CNAT

Final Report

Client:
In Motion Care LLC

Advisor:
Daji Qiao

Team Members:
Austin Sehnert
Benjamin Zaley
Brandon Elizondo
Kirkland Keith
Suzanna Gudivada

Team Email:
sdmay20-51@iastate.edu
Team Website:
http://sdmay20-51.sd.ece.iastate.edu

Table of Contents

List of figures

1. Project Introduction:

1.1 Acknowledgement:

We would like to acknowledge the contributions of the following organizations and individuals towards our successes during this senior design project: In Motion Care, Iowa State University, Green Hills Retirement Community, WISER Systems, and Dr. Daji Qiao.

1.2 Problem Statement:

At Green Hills Retirement Community, the healthcare staff have been facing the problem of all resident care plans, the detailed information on individual residents and their needs, have always been received using large paper packets. Updated multiple times a day, these 27 page packets are printed and delivered to each one of the facility's 75 employees with nearly every update. This causes multiple issues where employees find themselves not necessarily having the most updated packet and thus not delivering residents the proper care they need.

1.3 Project Overview:

To remedy the use of paper packets, our project sets out to create an online care plan delivery system. Done in the form of a dashboard application, we will be re-implementing In Motion Care's care plan application in an easier to use and accessible manner. This care plan application will feature all necessary and updated information for a Green Hill resident in a detailed and concise manner without using no more than a single page of paper.

Furthermore, we are utilizing a system of previously setup tracking devices and tags to help track healthcare staff and equipment as they move around the facility. Though currently planned to ensure proper care is being done by the staff, the system will eventually be able to give the staff real-time updates on the residents as they enter their rooms.

Finally, this application will also feature an administrative function to control what user's (the staff) have access to within the application, including the changing of care plans, viewing of the map, and other supervisory details.

As stated before, this project has a planned operational environment of Green Hills Retirement Community in Ames, IA. This facility is completely indoors, however, both residents and healthcare staff may act as hazards due to interactions with the hardware. Throughout the course of this project, we worked closely with our client, In Motion Care, to develop this application to support the growing needs of the Green Hills Retirement Community, its residents, and its staff.

1.4 Requirements:

Our requirements initially were to supply our client with four major deliverables within the web application. As our dashboard application will be used primarily by the healthcare staff and administrative personnel, our focus is to make the application intuitive and user-friendly so that it may improve the staff’s day-to-day tasks.

Our application needs to grant the healthcare staff on duty an active map that is paired with In Motion Care’s current WISER tracking sensors. The real-time map portion of our project will display the location of all tagged personnel and equipment within the tracked area in Green Hills Retirement Community. Green Hills has a policy that rooms must be visited at least once every two hours. The map will alter the color of the rooms to signify when a room needs to be visited, similar to an order tracker in a fast food restaurant. The statistics portion of our dashboard application will display room activity over a selected period of time.

Green Hills has requested that the information be presented in a priority-based manner, in which the rooms with more faculty-resident interactions are presented first. Our playback feature, which will allow a user to view location data from a given date and time, will utilize the stored data from the tracking sensors inside our database. This will display similar to the real-time map, in which the path data is tracked over a given time interval. The dashboard will acquire its data from anchors placed in the facility and tracker tags placed on the asset we will be tracking. The anchors and tags we are dealing with are by a company Wiser Systems. The tags cost $69 and the anchors cost $209. Accuracy of the data collected is directly proportional to the number of anchors. Data management systems are maintained on Google Cloud.

The anchors will be strategically placed in locations that will not disturb the patients or the health care staff’s daily routine. The anchors require power through wall-outlet or ethernet. Implementing a method for tags to be carried with the least chance of damage while maintaining optimal accuracy.

Software Requirements for Front-end:

- ReactJS
- HTML 5
- Google Chrome

Software Requirements for Back-end:

- Spring Boot Server
- MySQL on Google Cloud
- Wiser Systems API

1.4 Intended Users and Uses:

Our intended users are Green Hills healthcare staff and administrative personnel. The administrative and healthcare staff would have the ability to view a real-time map of the facility, displaying the location of all tracked personnel and equipment.The map would be used by the healthcare staff to know when a room needs to be visited. The healthcare staff also has access to update the care plan using the integrated application from In Motion Care.

The administrative staff would have additional privileges to view statistics of healthcare administration over a selected period of time. These statistics will be used to determine where the facilities resources are being used the most among the residents. Also, the administrative staff will have the ability to access the playback feature. The playback feature serves as a way for Green Hills to look back on the location data if an audit is required, or to gain a better understanding of a time period following an event.

1.5 Assumptions and Limitations:

Assumptions:

- The dashboard will not be used in any facility other than Green Hills Retirement Community.
- Users of the dashboard speak english.
- The facility has a consistent supply of power.

Limitations:

- No tag carrier can accurately be tracked if traveled between more than two rooms in less than .2 seconds.
- For power, anchors need to be placed near a power supply when wireless or be tethered to each other via ethernet cable when not.

