Report for Detailed Design Phase
Rapid Prototyping of Computer Systems
05-540, 05-872, 18-540, 18-745, 39-648

Editors: Allen Chao, Joe Fifty, Yi He, Vineet Kanetkar, Chinmayee Purohit, Dingze Wang, Mingyu Hu, Yuling Liu, Anyi Diao.

1. Overview

Alzheimer's is a disease that degenerates cognitive functions starting with short term memory, moving to activities, moods and emotions, paranoia and hallucinations, logical thinking, and finally long term memory. We are developing a system for Alzheimer's patients, caregivers, and doctors to better manage and analyze Alzheimer's symptoms. In this phase each team went from the initial conceptual design to a more detailed design, determining the functionalities that will be implemented and how they will be implemented.

2. Conceptual Design

2.1 Problem Definition

Alzheimer's patients often have difficulty navigating everyday tasks. This and risks such as falling or wandering place a large burden on the people responsible for the Alzheimer's patient's care. Many caregivers are forced to seek outside help because they cannot sustainably provide the care their loved one needs. Physicians for Alzheimer's patients attempt to gauge the progression of the patient's disease, but with imperfect information and tests there is a less than desirable level of granularity in the assessment.

The following sections describe the current issues faced by Alzheimer's patients and caregivers and the functional requirements to build a system that meets these identified needs. The scenarios are divided by the progression of the disease, with short term memory first affected and long term memory least affected.

2.1.1 Baseline Scenario

Jillian is a 68 year old patient with advancing alzheimer's symptoms. She has been struggling with her memory for quite a while, but was only diagnosed six months ago. She began experiencing gaps in long term memory as well as mood swings about a year ago. Jillian is cared for by her husband, Mark, and her health care providers. Jillian is hard working, loves to socialize, and loves to go out to eat, but she can be easily frustrated with her condition.

2.1.1.1 Short Term Memory

Jillian has been having trouble remembering things like her keys, wallet, or phone as she's leaving the house which can cause problems for both her and Mark. Her deteriorating memory also causes her to struggle with remembering changes to her routine. Jillian is trying to cope with this issue by writing down reminders, upcoming events, and other information she wants to remember in her notebook. Unfortunately, this is not always helpful as she doesn't carry her notebook with her constantly, and it is not always reasonable to constantly check the notebook.

Mark tries to help her by reminding her of her keys or medication and by patiently answering all her questions, even if he already answered them 15 minutes ago. Jillian doesn't enjoy being this dependent on Mark, but she doesn't know how to make the situation better.

2.1.1.2 Activities

Physical activity is a key component of Jillian's routine, and she walks or runs everyday. This is something she's always enjoyed and wants to take advantage of the ability to do for as long as possible. Although Mark's biggest worry is that Jillian may go wandering, she still continues to go on walks by herself. In order to assuage Mark's concerns, Jillian makes sure she lets him know where she's going, because she knows that it would not be fair to Mark if she were to wander off without letting him know. It's important to her that she do all that she can to reduce the burden on Mark. Mark feels some security because the Apple watch he gifted her allows him to check her location. The watch is particularly important to Mark's peace of mind because, as of recent, Jillian has had two incidents in which she was separated from Mark, and he was worried that she may not be able to find her way back to him. Mark is concerned that as her symptoms progress, she may begin wandering in the house and possibly leave the house at night. This is particularly concerning as Jillian usually takes her Apple watch off at night to charge it, meaning it wouldn't be on her body.

2.1.1.3 Moods and Emotions

Often times as Mark and Jillian are sitting and conversing, Jillian will suddenly become agitated. Mark has trouble tracking what causes this agitation and how to calm her down. Sometimes he continues to talk to her and try to reason with her, but that often does not help. He's found that if a family member or friend calls, that can distract her and help her recover. That's not always possible, however, and in those situations he often just has to wait it out, which can be painful as Jillian usually has a sweet and happy personality.

2.1.1.4 Paranoia / Hallucinations

Jillian has begun to display symptoms of the sundown effect. She may hallucinate and relive past memories vividly or become paranoid about her safety. One night, while relaxing with Mark after dinner, she begins to relive a New Year's dinner party they hosted. She goes to the kitchen and starts setting their fancy china plates on the table. Mark doesn't realize Jillian is hallucinating for a while, so he has no context for her actions or her asking him to hurry up and get ready. Once he understands what's happening, he struggles with breaking it to her that she's hallucinating and there's no dinner party happening.

2.1.1.5 Logical Thinking

Jillian is sometimes confused with complex tasks that involve many steps to complete, such as preparing dinner and setting up tables. She used to solve sudoku puzzles with her husband Mark in their free time, but she starts to have trouble working with numbers now, which is upsetting to her. Mark has taken over managing the house budget, something that Jillian managed for most of their marriage.

2.1.1.6 Long Term Memory

Jillian has begun to have difficulties with her long term memory. Last month, she forgot the name of their first dog together, and just two days ago, she forgot that they had 2 daughters. Mark is scared that she'll soon forget details about herself.

2.1.2 Key System Requirements

Category	**Item**	**Functionality**
Reminders	Medication	Caregiver is reminded to give patient their daily medication throughout the day
	Medical appointments	Remind caregiver of patient's upcoming medical appointments
Notifications	Caregiver	Caregiver should be / can be notified in the following situations: ● Patient falls ● Patient leaves the home ● Patient is wandering ● Patient is hallucinating ● Patient is agitated Caregiver should also be notified with doctor's suggestions in appropriate scenarios such as an agitated mood or hallucination.
	Doctor	Doctor is notified when: ● Patient suffers serious physical injury ● Patient suffers severe agitated episode ● Patient suffers rapid cognitive design
Data Collection	Moods	Detect varying moods and log the moods for the doctor to review.
	Hallucination	Detect when patient starts hallucinating (based on speech patterns, heart rate, etc.). Log the patient's behavior for doctor to review.
	Logical Thinking	Record and analyze patient's performance in brain training exercises. Log the performance for doctor to review later.
	Activities	● Track patient's movement through the house, particularly at night. Log the activity so that the caregiver can review it the next morning. The doctor can

		receive summary reports that also provide an insight into patient patterns. ● Track if patient has taken daily medications at the correct intervals
Database	Activities	The patients activities are stored in the database so that patterns can be identified.
	Contacts	Stores key contacts (caregiver, doctor, children, emergency contacts)
	Medication Information	List of patient's current medications with dosage, frequency, side effects and issue mitigation solutions

2.2 Initial Solution Concepts

2.2.1 Table of selected technologies

Team	**Technology**
Doctor & Caregiver Interactions	Frontend libraries: ReactJS, Chart.js
Core Tablet	Samsung Galaxy Tab A 8.0''
Games & Assessment	Android Studio
Short Term Memory	Android Studio, Find It Tiles
Watch	Huawei Watch 2
Wandering Tracker	Shoe tracker
Home Sensor	Door sensor, pillow sensor, mat sensor
Server/Database	PostgreSQL, Django, RESTful API, GraphQL
Core Analytics	PostgreSQL, Django

2.2.2 Visionary Scenario

2.2.2.1 Short Term Memory

Jillian is a retired teacher who was recently diagnosed with Alzheimer's. Mark, Jillian's husband, is her caregiver. He notices that Jillian's symptoms have been gradually advancing, and she has been having forgetful episodes, especially in regards to remembering things like her keys, wallet, or phone. To help Jillian manage the forgetfulness, Mark adds sensors to her frequently lost items. Jillian can then use her app to make the sensor on her lost object ring so she can find her item. When Jillian uses the app, it logs the incidence so that her doctor can review it later. Jillian also fills out a short term memory form every morning so that Mark and her doctor can track her short term memory status.

2.2.2.2 Activities

Jillian has a hard time getting ready in the morning, often forgetting take her medicine. She spends the day at a care center socializing and doing activities, but she keeps a strict routine so she always knows what she'll be doing. Occasionally, Jillian wanders at night, which is extremely worrisome to Mark who has to go out and find her.

Part 1

One evening, Mark goes to bed while Jillian stays up to finish an episode of Wheel of Fortune. She gets distracted and begins to walk out of the living room. Jillian enters the kitchen, and because Mark has indicated to the system that she should not be there alone, the system logs her presence for the next mornings' review report. She notices a bag of coffee grinds on top of the fridge and as she reaches for them, she loses her balance - falling to the ground. Her wearable sensor detects her fall and immediately alerts Mark. The system then begins to countdown; if Mark does not acknowledge the alert within 60 seconds, an emergency contact will be alerted to Jillian's situation. Mark wakes up hearing the alert and rushes down to check on Jillian. Luckily, she is fine this time and sustained no major injuries. He helps her up from the floor and guides her to bed so she can rest.

Part 2

Another evening, Mark and Jillian are watching Wheel of Fortune together. Mark falls asleep on the sofa and Jillian gets up to use the restroom. Jillian opens the front door, an action that causes the system to set of a sequence of beeps to alert Mark and is also logged into the system. Unfortunately, Mark is deeply asleep and does not hear the beeps. As Jillian begins to wander down their driveway, her wearable device detects her departure and reminds her to notify Mark that she's leaving. Unfortunately, she doesn't notice the alert either. As she approaches the end of the driveway, her wearable device detects that she is more than 30 ft from Mark. Because Mark had set that as their range in the Nemosi system, he is alerted by his wearable that Jillian is wandering alone. He wakes up to the alert and is able to see that she has just passed their neighbor's house and rushes to bring her back inside.

2.2.2.3 Moods and Emotion

Mark sits down with Jillian after they have finished their breakfast. He begins talking to her, and Jillian becomes agitated. Mark doesn't know why and unsuccessfully tries to calm her down. Together, using the Nemosi smart assistant, Jillian and Mark make decisions on how the smart assistant should respond when Jillian is upset. Later, when Jillian becomes upset and agitated, the system detects her emotional arousal and sends an alert to Mark. He looks at either his phone or smartwatch to receive her mood change and is asked by the system to confirm Jillian's emotional valence (i.e. anger, sadness, etc.). Mark confirms with the smart assistant and the system automatically plays Jillian's favorite calming music. The system also provides Mark with doctor approved suggestions on how to manage the situation, including a calming guided breathing exercise. When Jillian is very agitated, the system notifies other loved ones to contact and distract her in the event that Mark is unable to assist her in the moment. The system logs Jillian's mood in a report which is available for Jillian's doctor to review and inquire upon in their next visit.

2.2.2.4 Paranoia / Hallucinations

One evening, Jillian begins to hallucinate that she needs to get ready for a dinner party with some old friends at her home. This was an event that already happened several years ago. She goes to the kitchen and begins taking out pans and food to cook dinner. Mark is upstairs and doesn't realize Jillian is hallucinating, so he doesn't stop her from starting to cook. Luckily, the Nemosi system detects via the wearable device and sensors that Jillian is under stress and is performing activities in a designated "danger area" so it sends an alert to Mark. He receives that alert but doesn't know what he should do to calm her in this situation. The system provides Mark with doctor recommended solutions under patient hallucination. Mark follows the instructions and is able to calm Jillian down and get her to put down the items. The system logs this hallucination event for later doctor review in Jillian's monthly check-in.

2.2.2.5 Logical Thinking

Jillian was recently diagnosed with Alzheimer's. She is sometimes confused with complex tasks that involve many steps to complete, such as preparing dinner and setting the table. She loved solving Sudoku puzzles with her husband Mark in their free time, but she is starting to have trouble working with numbers now, which is upsetting to her. To help keep her brain active and delay the advancement of symptoms, her doctor recommended a few home brain training activities. In the afternoon, the system reminds Jillian to do brain training exercises. She is provided with interactive, step-by-step instructions for each exercise. Jillian and Mark usually enjoy doing these activities together, and it gives them a chance to bond and interact. The system records Jillian's performance and logs the data for Dr. Smith to review.

