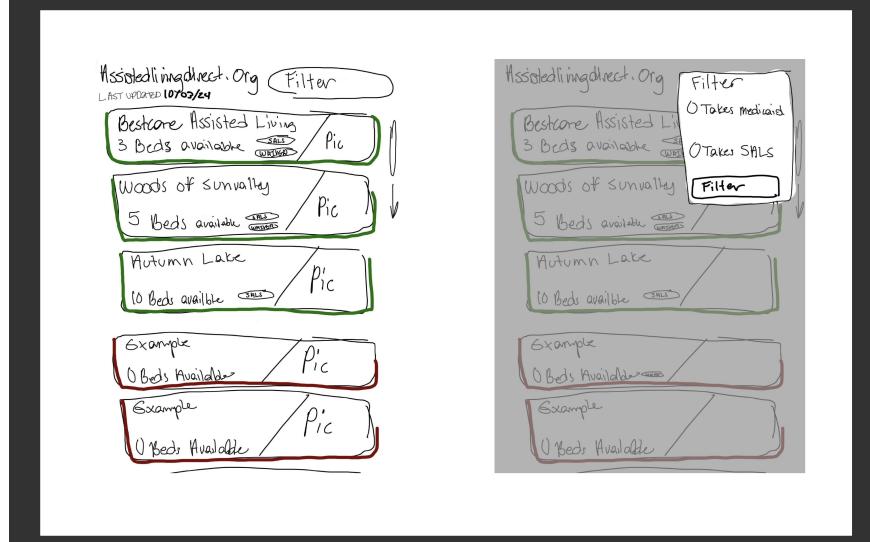


Figure 6: Searching For Facilities + Filter

Our client has also hoped to display a small image for each facility listing so that users can get a brief look into where they are booking exactly. There is also a variant of the filter shown previously which would appear at the top right. It is much more simplistic in nature but still conveys a general idea of how our client would want the web application to be designed.

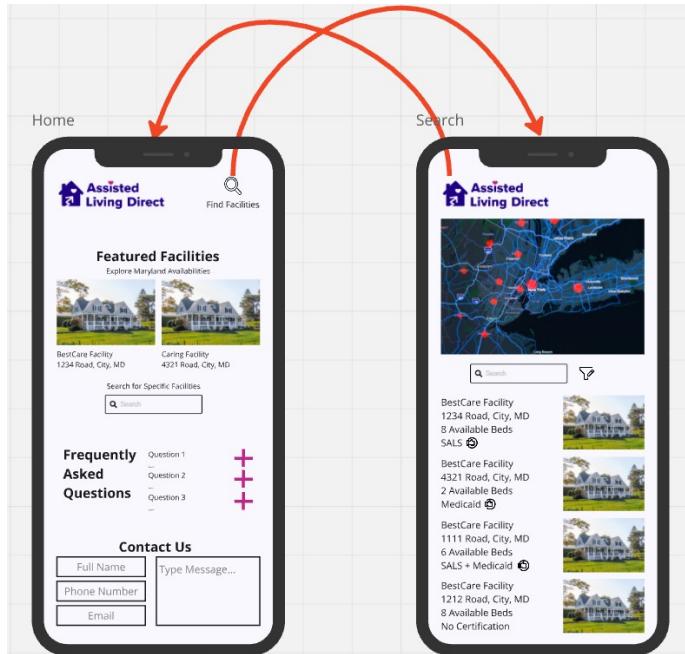


Figure 7: Mobile Website

Along with our client's mockups we have also created our own wireframe for the website based on how it would look as a web application on mobile devices.

6 Iteration 1

6.1 Activities

This sprint established the foundation of our application by setting up the MongoDB database and adding the necessary data to it. Additionally, we developed a script that our client asked for to efficiently clean and process a messy Excel sheet they receive from the government each month. This dataset contains the most up-to-date information on assisted living facilities, including details on whether they accept SALS or Medicaid.