2. Revised Project Design

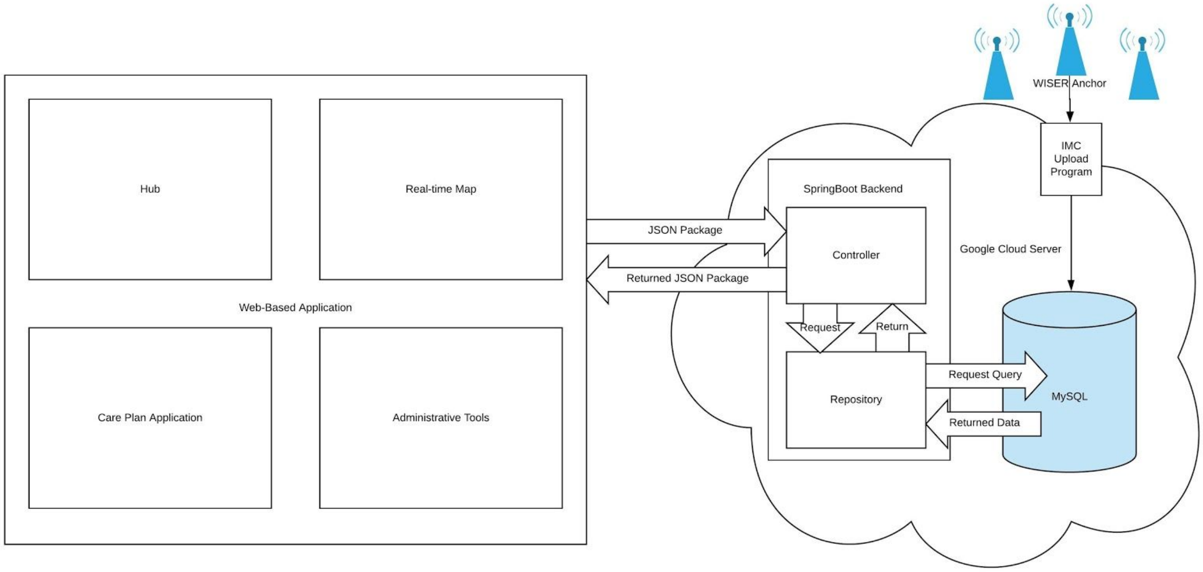


Figure 1. Conceptual Design Diagram depicting communication between different parts of the system.

The entire application is broken down into three main parts. First, the Wiser Systems anchors will pick up locational data from RFID tags worn by healthcare staff and placed on certain equipment in Green Hills Community Center. Each tag has a tag ID that is sent to the database via an upload program made by In Motion Care prior to the start of our project, along with the locational data.

The database is a MySQL database that is connected to a Java SpringBoot backend. Both the database and backend sit on a Google Cloud server. Then information is sent and received between the backend and the frontend through web sockets for the tag locational data and JSON requests for all other data.

The frontend of the application is built with JavaScript React and has four main pages. The first page is the central hub that has hyperlinks to each of the other pages. Then there is the admin page that holds data on the employees, the care plan which holds data on the care of residents, and the map which displays current locations of nurses in the facility.

3. Implementation Details

3.1 Front-end Design

The front-end of our application is built on a JavaScript React framework.

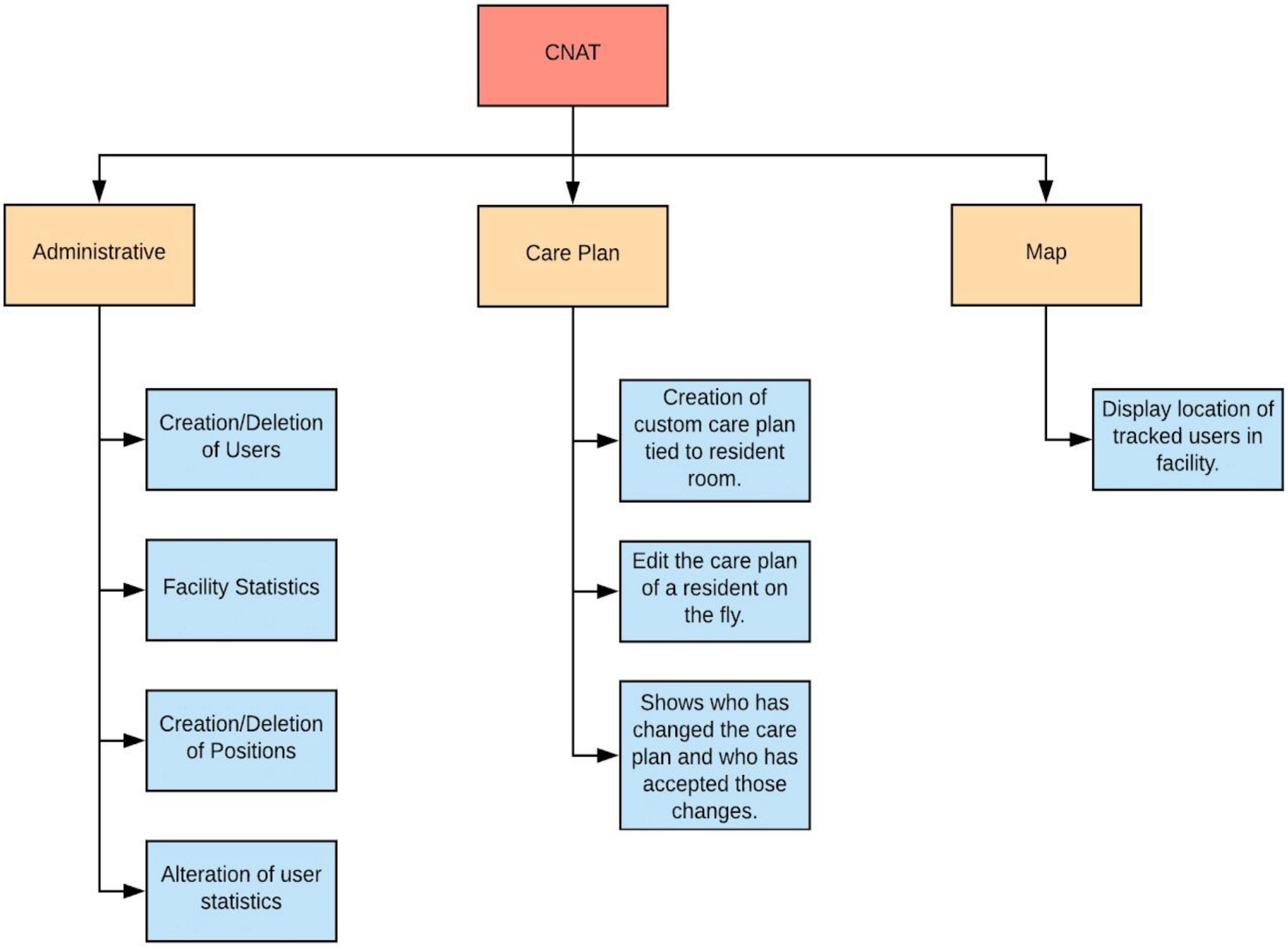
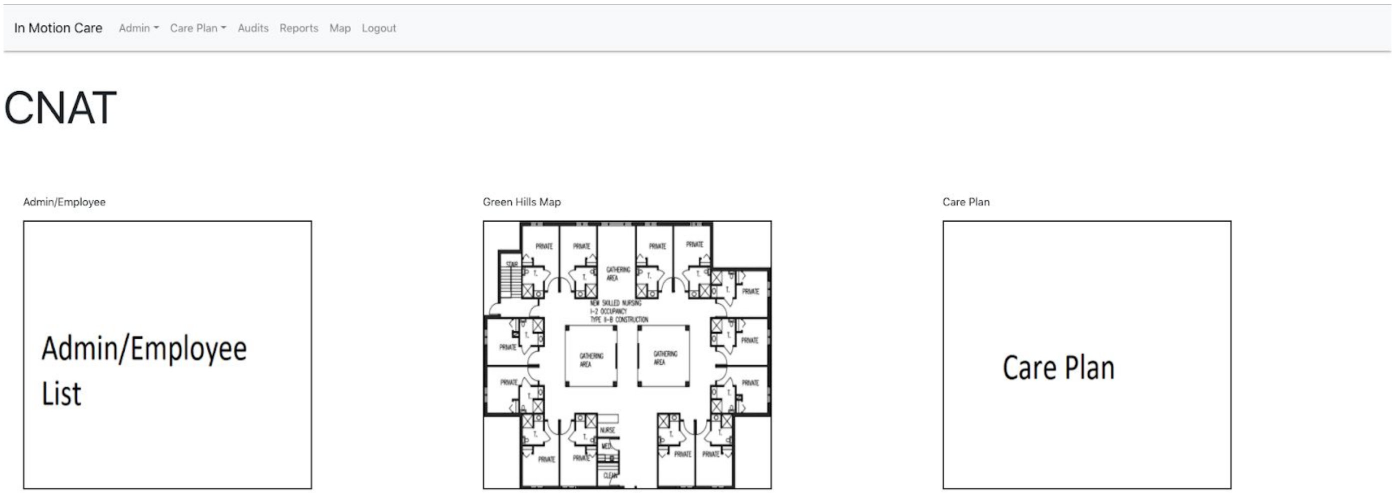