2.2.2.6 Long-Term Memory

Part 1

Last week Jillian visited Dr. Smith for her monthly check-in appointment and was assigned a new medication. Two days after the treatment, Jillian forgets that she has two children. Within a week, Jillian has experienced her 7th incident of long-term memory loss. Mark is very worried about Jillian's condition and doesn't know what he should do. Luckily, the he can log the incident in the Nemosi system and Dr. Smith is notified in the application of Jillian's accelerated decline. The doctor calls Mark to bring her in for a follow-up appointment. The doctor

determines that Jillian needs to be taken off the medication as it is not producing the desired results.
Part 2
Jillian often forgets the same basic information about her life and asks Marks the same questions over and over. Mark stores this frequently forgotten information in Jillian's app along with pictures. Jillian can click on the pictures and hear a description recorded in the voice of her loved ones.

2.3 Conceptual Design

2.3.1 Selection Criteria based on Visionary scenario

Team	**Feature**
Doctor & Caregiver Dashboard	Doctor Dashboard ● Search Patient ● View Patient Trends ● View Patient Incidents Caregiver Dashboard ● Find Patient Location ● View Exercise / Sleep Trends ● Report Patient Incident
Core Tablet	● Family Album ● Play Calming Music ● Check Schedule
Server/DB	● PostgreSQL database ● RESTful API & GraphQL Endpoint
Core Analytics	● Text Message for Alerts to Caregiver ● Analyze Raw Sensor Data
Watch	● Fall Detection ● Physical Activity Tracking ● Hallucination
Wandering Tracker	● Shoe Tracker
Home Sensor	● Bedside Mat Sensor ● Bed Pillow Sensor ● Bathroom Activity Sensor ● Kitchen Activity Sensor ● Front Door Sensor

Games & Assessment	● Cognitive Assessment Games
Short Term Memory	● Find It Tiles ● Memory Check Questions

2.3.2 Product Design Specifications

Find It Tiles - Short Term Memory
Tiles can attached to easily and commonly misplaced items such as keys, wallets, and TV remotes. These objects can then be located using “Find It” buttons placed around the house or using the Patient tablet app. Pressing the “Find It” button or the app will cause the tile to ring, leading the patient or caregiver to the object.

Memory Check Questions - Short Term Memory
Patients can do quick sets of memory check questions to monitor how their short term memory is faring.

Bedside Mat - Home Sensors
The bedside mat utilizes pressure sensors to detect when a patient gets out of bed. This information is then sent through the data analytics team before appearing in the caregiver dashboard.

Bed Pillow - Home Sensors
The patient’s pillow utilizes pressure sensors to detect if a patient is laying down in bed. If they are not, this information is sent through the data analytics team before appearing in the caregiver dashboard.

Bathroom - Home Sensors
The bathroom uses RFID and PIR sensors to detect if the patient is in the restroom while still maintaining a level of privacy. If the patient is in the restroom an extended amount of time at night (over 15 minutes) the caregiver is alerted.

Kitchen - Home Sensors
The kitchen utilizes a motion sensor to detect if the patient is in the kitchen at night. It also utilizes an infrared camera to detect if the stove has been left on, unattended. The caregiver is alerted whenever the patient enters the kitchen at night as well as when the stove has been left on.

Front Door - Home Sensors

RFID sensors and line of sight sensors at the front door work in conjunction to detect if the patient leaves the home. In such situations, the caregiver is alerted.

Patient Shoes - Wandering
The patient's shoes are equipped with GPS technology to track the patient's location. They come with a charging mat so that they are always charged. The caregiver can view the patient's location information through their dashboard.

Fall Detection - Watch
The patient's watch can detect if they have fallen and this information is passed through the data analytics to alert the caregiver.

Physical Activity - Watch
The watch tracks the steps the patient takes each day. This information is then made available to the caregiver and doctor to ensure the patient is getting sufficient physical activity.

Hallucination - Watch
The patient's watch measures certain biometrics to identify patient hallucination, paranoia, and distress. This information goes through analytics before being logged into the patient incidents data set and alerting the caregiver.

Cognitive Assessment - Games
The patient can play cognitive assessment games through the patient dashboard to help monitor and potentially slow down cognitive decline. The games help with logical thinking as well as memory. Their scores are available to the doctor for the doctor to monitor the trends.

Family Album - Long Term Memory
The patient can look through a family album and review information about their family members to help keep memories fresh. This can be done through the patient dashboard.

2.3.3 HCI

For Phase 1, we began by researching Alzheimer's to gain an understanding of the issues faced by Alzheimer's patients, caregivers, and doctors. We synthesized the information we learned from readings and interviews into an affinity diagram. Using our research, we developed an initial baseline scenario. After receiving feedback on the baseline scenario, we divided the scenario into different sections based on areas affected by the disease. Considering our baseline scenarios, we brainstormed ways technology could serve Alzheimer's patients, caregivers, and doctors to alleviate the pain points we identified. We then created our visionary scenario in the same format of the baseline scenarios. Finally, we created a list of functional requirements needed to achieve our visionary scenarios.

2.3.4 Dashboard / Data Visualization

In phase 1, we designed dashboard for three personas (patient, caregiver, doctor) on four types of devices (mobile phone, watch, tablet and desktop), which makes up 12 independent applications. Tablet and desktop apps are designed to conduct complex tasks and view graphs, while mobile app and watch app mainly aim at notification, warning and tracking. In phase 2, due to implementation time, we need to reconsider the necessity of doing all these apps for all personas and refine the interface for real implementation. The discussion for dashboards in phase 2 will be included in detailed design part of 'caregiver and doctor interaction' and 'core tablet' groups.

2.3.5 Short Term Memory

In regards to the Alzheimer's field, Short Term Memory is quantified as in terms of measurement and assistance. Our main throughout this project was to assist Alzheimer's patients dealing with deteriorating short term memory. We implemented systems to help patients find lost items as well as a their lost tablet that they use for other modules in this project. Alzheimer's patients lose their memory capabilities over time and it's important for doctors to measure cognitive decline. As a result, added a diagnostic test which has been clinically proven in the Alzheimer's field. We also added some of our own elements to this test that we felt were important in measuring cognitive decline.

2.3.6 Activity Sensors

The activity sensors group from Phase 1 particularly identified the unusual activities that are synonymous with Patients of Alzheimer's. These activities pretty much covered all the critical movements of these patients that require extra care and monitoring. The activities that were thus identified were falling, wandering, misplacing objects, sneaking out.

There is always a risk of these patients falling due to hallucinations, seizures, fits etc. On such occasions a caregiver is possibly not nearby and hence it is important to alert them in such conditions to provide the patients with timely help. In order to realize this the team proposed the use of a smart watch. The accelerometer present in the watch can help to detect a fall

Misplacing objects such as remote, keys, mobile phones are common. Hence the use of tiles were suggested by the team. These tiles work on RFID and can be teamed into a network to detect their locations accurately.

On interviewing our client we realized that she loves to walk and walks regularly. However these patients may face disorientations anytime and could potentially deviate from their regular route. Apart from this they could sneak out of the house unknowingly at night. Under such conditions, it is necessary to track their locations with respect to the caregiver. In order to implement this the team proposed footwear tracking as in most cases the patient wouldn’t leave out their footwear before leaving the house. Alongside it is possible that the patient may walk out of the house or use the restroom or kitchen at night. These are a few places where the patient can hurt himself/herself typically due to slipping or falling. Hence an alerting mechanism to the caregiver was proposed by the team.

2.3.7 Mood and Emotion Sensors

In phase 1, for detecting the mood of the patient various technologies were surveyed. Some of the sensors researched were Audio sensors, Galvanic Skin Responses(GSR), Video Sensing, Heart Rate Sensors and Motion sensing.

2.3.7.1 Audio Sensors

Vokaturi: It can detect 5 emotions including if you are Happy, Sad, Afraid, Angry or Neutral. You can use the SDK to develop an app for any platform (iOS, Android, Windows, MacOS, Linus, Raspberry Pi, Odroid). The accuracy of the SDK is 66.5-76.1% on an open source code for free.

2.3.7.2 Galvanic Skin Responses (GSR)

Seeed Grove GSR Sensor: The Seeed Grove sends its data to a microcontroller via a wired connection. To use this sensor, our team would need to wire it to a microcontroller and connect to that microcontroller via Bluetooth or Wi-Fi. The microcontroller, in turn, would need an appropriate battery and a Bluetooth or Wi-Fi module if not built-in.

2.3.7.3 Video Sensing

Affectiva: The Affectiva SDK can be used on any platform including IOS, Android, Web, Windows, Linux, MacOs, Unity, or Raspberry Pi. Using this application, we would require very low memory space, as the data does not necessarily need to be stored. This sensor does not require cloud storage as all the data will be stored on the device.

2.3.7.4 Heart Rate Sensors

Vivosport: This is a Smartwatch which can be used for Activity Tracking purposes. It has an open source API but requires a one-time registration fee.

2.3.7.5 Motion Sensing

There are no readily available technologies for motion sensing, that could incorporate the detection of emotions and moods of a user.

Among these choices, we concluded that the video sensing technology and the detecting emotions with heart rate were the most viable options. Since, video sensing cannot monitor the patient continuously, for Phase 2 we decided to opt for tracking emotions like stress, anger using heartbeat sensors in a smart watch.

2.3.8 Hallucination and Paranoia Sensors

There are two types of hallucination, one is an auditory hallucination, and the other is visual hallucination. If someone has auditory hallucination, he/she may sense that the sounds are coming from inside or outside your mind. He/she may hear the voices talking to each other or feel like they're telling you to do something. If someone has visual hallucination, he/she may sure he/she seen something, then realize it's not actually there.
Since hallucination may cause anger, nervousness, anxiety, etc. these physiological responses affect the heartbeat, blood pressure, and pulse. Besides, hallucinations may also intensely disturb to patients' sleep. If we can also detect the sleep quality of a patient, we have more data to tell the level of the hallucination. Clinically, a sleep study (polysomnogram) is recommended if you have hallucinations. A sleep study involves attaching wires and equipment to your head and body to chart your brain waves, heartbeat, and breathing as you sleep. It also records how your arms and legs move. Thus, we decided to use health band to get the data of heartbeat, sleep quality, and stress.

2.3.9 Logical Thinking Sensors

In phase 1, the Logical Thinking Sensors group provided technologies related to both software and hardware, designed a sub-system to measure the patient's ability of logical thinking, we used different types of games to let patient play with them, and used different types of sensors to record patient's movement, and we used microprocessor to transfer the data that we got to our system to be analyzed by other groups. In phase 2, due to the changes of our system's architecture, we would measure patient's logical thinking ability using software in a tablet rather than a sub-system of the combination of software and hardware. So we would generate some games running on the tablet, and we would record the performance of patient directly using the software, and gather data to analyze the patient’s logical thinking ability.

2.3.10 Long Term Memory Sensors

In Phase 1, Long-term memory sensors group was supposed to examine the long-term memory of patients by figuring out ways to test this.The group initially planned to have a voice assistant to be placed at different corners of the house that would be scheduled to carry on tests. But this appeared to be an intimidating assessment for the patients as random voices can scare them off and was finally discarded. The team eventually narrowed down it’s requirement to one good smart device and associated accessory devices. Tablet was the best fit and was selected over all small screen mobile phones because of its ease of use and larger battery life. It was planned that there would be a couple of games dealing with each type of memory. In Phase 2, the functionality of the group got split into the Core Tablet Group and the Games and Assessment Group. Taking into consideration the changes in the system architecture, there were some minor changes in the approach towards dealing with long term memory issues. The idea of having a voice assistant did not seem viable, hence instead of that, interactive games would be used in the final product. Because of the time constraints, a few of the games would be bought directly while some other games that are not very readily available would be developed from scratch like games dealing with episodic memory.