This sprint saw us making a lot of changes in terms of our planning. We originally had planned to do user stories related to registering user accounts, logging in/out, and making sure that each account has the correct information when stored in our database. However, we did not complete any of these intended user stories as we soon realized that the purpose of our web application was of much higher importance. We needed to get the assisted living facilities listed on our website and figure out how to sort through their given data sheets. With the facility listings it would also make it easier to work on the stories revolving around the user accounts as we would already know what attributes to apply for the user class. In order to do this, we also needed to establish the database and establishing our database first and foremost would be essential for future success. If we could setup and connect our database properly, along with handling API endpoints between frontend and backend, we could then continue on with the rest of the sprints focusing solely on the stories themselves.

Instead of working on Register (S1) and Log In/Out (S2) as they are not essential stories, we instead worked on Information for Facility(S6), Google Maps API (S13), Search for Facility (S3), and completed our technical task of Cleaning the Database. We were also slightly held back from our client as it took some time for them to send us all the relevant information for our database.

6.2 Retrospective and Reflection

Overall, Sprint 1 did not go as planned and resulted in a slower start than expected. We underestimated the time required to set up our technologies and database, and there was a learning curve with MongoDB, and neither of us had prior experience with it. However, now that we have successfully received all the necessary data from our client, we are confident that the sprints moving forward with progress more smoothly.

6.3 Planning For Next Sprint

For the next iteration we are going to continue working on the facility listings and make sure that they contain all the correct information. Not only this, but also implement new additions to them such as images for each facility (if they are included) and providing links to each facility to allow for users to contact their desired facility.

If time permits, we will also work on getting the user accounts created and simple logging in/out for them. These stories are not exactly essential for the functionality and purpose of the website, however, if we are to implement features planned in the future then it would be optimal to get these stories taken care of sooner rather than later.

6.4 Testing

We did not get to testing this iteration due to a slow start, however, testing will be prioritized next sprint.

7 Iteration 2

7.1 Activities

This sprint saw us working primarily on the main functionality of the website. We tackled both the search functionality and Google Maps API for the main searching of facilities. Furthermore, we worked on adding an admin panel for proper management of the facilities directly from the website. This is to ensure that the client does not need to manually enter the database and change any values. The UI for the admin panel is straightforward and follows the principle CRUD operations.

This sprint saw us making much less changes to our planning than before. We were able to mainly follow through with our planned stories and did our best to get to each story. Overall, it was very difficult to try and get to each story perfectly but we certainly tried to hit every story we had planned out.

7.2 Retrospective and Reflection

Overall, Sprint 2 felt a bit sluggish and tedious. There were many nuances and problems we were having with each story. Whether it be the Google Maps not displaying correctly or difficulties in uploading/updating images for facility listings in the admin panel. There were many tough obstacles to overcome and this also meant starting testing much later.

7.3 Planning For Next Sprint

For the next iteration we are going to continue working on the facility listings and admin functions. We still need to create user accounts too but that should not be too much of a problem. The hardest part will be user reviews and is something that we will need to focus heavily on if we want to get it done in time.

7.4 Testing

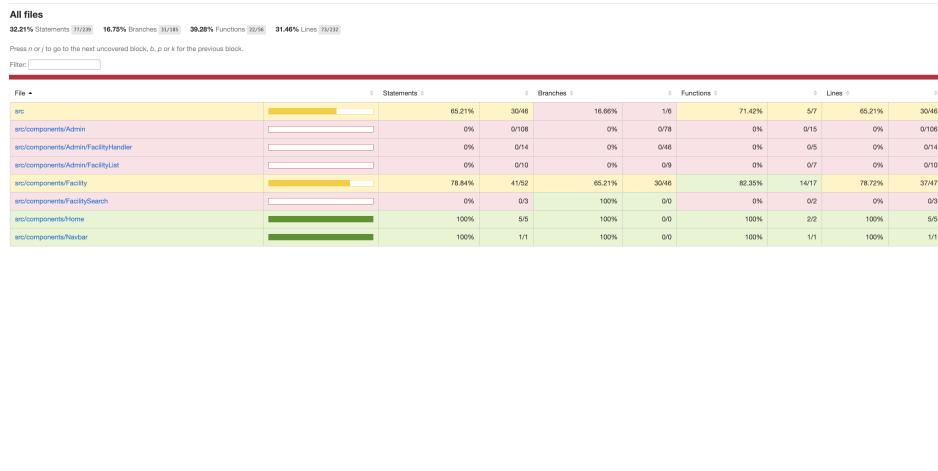


Figure 8: Enter Caption