CNAT
Administrative
Care Plan
Map
Creation/Deletion of Users
Facility Statistics
Creation/Deletion of Positions
Alteration of user statistics
Creation of custom care plan tied to resident room.
Edit the care plan of a resident on the fly.
Shows who has changed the care plan and who has accepted those changes.
Display location of tracked users in facility.

Figure 2. Functional Decomposition of CNAT webpage

The figure above shows the general page flows of the application. When a user logs in, the user will be directed to a hub page with access to the three different sections of the application depending on user permissions shown below in Figure 2.



CNAT

Admin/Employee

Admin/Employee List

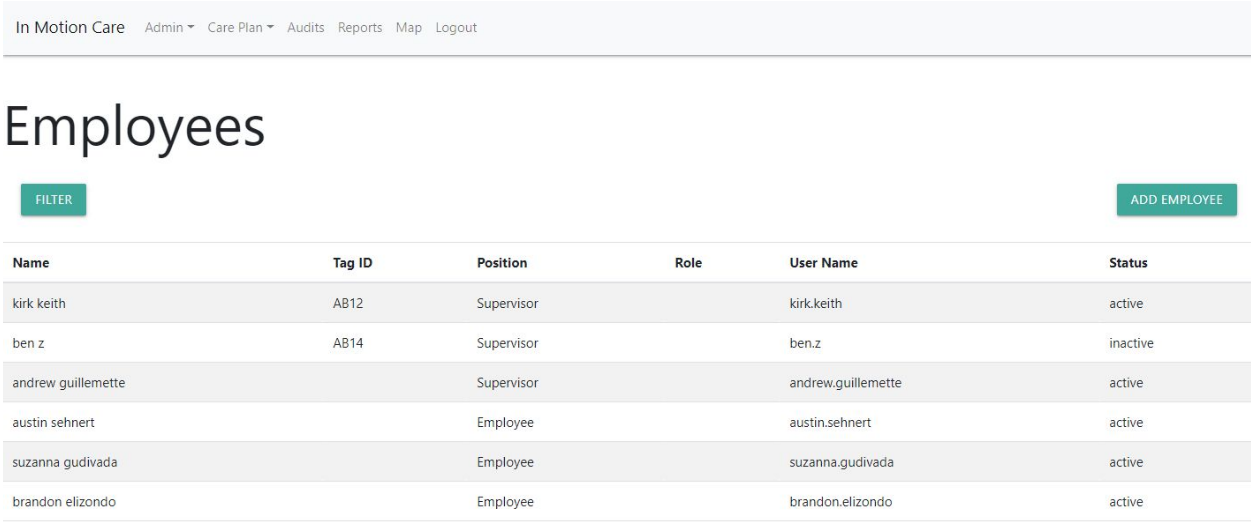
Green Hills Map

Care Plan

Care Plan

Figure 3. Hub/Homepage with easy access to most used pages

From the hub, the user can navigate to an admin page which allows system administrators to view, edit, add, and delete information pertaining to the healthcare staff working at Green Hills Community Center. As shown in Figure 4, this administrative page contains information pertaining to employee status, role, position, and user information. The information for employees is retrieved through an API call to the back-end on page load.



Employees

Name	Tag ID	Position	Role	User Name	Status
kirk keith	AB12	Supervisor		kirk.keith	active
ben z	AB14	Supervisor		ben.z	inactive
andrew guillemette		Supervisor		andrew.guillemette	active
austin sehnert		Employee		austin.sehnert	active
suzanna gudivada		Employee		suzanna.gudivada	active
brandon elizondo		Employee		brandon.elizondo	active

Figure 4. Employee Page within the Admin functionalities

The next section is the live map, which shows the location of nurses in the facility. The current project only has data for a small section of the facility due to the current scale of the project. To receive this data in real-time, a socket connection is established with the back-end to allow back-and-forth communication. Healthcare workers are displayed as a colored circle on the map and equipment that is being tracked is shown as a colored triangle. At the time of creating this document, there is no access to the data from the nursing home due to safety concerns about the COVID-19 epidemic. To display information on the map, legacy data is used and played back over the socket connection to show how the map can be used.