2.3.11 Operating Systems and Databases

In phase 1, the OS/DB is mainly responsible for research on a wide range of technology choices and choose the technologies that are suitable for implementing the system for the benefit of both easy to use and efficiency. Therefore, we have decided to use the local server instead of cloud services for better privacy and security issues. We list suitable programming languages to program the system. We choose Progres as the database technology because of its easy to use and extensibility. And finally, we designed the software architecture and improve it over time to meet the requirements. In phase 2, we move on from the former phase to detail design and some implementation work. During this phase, we worked with other teams closely to collect feedbacks and designed the data templates to meet all the other team’s needs and at the same time, we set up the server and started to work on API docs for the next phase’s implementation.

3. Detailed Design

3.1 Caregiver and Doctor Interaction

3.1.1 Functionality

3.3.1.1 Caregiver Dashboard

We designed the functions of caregiver tablet application based on their using scenarios from morning to evening.

Time	Function	Detail
Morning	Review patient's day	1. User can select the day he wants to see 2. User can see event title, description, start time, repeat frequency for each event
	Edit patient's day	1. User can add event to schedule by clicking add icon 2. User can edit events on a pop out window
	Review patient's sleep	1. User can review patient' sleep trend by clicking top right menu 'Jenn's trend', then clicking 'Jenn's sleep' 2. User can select the time duration he wants to see (choose the date) 3. Once user select the time duration, he can see the descriptive and number summary updated with the chart
Day	Monitor patient's location	User can see patient's location (in the format of map) on homepage. The map updates every minute.
	Check schedule	During the day, user can open the app and check the schedule of the patient anytime if they forget what to do next.
Evening	Review patient's activity	1. User can review patient' exercise trend by clicking top right menu 'Jenn's trend', then clicking 'Jenn's steps' 2. User can select the time duration he wants to see (past 1 week, past 2 weeks, past 3 weeks, past month) 3. Once user select the time duration, he can also see the summary updated with the chart

	Input incidents	1. User can click ‘Jenn’s Incidents’ in order to go to the incident page 2. On incident page: user can filter the incidents by time duration (Yesterday, Past week, Past month) and incident type (pre-set categories) 3. User can see number summary for each type of incidents, which is updated every 12 hours. 4. User can click the ‘Record Incident’ button on homepage or add icon on incident page to go to the editing page.

3.3.1.2 Doctor Dashboard

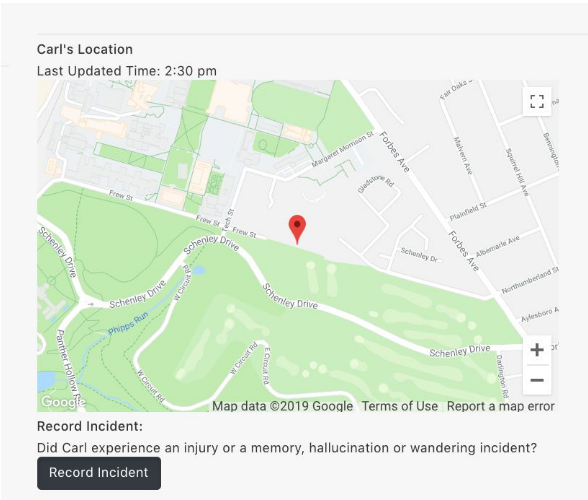
We designed several functions for the doctors according to their daily routines.

Function	Detail
See today’s appointments, select a day to see its appointments,	1. User can select what day to see and the selected date and day will be displayed 2. User can view active scheduled appointments (entered by administrative people) in a day including patient’s name and specific meeting time. 3. User can search the patient’s name or look up their names alphabetically. Patients who need extra attention are signaled with warning icon and listed just below the search bar
Select a patient and see critical patient trends	1. User can see patient’s demographic information and medication prescribed. 2. User can type in, save, and edit patient notes. 3. User can see next appointment time 4. User can see patients trends including: - Cognitive Assessment results - How often patients get up to use bathroom at night - Overnight patient awakenings / wanderings - Trend of caregiver reported patient injuries
See the complete log of incidents experienced by the patient	1. User can filter the incidents by start day and end day. 2. Each incident contains type, date, avg heart rate and caregiver notes

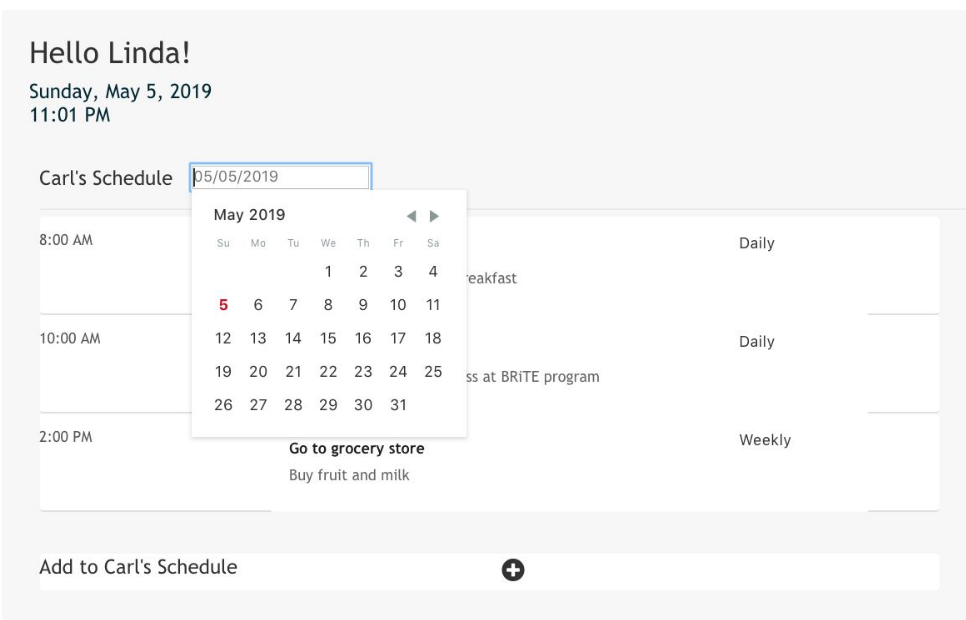
3.1.2 Interactions

3.1.2.1 Caregiver Dashboard Interactions

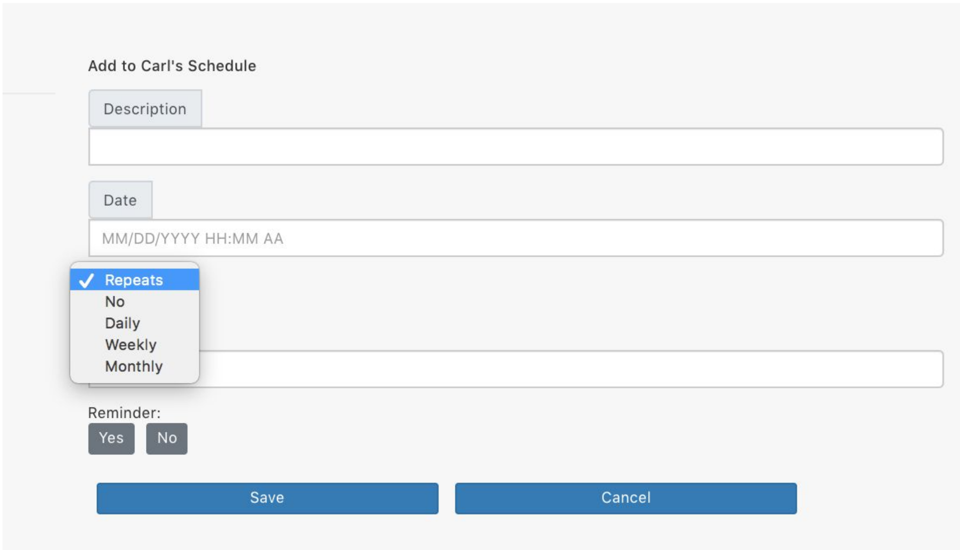
Caregiver can see the patient's real time location on home page where they can interact on the map as they are used to doing on google map. Right below the map is the quick entry button for adding incident. Caregiver will be directed to incident adding page if he clicked this button.



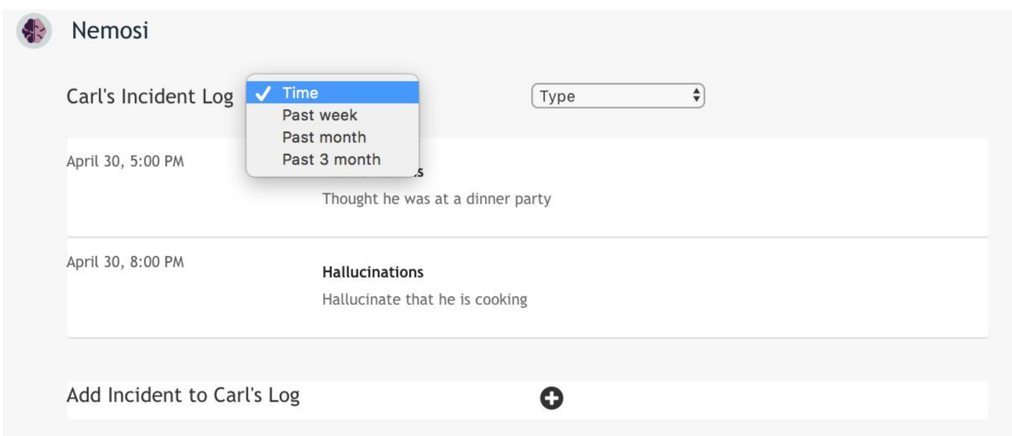
On home page, caregiver can see patient's schedule on specific day by date filter (The default date is today).

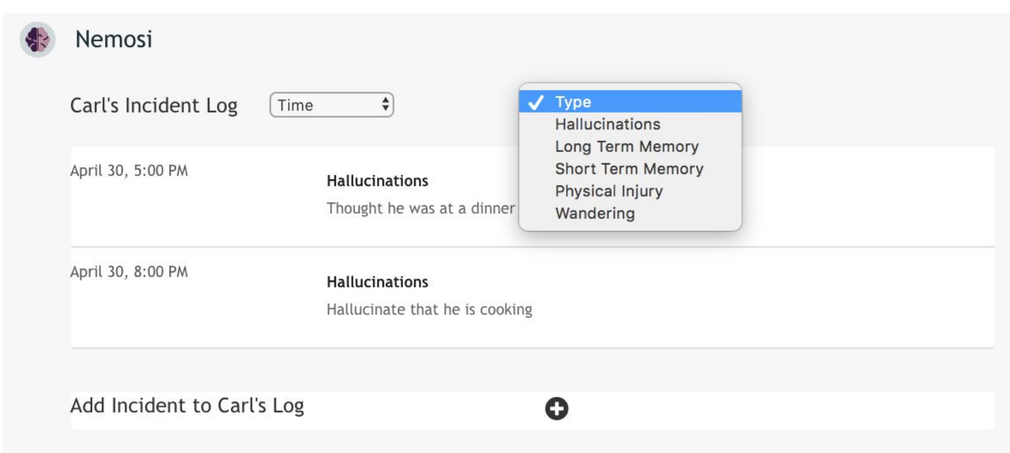


On the add schedule page, caregiver can set repeat frequency and reminder on/off so that they can save efforts on inputting same routine schedule.



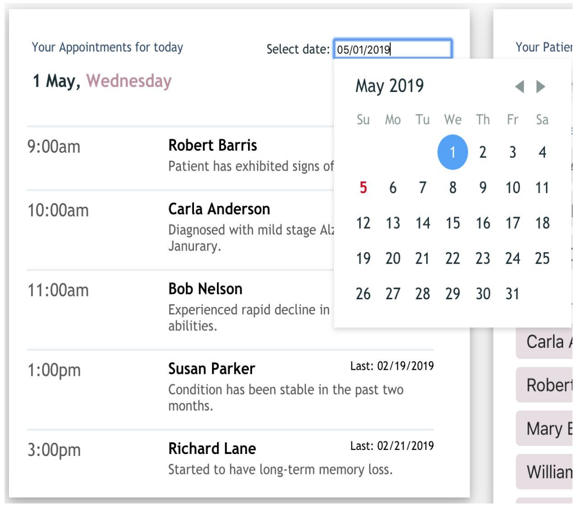
On incident page, caregiver can set the time and type filter in so that they can know one type of incidents during specific time period.