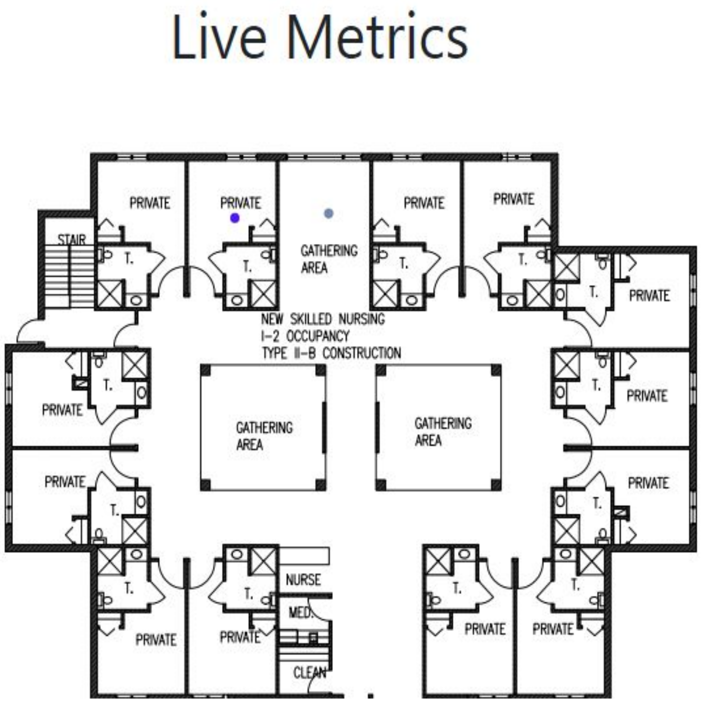
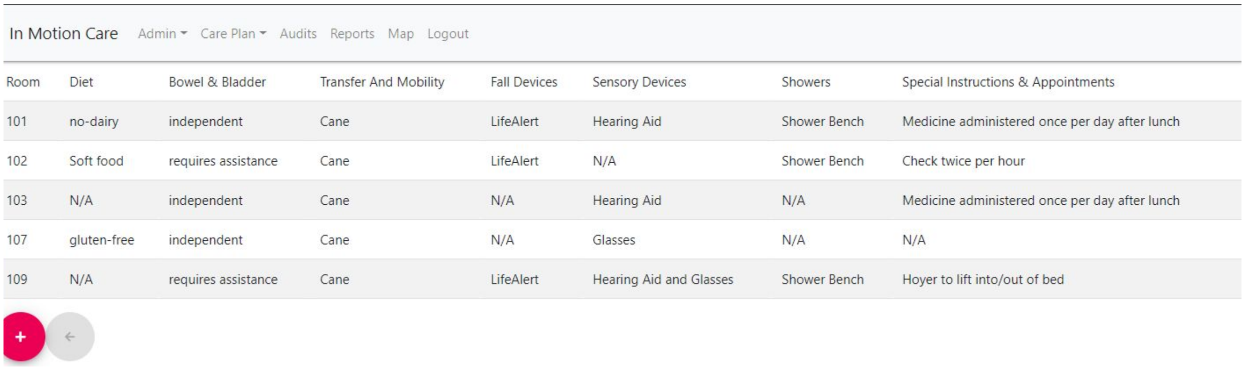


Figure 5. Live Map

Finally the care plan page allows healthcare staff to view and edit individualized care for residents at Green Hills. The care plan for each resident has seven different sections: diet, bowel & bladder, transfer & mobility, fall devices, sensory devices, showers, and special instructions & appointments and shown in Figure 6.



Room	Diet	Bowel & Bladder	Transfer And Mobility	Fall Devices	Sensory Devices	Showers	Special Instructions & Appointments
101	no-dairy	independent	Cane	LifeAlert	Hearing Aid	Shower Bench	Medicine administered once per day after lunch
102	Soft food	requires assistance	Cane	LifeAlert	N/A	Shower Bench	Check twice per hour
103	N/A	independent	Cane	N/A	Hearing Aid	N/A	Medicine administered once per day after lunch
107	gluten-free	independent	Cane	N/A	Glasses	N/A	N/A
109	N/A	requires assistance	Cane	LifeAlert	Hearing Aid and Glasses	Shower Bench	Hoyer to lift into/out of bed

Figure 6. Care Plan

Each of these sections has its own pop-up modal to guide a healthcare worker through updating the care plan of a resident to meet the standards of Green Hills as seen in Figure 7 below. After any change is made to the care plan, a log is kept of that change recording the user who made the change and the time that the change was made.

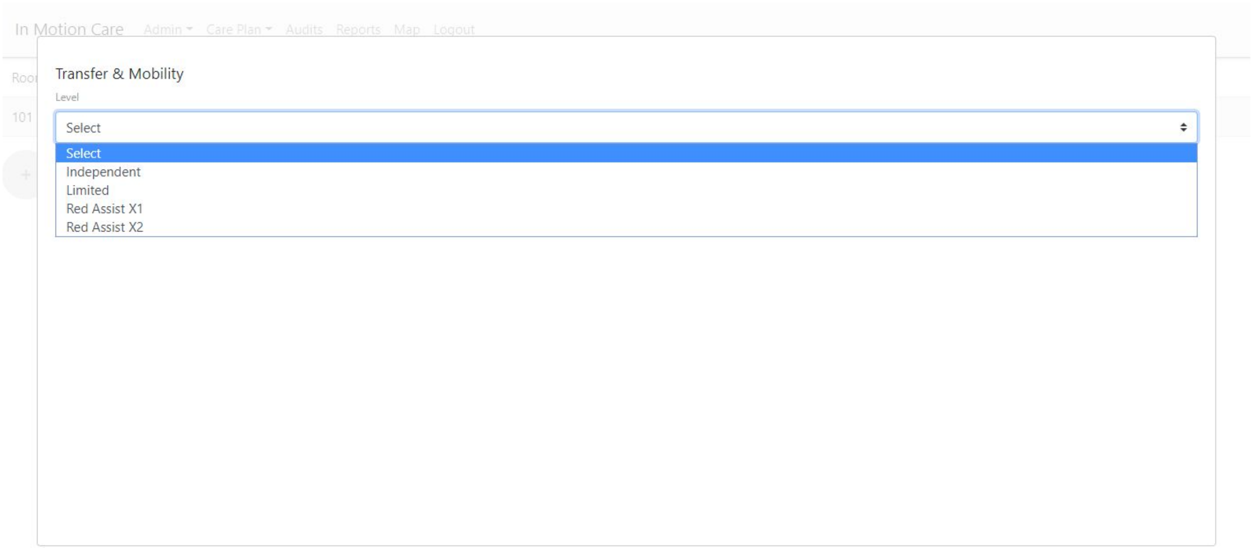


Figure 7. Care Plan Transfer & Mobility Modal

3.2 Back-end Design

The back-end for this project focuses around a Java Spring Boot Server application. Java and Spring Boot were chosen due to the client company already using Spring Boot with other systems and products as well as our group's familiarity with the system. Furthermore, Spring Boot allows us to create a system that can be modified and run without issues.

As part of the design, the back-end was made to communicate with both the front-end and the database. Communication with the front-end was achieved through the use of two different mediums: HTTP Requests and Websockets. To use these methods, the back-end is broken up into two keys parts: the controllers and repositories.

To communicate with the front-end using HTTP requests, POST functions are created within the controller, targeting specific URLs or channels that the front-end contacts directly. Once these channels are contacted and the front-end and back-end are connected, the controller contacts its respective repository. The repository executes a SQL query to the database which is then returned in a way that the back-end can properly manipulate or format for the front-end to see. The data is then sent to the front-end and the connection is closed.

For the Websockets, a target URL is created in a similar way, however, the connection remains open until it is purposely closed by either the front-end or the back-end. For this action, a message can be sent to the back-end at any time. When a message is sent, the back-end can respond by broadcasting to all those connected to the channel. In this case, we use it to transmit map data updates whenever the front-end would like to show new map movement. Currently, this connection is only closed when the front-end views a different page or the back-end shuts down.

3.3 Database Design

The bulk of CNAT's information is stored within a MySQL database. The database handles user information, the permissions granted to the users, tags, equipment, location data, and the care plan information.

In any table where data can be added or altered, an item within that table has actor, timestamp, and version columns. These columns provide the functionality of logs, because if a user were to alter an existing entity, the database would capture the user making the change, the current date and time, and would update the version. This allows a user with supervisory permissions to easily retrieve a log of changes made within the database.

The database is also built with modularity in mind. As the healthcare industry is constantly evolving, the database is built to accommodate new additions without a complete restructuring of the schema. To accomplish this, the *user* and *care plan* tables have several foreign attributes from other tables that build out an entire entry. For example, the user has a position within the facility, yet not all facilities will have the same position names or privileges. New positions can be created within the *positions* table by an administrative user, and then assigned to a user upon creation. Likewise, no longer relevant positions can be "removed" by setting them to inactive. The database schema can be viewed in Figure 7 below.

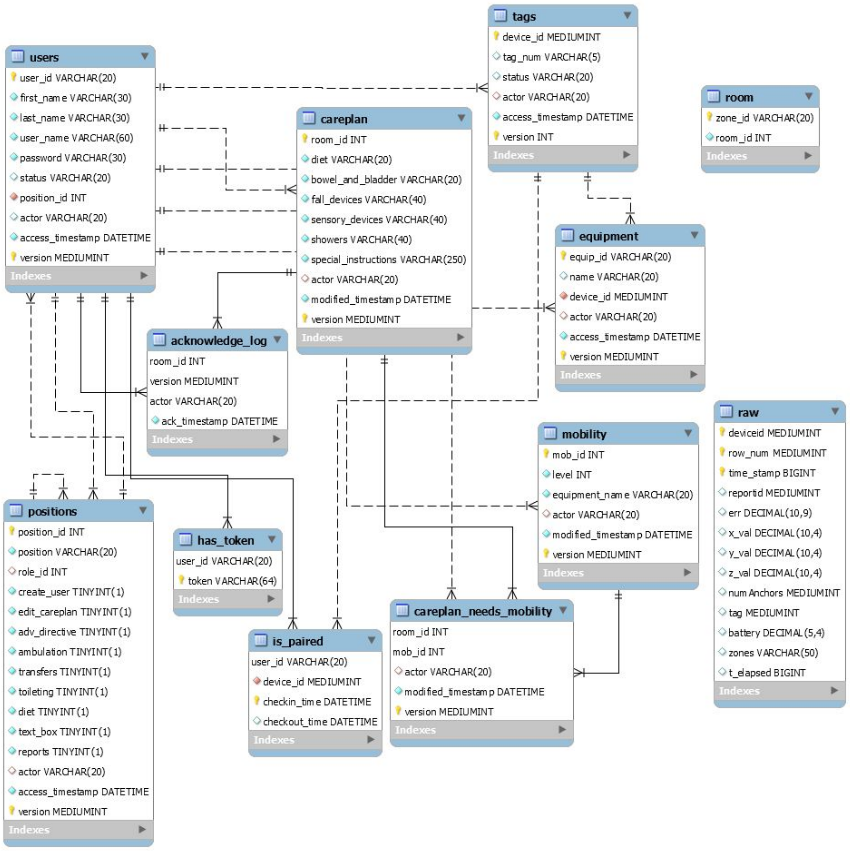


Figure 8. Current Database ER Diagram

4. Testing Process and Testing Results

In order to test operations on the back-end of the system, we used Postman as an on-the-fly API testing unit. Postman was used to test all HTTP requests to the back-end. This was done by running the server on a local machine where Postman was also available.

With Postman, a local URL was targeted and the information received within Postman was compared to the expected values.

To test active movement on the map, a Python script was created to upload data from a csv file into the appropriate MySQL table. The script uploads data into the database in .2 second intervals, simulating the update rate of the WISER anchors. This was used in tandem with the front-end and back-end to ensure data was both uploaded correctly and received in an usable fashion by the front-end.

5. Related Products

Currently, Centrak offers a similar product to what In Motion Care hopes to deliver. Centrak's "Real-Time Tracking System" uses a combination of IR, RFID, and other technologies to create a personnel and equipment tracking system specifically for healthcare facilities. Their combination of multiple sensor technologies allows for accurate tracking of personnel, even within non-optimal environments for signal propagation.

In Motion Care's technology currently only utilizes WISER System's RFID sensors. Where their product differs, is they hope to deliver updated care plans to nurses based on proximity to a patient's room alongside the personnel and equipment tracking capabilities that Centrak currently provides.