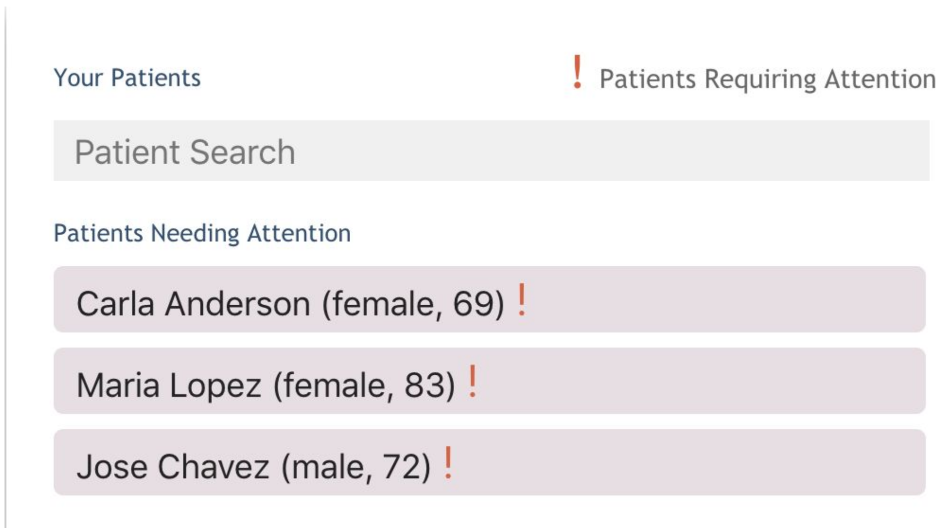


3.1.2.2 Doctor Dashboard Interactions

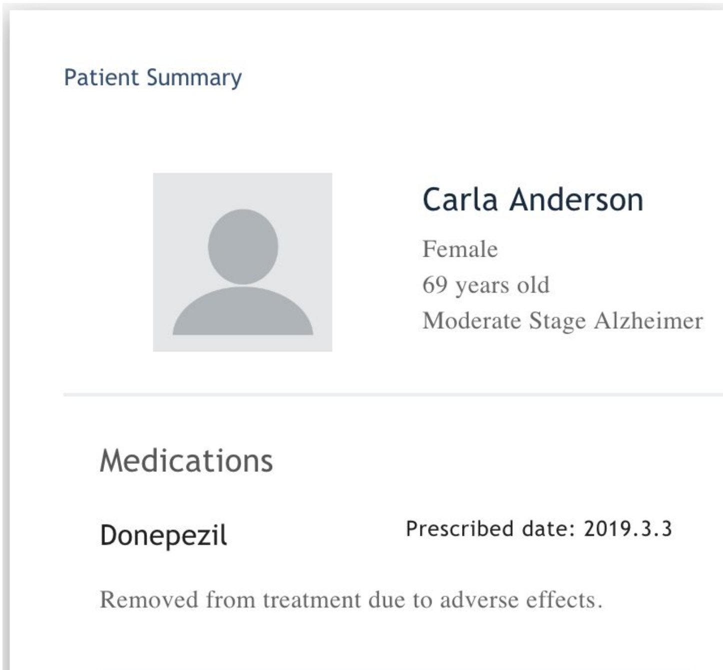
Our home page is where the doctor can check his or her appointments. On the left side, the doctor can see today's appointment, or the doctor can select another day to view that day's appointments.



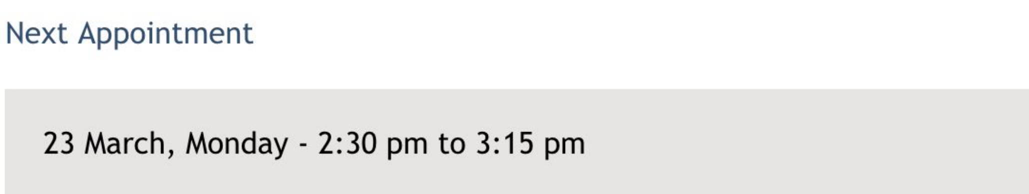
On the right part, the doctor can search the name of any patient. Also, patients that need extra care are listed above the name list.



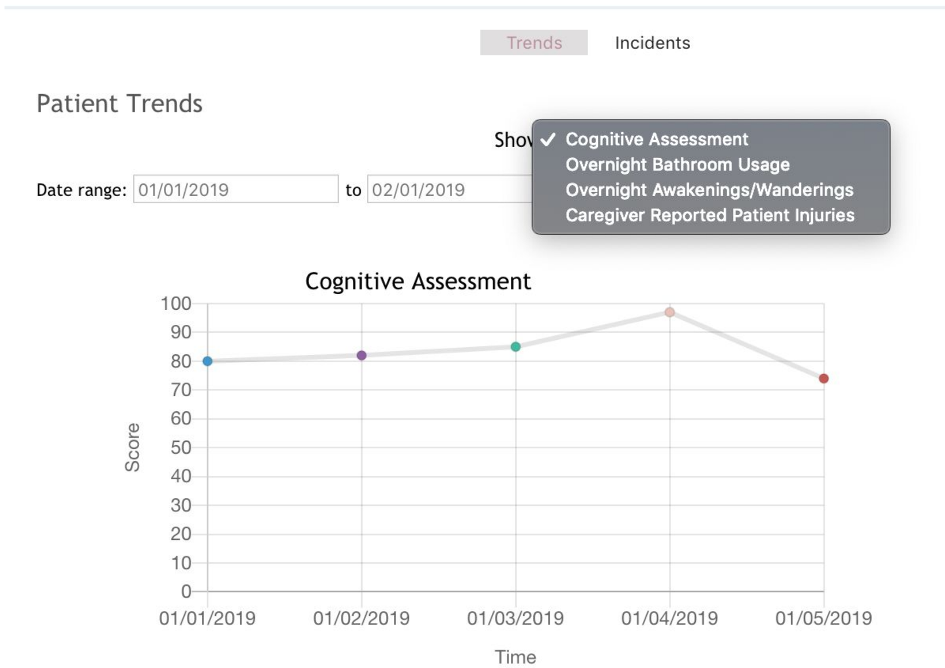
If the doctor then click on a patient's name, the patient's page will appear. Now on the left the doctor can see the patient's demographic information, prescribed medication.

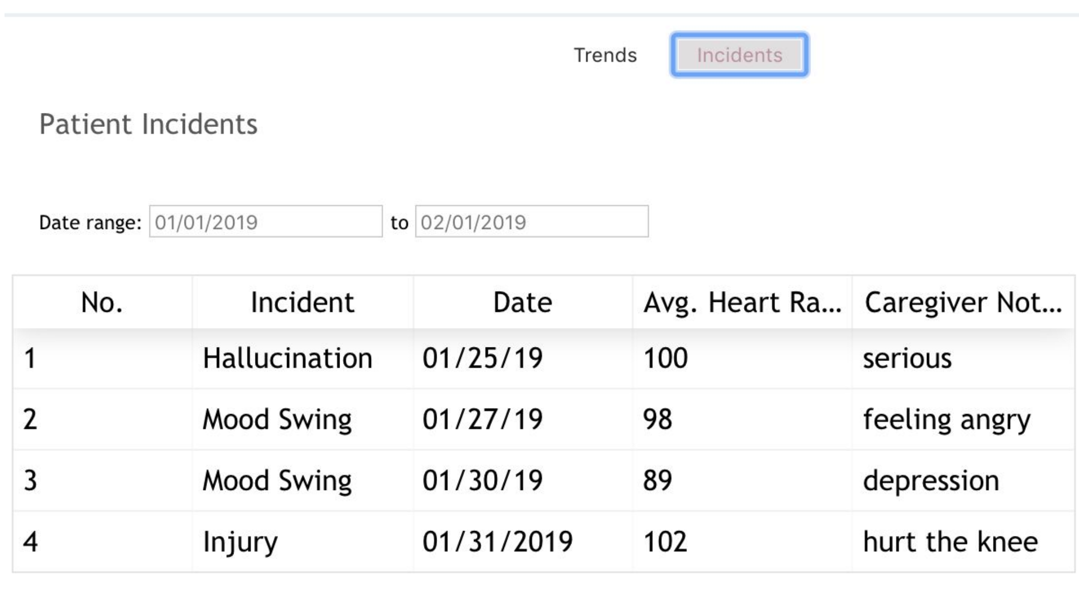


We also design a note box that allows the doctor to type in notes, and save or edit it. On the right side, first on the top can the doctor see when the next appointment will be.



Moreover, the doctor can switch between incidents and trends by clicking on the buttons and see the
detailed graphs or incident logs.





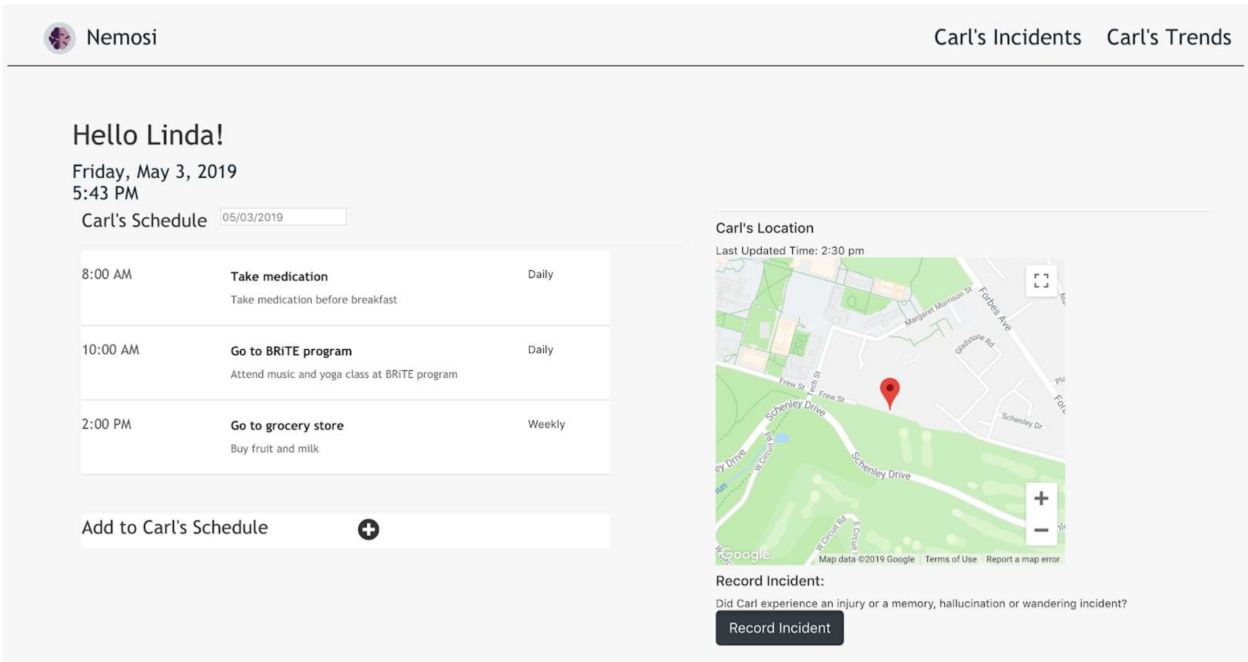
No.	Incident	Date	Avg. Heart Ra...	Caregiver Not...
1	Hallucination	01/25/19	100	serious
2	Mood Swing	01/27/19	98	feeling angry
3	Mood Swing	01/30/19	89	depression
4	Injury	01/31/2019	102	hurt the knee