Appendix I: Operation Manual

Back-end Setup

1. Download the back-end program from the project repository to your preferred server system (typically one setup with Linux). The system must be properly set up for both outgoing and incoming communication.
2. To begin running the back-end, navigate to its top-level folder. Here you can see multiple folders, including: mvnw/ and src/.
3. Call the following commands:
    a. mvn clean
    b. mvn package
4. At the end of the output for the mvn package, a folder called "target" will be created with a .jar file within it.
5. Call the command: java -jar target/"name of the created jar"

6. After around 40-45 seconds, the Spring Boot server back-end will be up and running.

Front-end Setup

1. Download the front-end program from the project repository.
2. In the case of running the application locally
   a. Node.js must be installed locally and configured to run correctly
   b. Run the command npm run start
   c. Run the command react-scripts start
3. In the case of running on a remote server and for the application to be served to the user
   a. The application should be placed on the back-end server and accessible via HTTPS
   b. Node.js must be installed and configured on the server to serve content over HTTPS
4. The URL should be given to users who need access to the site

Database Setup

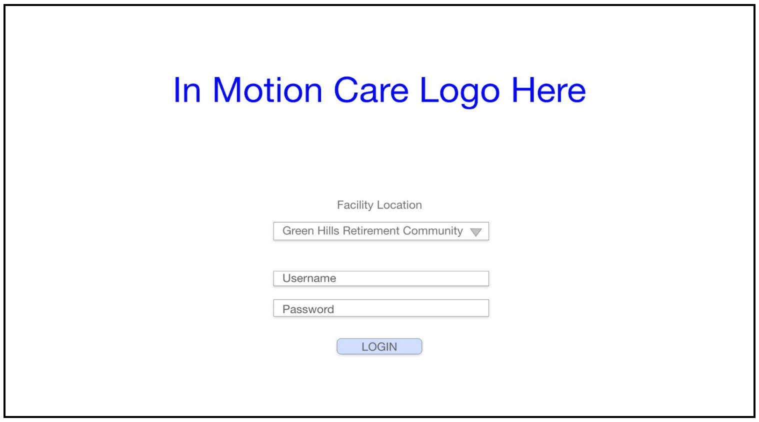
1. Download DBSchema.sql and positions.sql from the project repository.
2. Open DBSchema.sql and change the first line to be the name of the target database.
3. Through MySQL workbench or another SQL management system, execute the script on the target database.
4. Open positions.sql and change the first line to be the name of the target database.
5. Repeat step 3 for positions.sql
6. Open MySQL workbench or another SQL management system and check that all default positions have been inserted into the database.

Appendix II: Initial Versions

Over the course of two semesters our team has worked on four versions of deliverables:

Version I:
Initially we were more focused on a dashboard with live location and a CNA location playback video feature. Dealing mainly with WISER API only to gather location and legacy location.

GUI Screen Mockups:



In Motion Care Logo Here

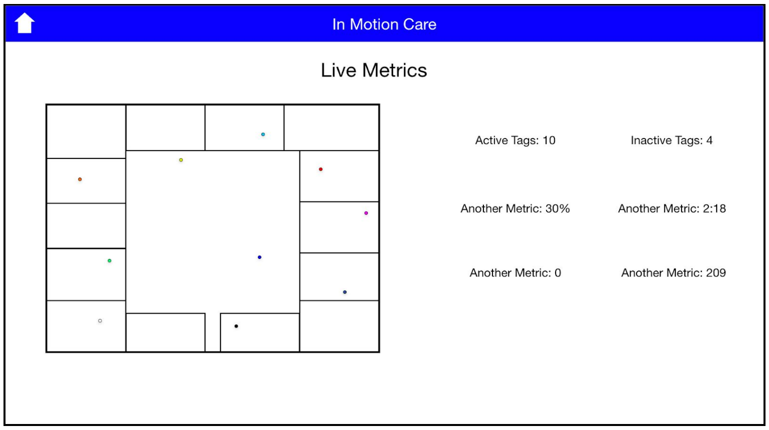
Facility Location

Green Hills Retirement Community

Username

Password

Figure 9. V1 Screen Design



In Motion Care

Live Metrics

Active Tags: 10

Inactive Tags: 4

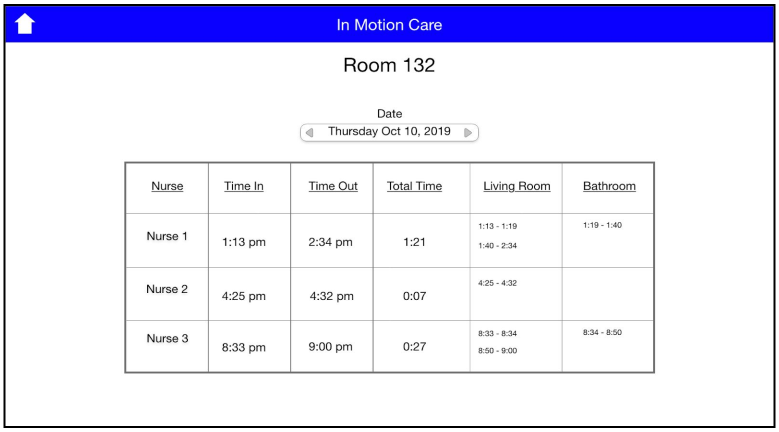
Another Metric: 30%

Another Metric: 2:18

Another Metric: 0

Another Metric: 209

Figure 10. V1 Screen Design



In Motion Care

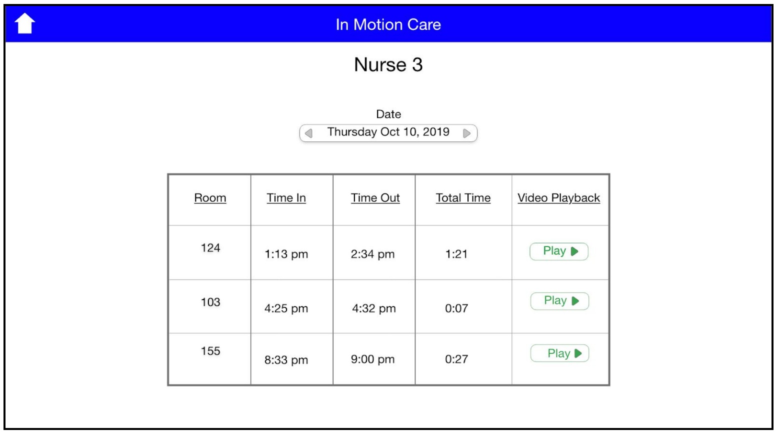
Room 132

Date

Thursday Oct 10, 2019

Nurse	Time In	Time Out	Total Time	Living Room	Bathroom
Nurse 1	1:13 pm	2:34 pm	1:21	1:13 - 1:19 1:40 - 2:34	1:19 - 1:40
Nurse 2	4:25 pm	4:32 pm	0:07	4:25 - 4:32	
Nurse 3	8:33 pm	9:00 pm	0:27	8:33 - 8:34 8:50 - 9:00	8:34 - 8:50