3.1.3 Screens

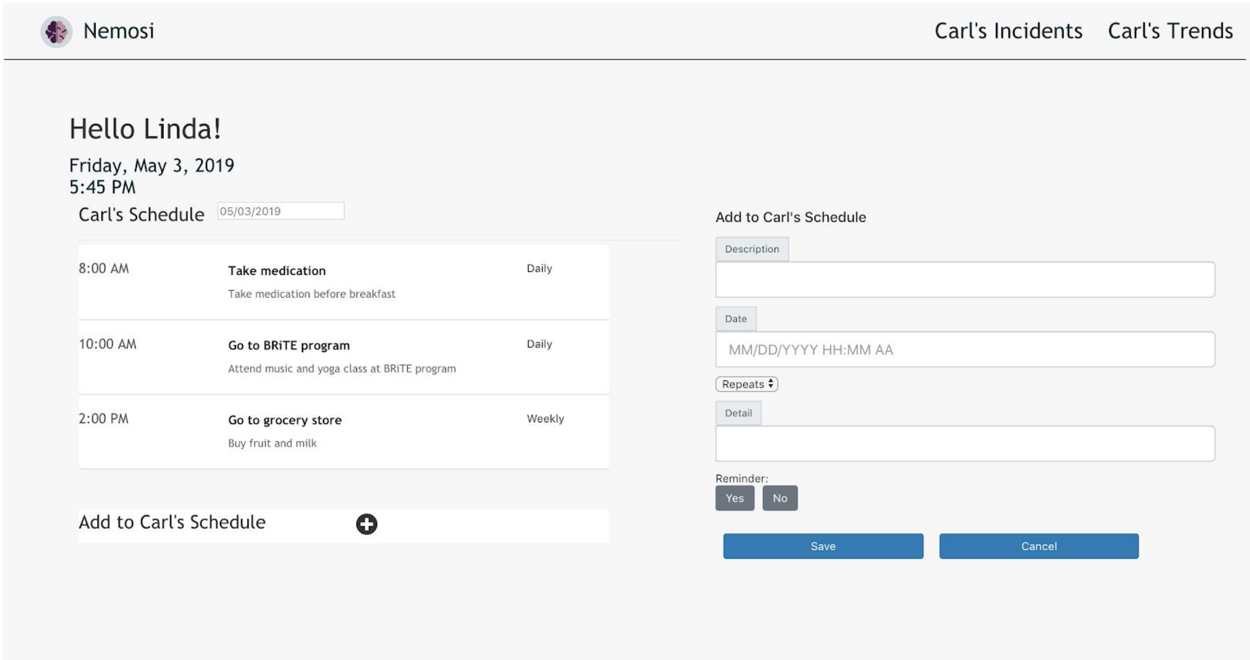
3.1.3.1 Caregiver Dashboard Screens

Caregiver’s dashboard is composed of three main pages: home page, incident log page and trend page.