Figure 11. V1 Screen Design

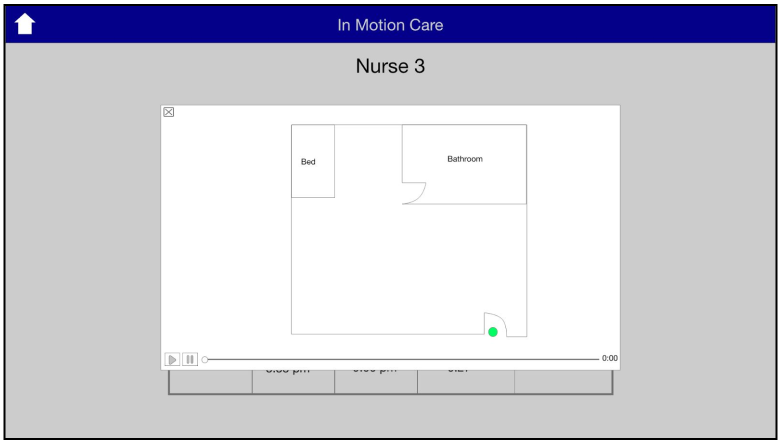

In Motion Care

Nurse 3

Date
Thursday Oct 10, 2019

Room	Time In	Time Out	Total Time	Video Playback
124	1:13 pm	2:34 pm	1:21	Play ▶
103	4:25 pm	4:32 pm	0:07	Play ▶
155	8:33 pm	9:00 pm	0:27	Play ▶

Figure 12 . V1 Screen Design


In Motion Care

Nurse 3

Figure 13 . V1 Screen Design

Once we developed the screens mentioned above in v1, our client pitched the website to the founders of Green Hills Community Center who critiqued the webpage and offered suggestions leading to our next version.

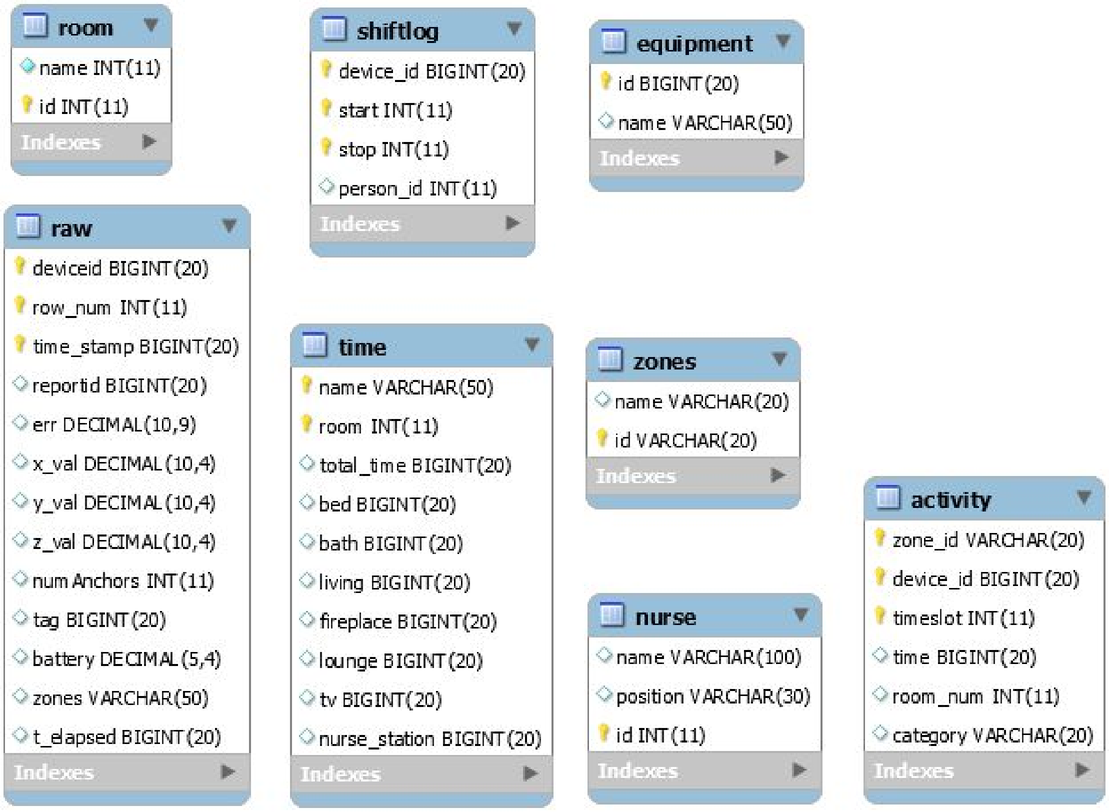


Figure 14 . V1 Database Design

Version II:

With the ideas put out by the founders of the Green Hills Community Center, our client requested to separate information from the map and add a daily summary page. Also to integrate an already existing system called the Care Plan into our webpage. The Care Plan application that our client had was in JavaScript which we then converted into ReactJS. We were no longer developing the location playback video feature.

We were requested to create a homepage with easy access to pages in a navigation bar style. Our client requested a timer on each room. When a tag enters the room the timer would reset, else the timer would turn the room red, indicating to the staff the particular room needs to be visited. Also have a floating mini-map of the facility on the bottom right corner.

Screen Designs:

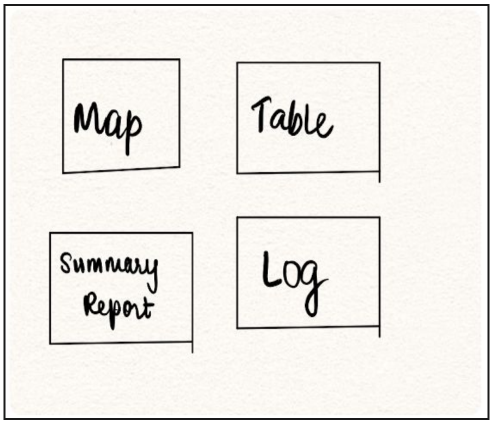


Figure 15 . v2 Screen Design

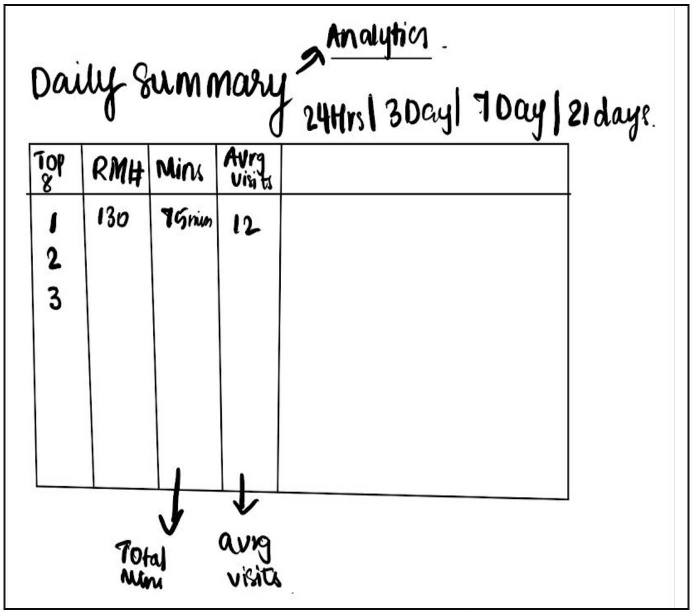


Figure 16 . v2 Screen Design

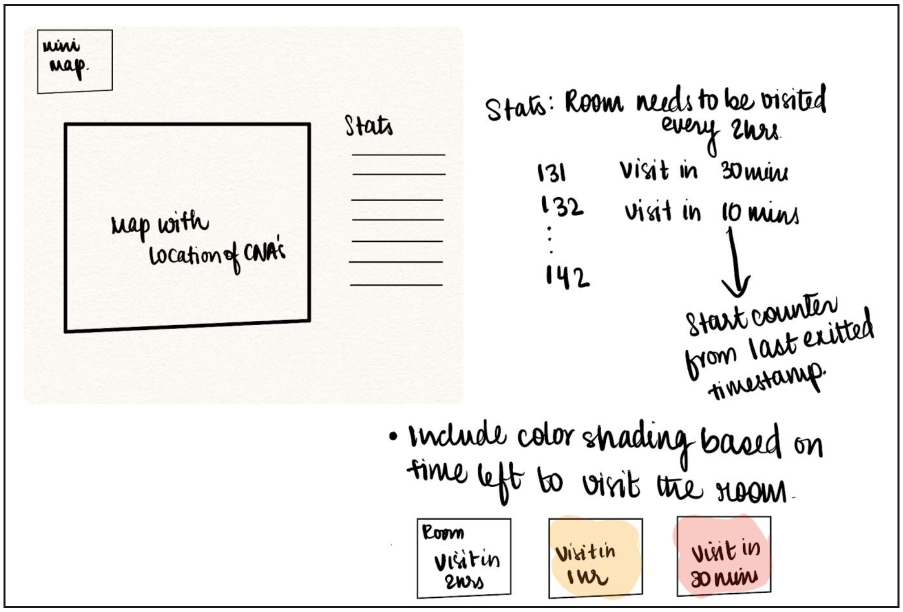

mini map.
Map with location of CNA's
Stats
Stats: Room needs to be visited every 2hrs.
131 Visit in 30mins
132 visit in 10 mins
:
142
Start counter from last exitted timestamp.
• Include color shading based on time left to visit the room.
Room Visit in 2hrs
Visit in 1hr
Visit in 30min

Figure 17 . v2 Screen Design

Version III:

As the team was developing above requirements going into the Spring 2020, we had the same screen flow as v2. Our client scraped out the room timer and mini-map from our deliverables. We were more focused on the Care Plan and worked closely with the founders of Green Hills Community Center. We continued to develop per request on the Care Plan side. We developed the administrative functionality and the hub page.

Database Design v3:

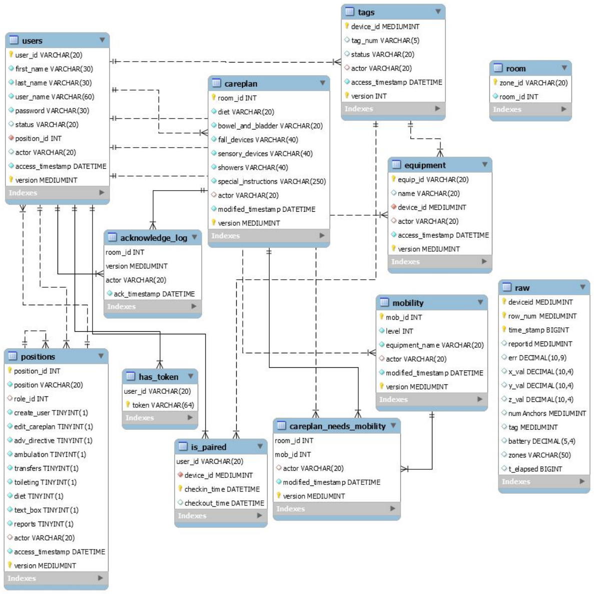


Figure 18 . v3 Database ER Diagram

Version IV:
Due to the COVID-19 situation, we are finishing up our deliverables of the project that we've made so far and we started developing our final and the most challenging deliverable, the real-time location tracking map that shows legacy location data of the assets. We are no longer working on the new user login system, though we have the front-end skeleton and back-end functionality in place for it. Unfortunately we were unable to continue with our testing phase in the Green Hills Community facility being locked down. As an alternative we proceeded our testing phase by testing the live data displayed on the map webpage against historical data set provided by the client to confirm the accuracy of our system.

Appendix III: Engineering Standards and Design Practices

ISO/IEC/IEEE International Standard - Systems and software engineering--Measurement process, IEEE 15939-2017

ISO/IEC/IEEE International Standard - Systems and software engineering -- Software life cycle processes, IEEE 12207-2017

ISO/IEC/IEEE International Standard for Software Engineering - Software Life Cycle Processes - Maintenance, IEEE 14764-2006

Health Informatics—Point-of-care medical device communication Part 10201: Domain Information Model, IEEE Std 11073-10201-2018

Software and Systems engineering —Software testing — Part 4: Test Techniques, IEEE 29119-4:2015(E)

Appendix IV: References

Centrak.com. (2019). CenTrak | Accurate Hospital Real Time Location System. [online] Available at: https://centrak.com/ [Accessed 8 Oct. 2019].

WISER Systems. (2019). WISER Systems. [online] Available at: https://www.wisersystems.com/ [Accessed 6 Jul. 2019].