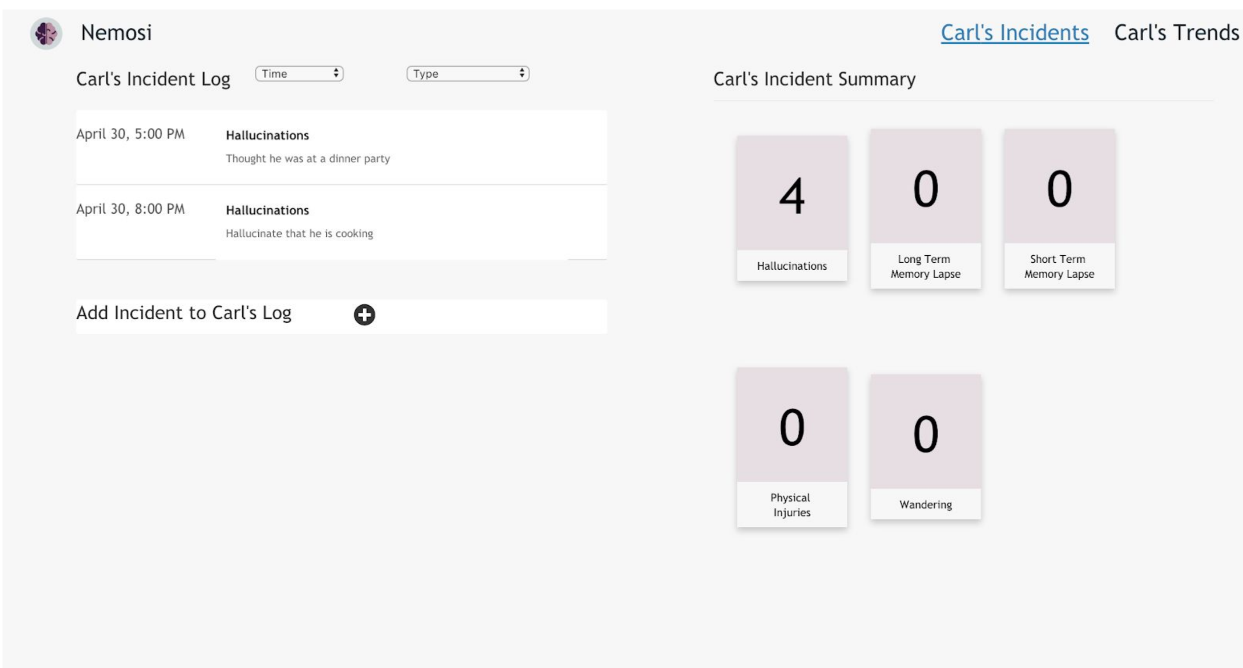
Homepage



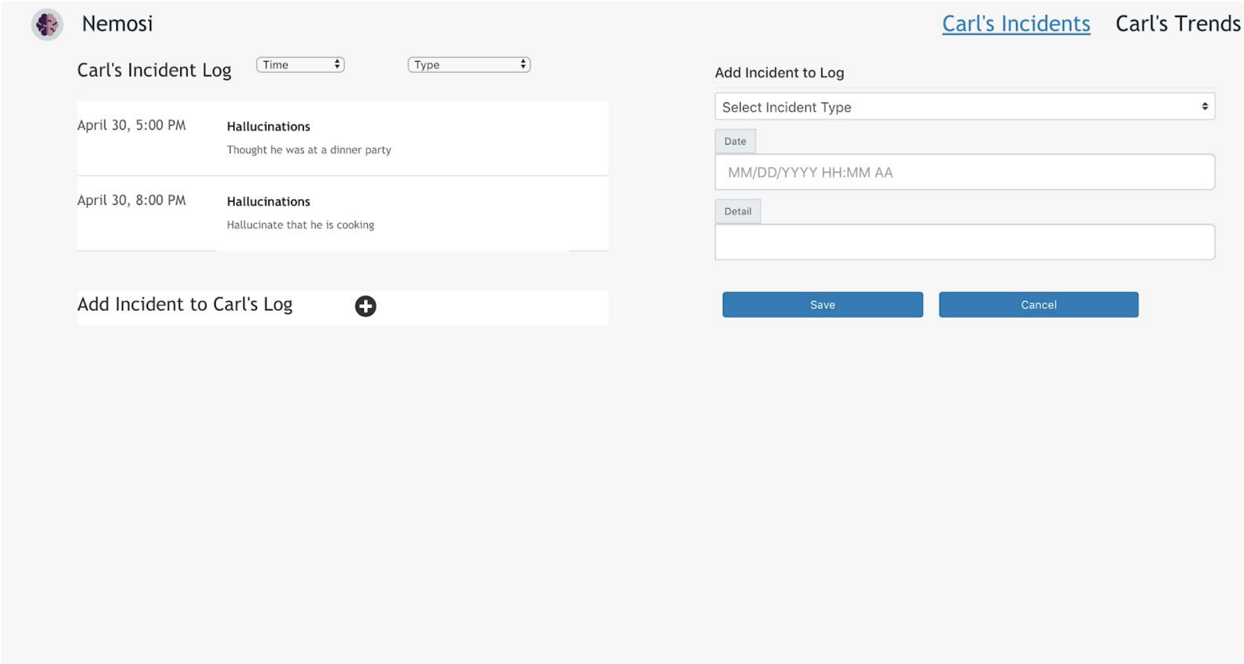
Home: Add schedule page



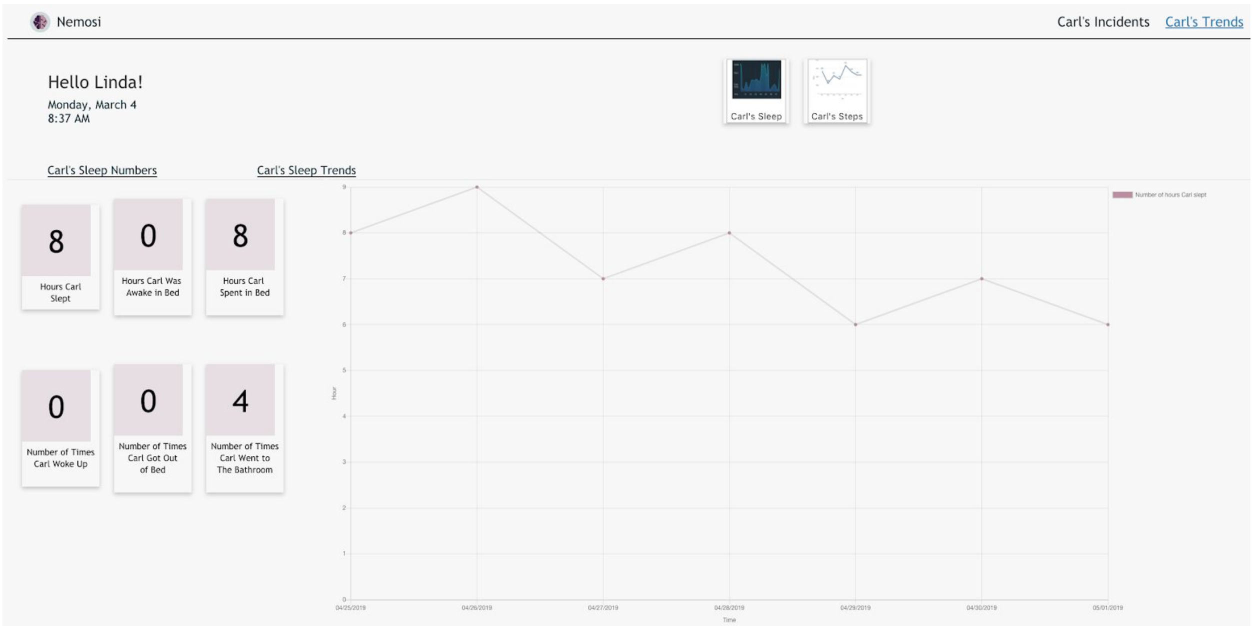
Incident page



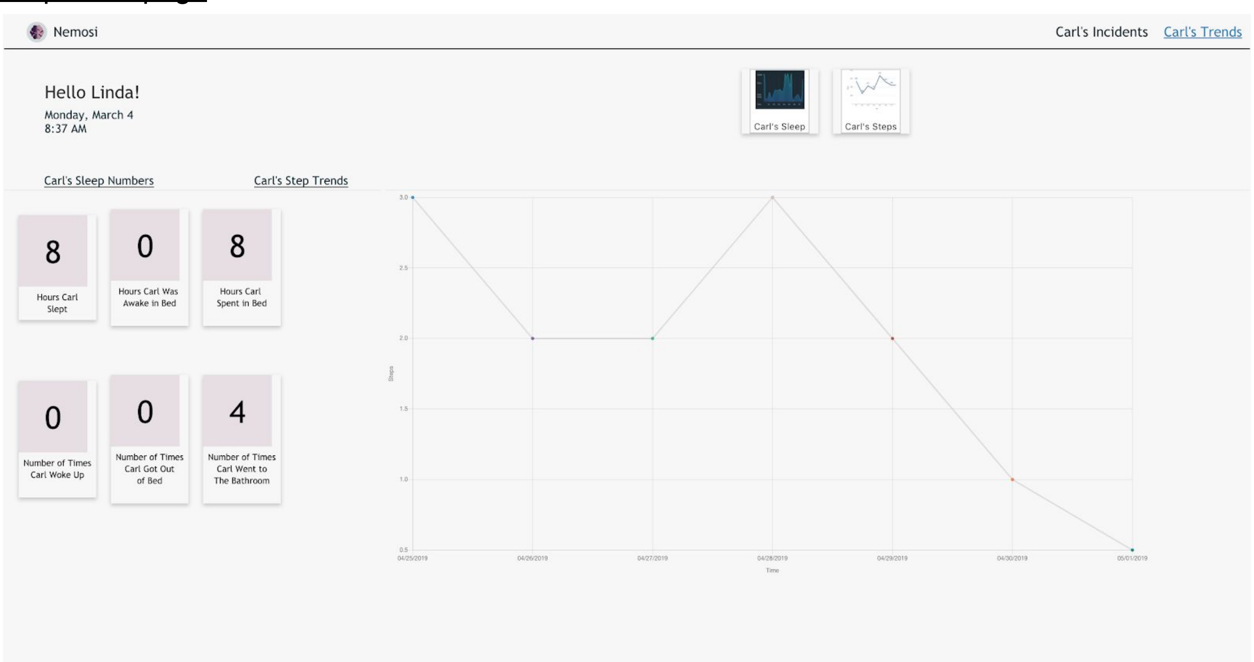
Add incident page


Nemosi
Carl's Incidents
Carl's Trends
Carl's Incident Log
Time
Type
April 30, 5:00 PM
Hallucinations
Thought he was at a dinner party
April 30, 8:00 PM
Hallucinations
Hallucinate that he is cooking
Add Incident to Carl's Log
Add Incident to Log
Select Incident Type
Date
MM/DD/YYYY HH:MM AA
Detail
Save
Cancel

Sleep trend page



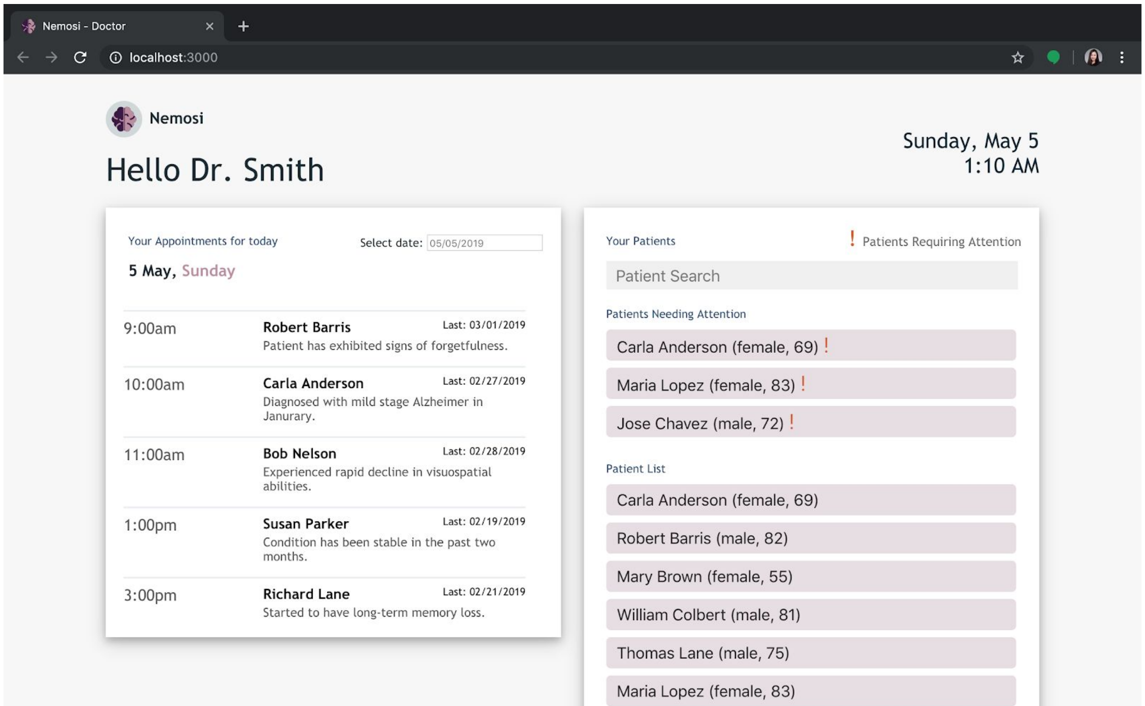
Step trend page



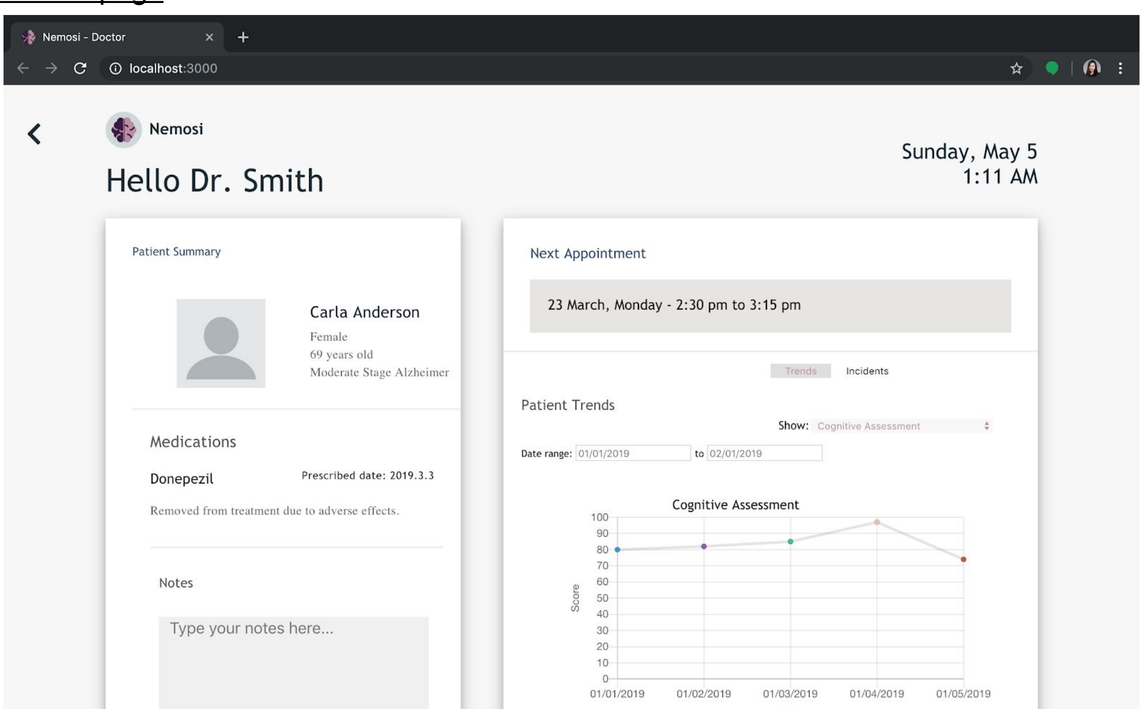
3.1.3.2 Doctor Dashboard Screens

We basically divided our web app into two main pages. One page is for the doctor to view all the patient lists, another is the page for individual patients.

<u>Home page</u>



<u>Patient page</u>



3.1.4 Experimental Evaluation

In the end of Phase II, we performed user testing on the flow of doctor and caregiver dashboards. Below are the feedback we received:

Doctor Dashboard:

1. The participants are unclear what's in the trends.
2. The names on homepage do not look clickable.
3. On the summary page, “incidents”, “recent incidents”, “trends” seem very confusing.
4. For “Views” the pop up page is not designed yet.
5. The “Notes” section is not clearly distinguished between doctor and caregiver.
6. What kind of alerts do the doctor want to see is unclear.
7. Maybe the recordings is useless.

Caregiver Dashboard:

1. User is confused by ‘End time’, how to rank the event on schedule? What if the time overlaps for two events?
2. what's in the droplist of the time filter?
3. What will happen if the user click reminder?
4. User is confused with difference between description and details
5. There is no date on the chart, so the user does not know which day the chart shows
6. There is no definition of ‘sleep’,’deep sleep’ and ‘awake’. So the user is confused by what it means
7. The X axis shows time from 01-07, the meaning is vague, not consistent with the time range below (past week)
8. User wants to go to home page, but he does not know where to find a home button. Finally, he thinks the brand name logo may be the home button.
9. Add to Jenn's schedule should be ‘add to Dave's schedule’
10. User is confused about the meaning of ‘movement’
11. User expects routes data rather than points on the chart (way of data visualization is not straightforward)
12. User thinks the summary is consistent with right chart, which means if the user chooses ‘past week’, he can see summaries like ‘average walking steps’, ‘meet the exercise goal or not’
13. The user remember there is a button on home page where he can record incidents (he does not go through the right top button)
14. On the incident input page: user wants to know how many photos he can add.
15. On the incident display page: user confused about what's in the dropdown list beside incident summary
16. User expects that he can click the name of incidents and see a pop out page like incident input page
17. User is confused about the meaning of ‘wandering’

Based on the feedback, we revised our design and developed our final implementation based on the revised design prototype. The revised screens are shown in the section above, 3.1.3 Screens.

3.1.5 Software Modules and Status

Module	Status
React	We used the React framework for the frontend development of both caregiver and doctor dashboards. It supports the basic functionalities we need for a responsive website, showing the components (buttons, charts, tables) and connecting with the real data from database. In the final demo session, we had both dashboards running in React development mode (doctor dashboard: localhost:3005/, caregiver dashboard: localhost:3000/).
Chart.js	Chart.js is a simple yet flexible Javascript library for creating charts. It is compatible with the React framework. In both doctor and caregiver dashboards, we use Chart.js to create the data visualization of patient trends (cognitive assessment, exercise, sleep trends, etc.).
React Table	React Table is a lightweight datagrid library for React. It provides us an easy way to render the patient incidents with other biometrics in a table format.
Day Picker API	Day Picker API is a flexible, highly customizable library for React applications. It is a UI component that allows us to quickly select a date or date range. This library is used in doctor dashboard for picking schedule, filtering date range for patient trends visualization and patient incident log.
Axios	Axios is a lightweight React library for data fetching from RESTful API endpoint. It allows for simple HTTP GET and POST requests with authentication headers. The caregiver dashboard uses Axios for data fetching.
Fetch API	Fetch API is a built-in Javascript library for data fetching from RESTful endpoints. The doctor dashboard uses Fetch API to fetch real time data from the database.

3.1.6 Alternative Evaluation Chart

In Phase III, we modified some of the functionalities in the dashboard because after research, the sensors team noticed some of the metrics specified in phase II design were difficult to retrieve. Therefore, we updated the features shown in the caregiver & doctor dashboard with the feasible data. Below is a list of features we updated in this phase.

Feature	**Original Data Used for Evaluation**	**Alternative Data Used for Evaluation**
Patient Exercise	Steps patient walks per day, recorded by smart watch	Distance patient walks per day. The location reported by wandering tracker every 5 seconds, the core analytics built a tool to calculate total distance from these location and timestamps.
Sleep quality	Hours of deep sleep vs light sleep per day	Total hours of sleep per day.
Hallucination	Detected by watch team according to abnormal sleep patterns	Incident reported by caregiver through caregiver dashboard, with date, time, details, and biometrics (heart rate, etc.) related to the timestamp.
Overnight awakenings	Detected by the smart watch	Detected by the sensor in pillow if the patient sits up in bed
Overnight wandering map	Heatmap of patient wandering frequency in each room during the night	Reduce to number of times patient goes to the bathroom, and number of times patient goes into the kitchen

3.2 Short Term Memory

3.2.1 Functionality

3.2.1.1 Assist Short Term Memory

One of our team goals throughout this project was to assist Alzheimer's patients dealing with deteriorating short term memory. In earlier phases, we found that patients are often forgetful and typically forget where they misplace items. We programmed functionality on the tablet app to interface with Tile (a tracking device) to allow patients to find their lost items. In addition to developing an android app, we interface with a Flic button which allows the patient to press a button to help locate the tablet in case it gets lost.

3.2.1.2 Measure Memory Decline

Alzheimer's patients lose their memory capabilities over time and it’s important for doctors to measure cognitive decline. We worked with Vista Life Sciences to add one of their memory diagnostic surveys to the tablet app. We also added some of our own elements to this test that we felt were important in measuring cognitive decline. We added functionality to save results of the test and and send these results to the database.

3.2.2 Interactions

During this project, the short term memory team worked with the following class groups: core tablet, games and assessments, and doctor patient interactions.

In regards to the core tablet team and games and assessments, we discussed GUI strategies to allow our android contribution to mesh nicely with their work. We shared common git repositories (Github and the class repository). From the start, our app was built in the same app space with close collaboration in general app structure.. This made merging code significantly easier than starting in separate projects.

With assistance from the database team, we added functionality in our daily assessment survey to send patient results to the database. We coordinated with the doctor patient interactions group to make our contribution to the class demo go smoothly.

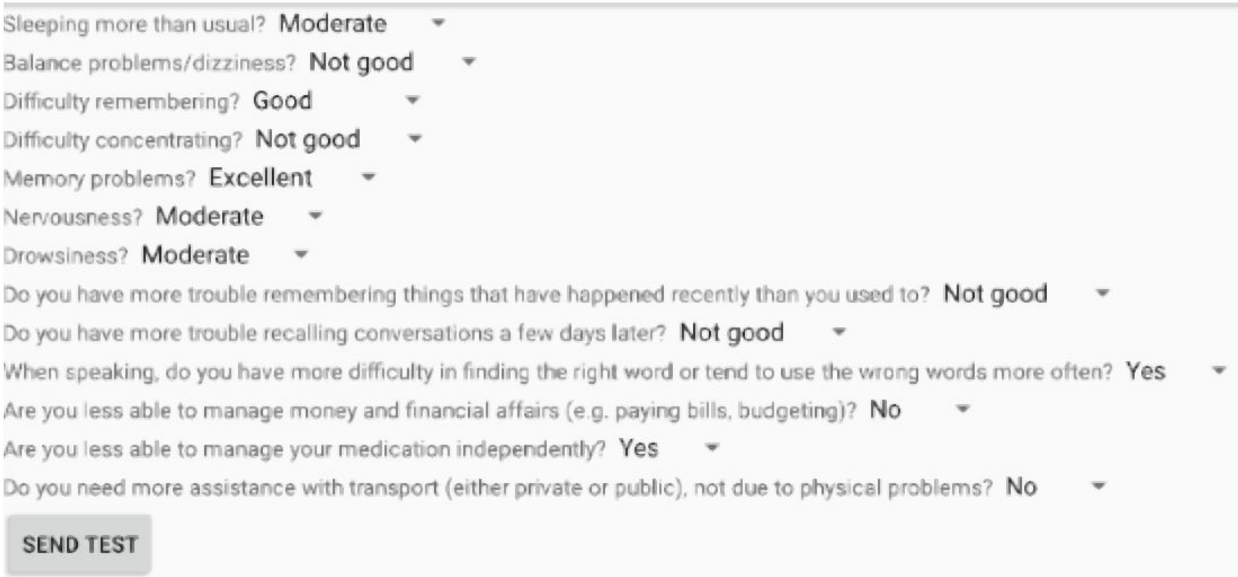
Individual members of the short term memory team collaboratively contributed code on the same github repository. Each member pushed code they each are responsible for different aspects of our team contribution.

3.2.3 Tablet Implementation

3.2.3.1 Daily assessment survey

We worked with Vista Life Sciences to get free beta access to their unreleased Alzheimer’s testing program. Their software has been clinically proven in the Alzheimer’s field. The top nine questions from our survey (screenshotted below) are from their ANAM diagnostic test. In these questions, the patient is prompted to select from options: not good, moderate, good, and excellent to answer Alzheimer’s related questions. We followed these questions by a few of our own that we found integral in measuring short term memory decline. The questions can be answered as yes or no.

After the patient fills out the survey, they can press the, “SEND TEST” button to send the results of the test to the database. The survey remembers user answers to the questions so that the patient does not need to reenter information if the answers do not change on a daily basis.


Sleeping more than usual? Moderate
Balance problems/dizziness? Not good
Difficulty remembering? Good
Difficulty concentrating? Not good
Memory problems? Excellent
Nervousness? Moderate
Drowsiness? Moderate
Do you have more trouble remembering things that have happened recently than you used to? Not good
Do you have more trouble recalling conversations a few days later? Not good
When speaking, do you have more difficulty in finding the right word or tend to use the wrong words more often? Yes
Are you less able to manage money and financial affairs (e.g. paying bills, budgeting)? No
Are you less able to manage your medication independently? Yes
Do you need more assistance with transport (either private or public), not due to physical problems? No
SEND TEST

3.2.3.2 Find lost items

For our tablet app, the patient is able to use the Tile tracker system. Shown below, the patient is able to attach Tile trackers to various devices they frequently lose. We recommended Tile trackers to be attached to the patient’s house keys, TV remote, and the charger for their phone. When the patient loses one of their tracked items, they go to the tablet app and then click on a button to find whatever they are missing. The Tile tracker will then beep and the patient will be able to locate the item by audio.

3.2.3.3 Find lost tablet

In a rare case, the patient could lose their tablet and then not be able to access the Nemosi app. This situation could be incredibly frustrating so we implemented a system to help find a lost patient tablet. We use Flic buttons, shown below, to solve this issue. The Flic button has an adhesive underside which would be used to stick the button to a surface. When the patient presses the Flic button, it would connect to the tablet over bluetooth and cause it to ring. Similarly to the Tile trackers, the patient would then be able to follow the tablet auditorily.

3.2.4 Experimental Evaluation

All of our features were thoroughly tested by each group member. Because we wrote code to implement or interface with proven ideas and products, there was very little need for extensive experimental evaluation. Our daily test was already proven by Vista Life Sciences to be effective in the alzheimer’s field. We directly implemented their proven tests. In regards to the Tile tracker and Flic button, we did not design these products, instead, we added android code to interface with them. We tested our code and these products have been used heavily by consumers for years. We are confident in our techniques to tackle short term memory in the alzheimer’s field.

3.2.5 Software Architecture

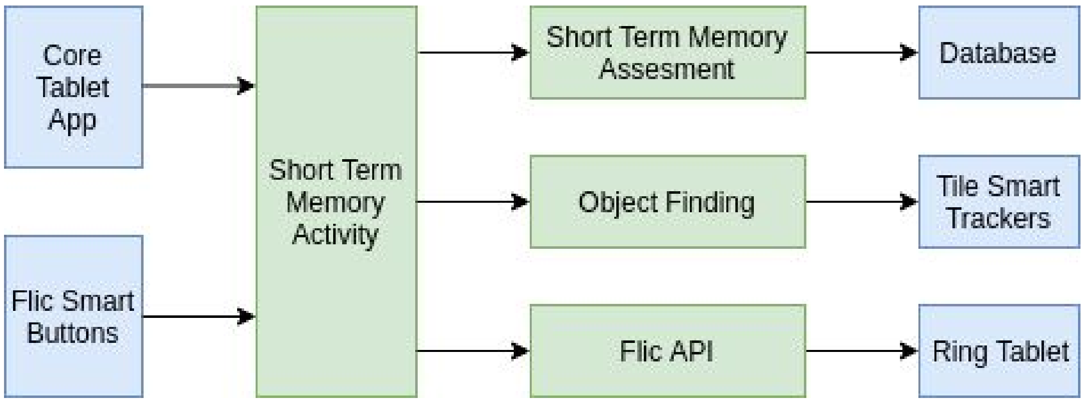


Figure 4.5.5.1 Software Architecture Diagram for Short Term Memory

The final software architecture for Short Term Memory has been revised since the iteration in Phase 2. The Short Term Memory component is now part of the overall patient Android application from the Core Tablet Team, and the daily assessment survey now sends the scores directly to the Backend Server / Database through HTTP POST requests to the backend's REST API. The control flow of the Short Term Memory component is now very straightforward and can be divided into three parts:

Daily assessment survey:

1. Patient opens the Core Tablet App.
2. Patient selects Daily Assessment Survey
3. Patient completes the survey
4. Patient hits "SEND TEST" button and the survey data is sent to the database

Find lost items:

1. Patient opens the Core Tablet App.
2. Patients selected the "Find lost items" button
3. Patient then chooses which missing item they want to locate
4. Tile tracker attached to missing item will beep
5. Patient locates lost item auditorily

Find lost tablet:

1. Patient presses Flic button
2. Tablet rings
3. Patient finds tablet auditorily

3.2.6 Software Modules and Status

The short term memory component is divided into three functional features: daily assessment survey, find lost items, and find lost tablet. These features were created to help Alzheimer's patients deal with loss of short term memory as well as allowing doctors to test short term memory in relation to Alzheimer's progression. Our features are fully implemented in the android app.

3.3 CORE ANALYTICS

3.3.1 Functionality

Functionality was set so that caregiver and interaction team could be able to request and retrieve the relevant data analytics that would be important for them to display, and real-time text notification could be sent to caregivers when an incident occurs to the patient. The script was designed so that it would be able to output all analytics and plots for a previous day or specified timeframe.

3.3.1.1 Data analytics

Raw data is analyzed in two ways: real-time and periodically (every 24 hours). Some data on the dashboard needs to be updated in real-time, such as number of times the patient wakes up overnight, heart rate & pulse rate when an incident occurs. Besides, data from sensors need to be analyzed in real-time so that we could identify an incident and send notification to the caregiver immediately. On the other hand, some data could be updated every 24 hours for analytical results over a time span e.g. average value in a day, week or month.

3.3.1.2 Real-time notification

Caregivers will receive text notification instantly when an incident occurs to the patient, and we identified the following incidents.

- patient falls
- patient uses the stove abnormally
- patient exits the main door and wanders
- patient gets off bed and went to bathroom overnight
- patient stays in the bathroom/kitchen for too long

The detection methods for these incidents will be discussed in detail in the following sections.

3.3.1.3 Data Visualization

Data visualization allows caregivers check patients' *Sleep Trend* and *Steps Trend* through Caregiver Dashboard, and allows doctors check patients' *Cognitive*

Assessment, *Overnight Bathroom Usage*, *Overnight Awakenings*, *Overnight Wanderings* through Doctor Dashboard. Caregivers and doctors can select the trend they want to see on the interface. They can also selection a date range for each trend. Detailed graphs are shown in Section 4.2.2.1.

3.3.2 Test and pictures

3.3.2.1 Data Visualization

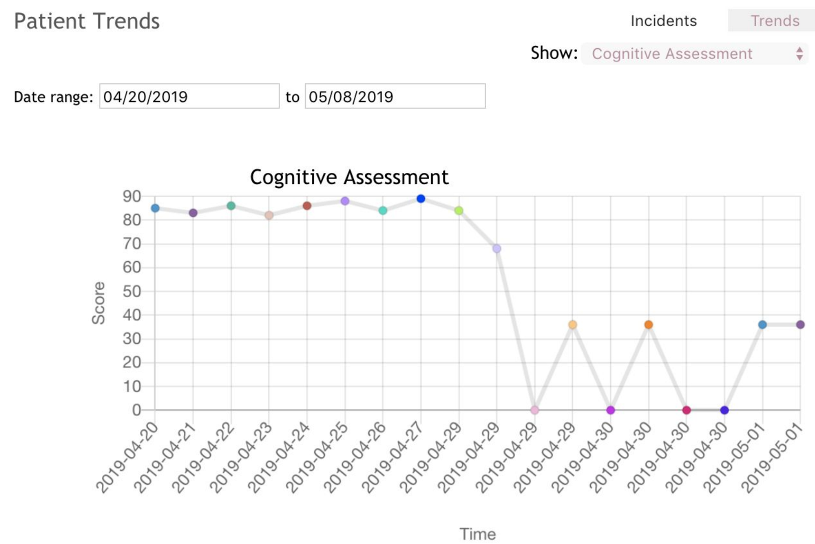


Figure 3.3.0 Doctor Dashboard Cognitive Assessment

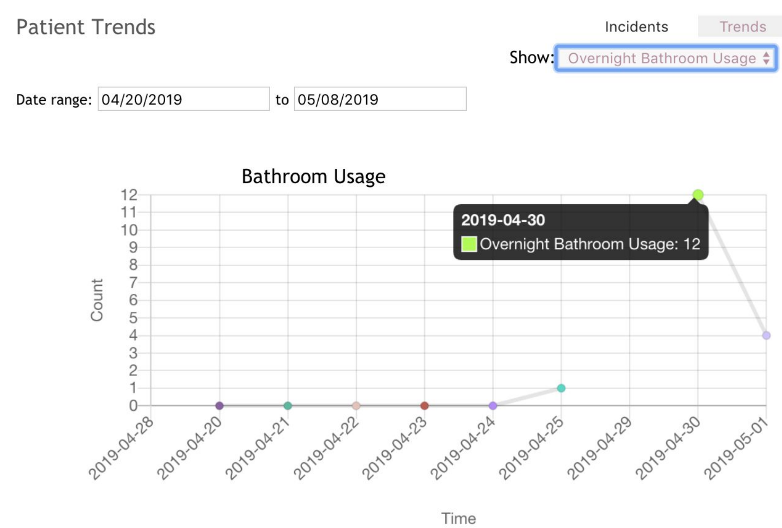


Figure 3.3.1 Doctor Dashboard Overnight Bathroom Usage

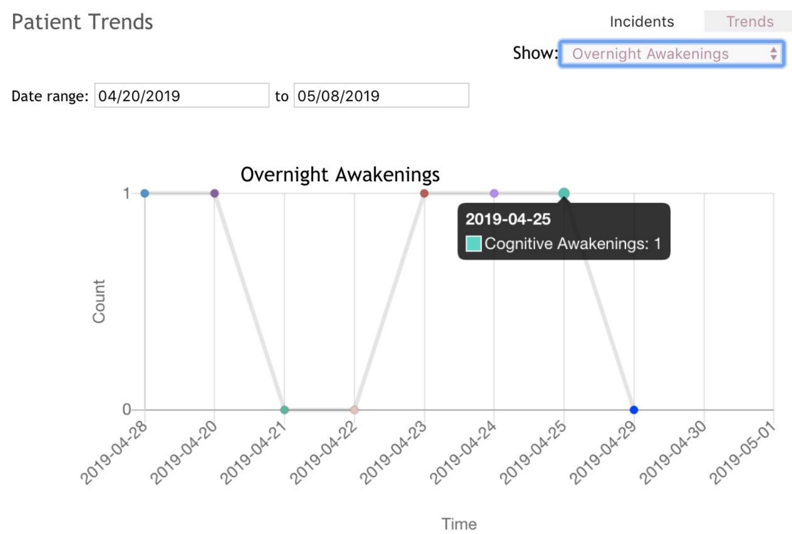


Figure 3.3.2 Doctor Dashboard Overnight Awakenings

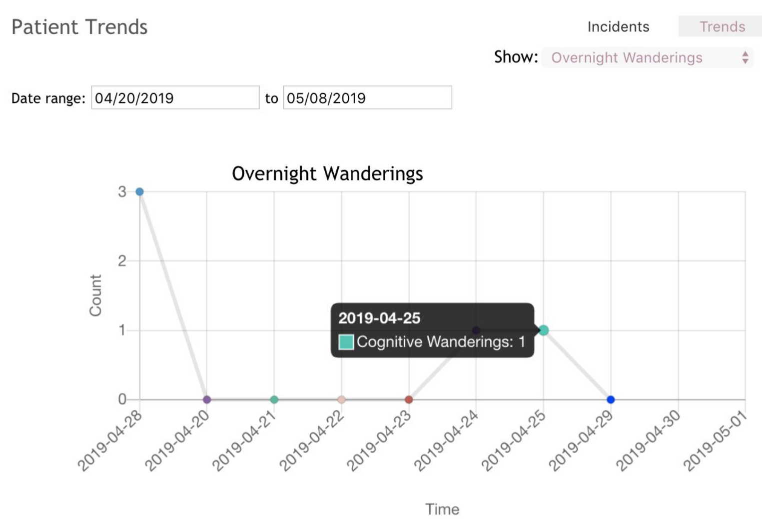


Figure 3.3.3 Doctor Dashboard Overnight Wanderings

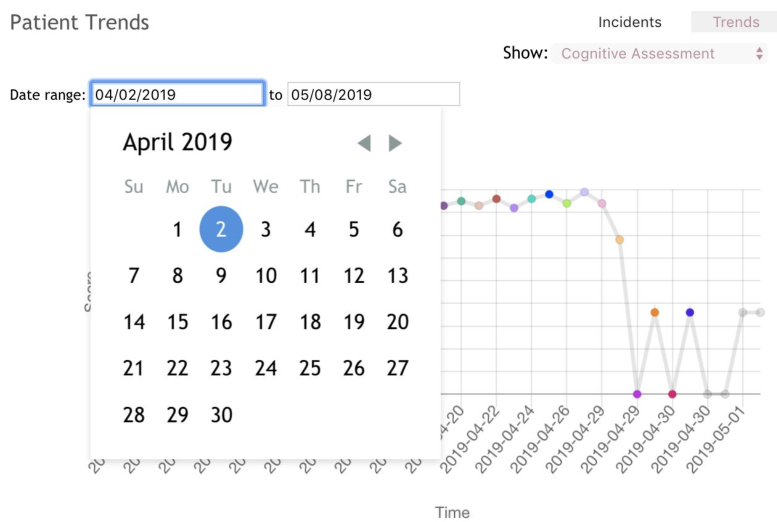


Figure 3.3.4 Doctor Dashboard Date Range Selection

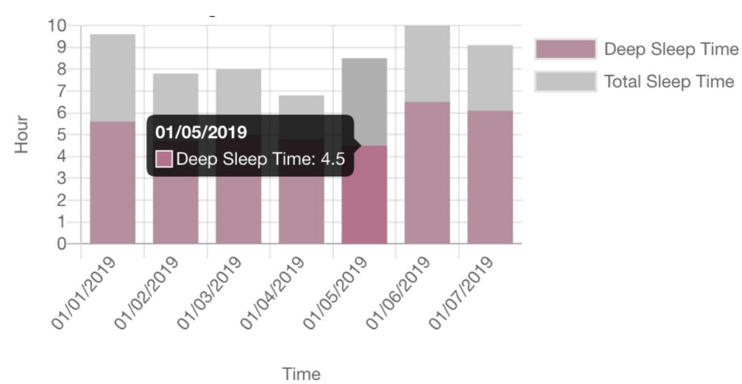


Figure 3.3.5 Patient Dashboard Sleep Trend

3.3.2.2 Notification

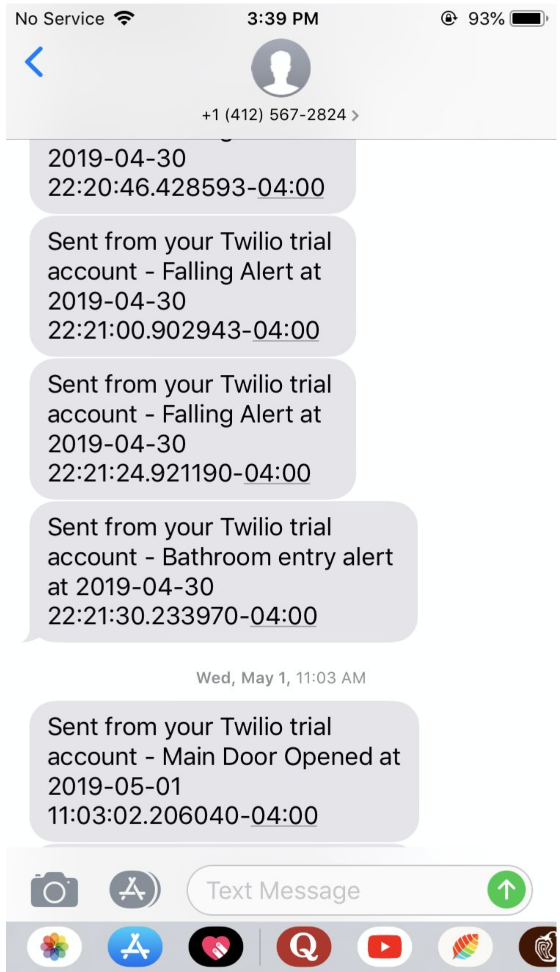


Figure 3.3.6 Data Table

Data Name	**Data Type**	**Description**
patient_id	IntegerField	Patient's unique identifier, default=0
date	DateField	Date of this data entry
hours_slept	FloatField	Hours the patient slept
hours_in_bed	FloatField	Hours the patient spent in bed
num_wake_up	IntegerField	Number of times the patient woke up overnight
num_get_our_of_bed	IntegerField	Number of times the patient got off bed overnight
num_go_to_bathroom	IntegerField	Number of times the patient used bathroom overnight

Table 3.3.0 Data table for sleep trend

Data Name	Data Type	Description
patient_id	IntegerField	Patient's unique identifier, default=0
date	DateField	Date of this data entry
num_ltm_lapse	IntegerField	Number of long term memory lapse
num_stm_lapse	IntegerField	Number of long short memory lapse
num_falls	IntegerField	Number of falling incident
num_wandering	IntegerField	Number of wandering incident
walk_distanece	FloatField	Walking distance

Table 3.3.1 Data table for incident summary

3.3.3 Experimental Evaluation

3.3.3.1 Patient incident analytics

Falling

This is the important function since falling is sometimes the serious incident for patient. Thus, when patient falling, we will have the real-time alert directly send to the caregiver to help the patient in time.

To implement this function, first, we have to catch the falling incident when it happen. The watch team will update the new falling incident in their database event table, while the sensor wore by patient monitor the patient is falling. After they update their table, we will fetch the row that event_category is "falling" and add one to the num_of_falling column for the patient in core analysis' incident summary table. Otherwise, if the patient id cannot find in the incident_summary table, we would insert the new row into the table and put one into the nums_of_fall column. The implementation is in the function of watch_analysis and update_num_of_falls.

Abnormal use of stove

When we analyze data about the stove temperature, we need to divide it into two parts--day and night. The stove sensor sent the data to the server, the data contain the timestamp, stove condition and 64 data about stove temperature. Firstly, we need to judge the time with the timestamp, if the time range is between 0am to 7am, we think it is night time data. Then we need to look the information about stove condition. The information about stove condition includes hot, warm and unknown. If it is hot or warm, and it is during the night, we need to send the notification to the caregiver. During the night, if the stove condition is unknown, we need to take the maximum value in 64 digits. If the maximum value if higher than the threshold, we also need to send the notification to the caregiver. After analyzing the data, if we find the data is during the day time, we need to add the movement sensor. If the movement sensor shows that the patient does not move, and the stove data is higher than the threshold, we need to send the notification to the caregiver.

Wandering

There is a tracker sensor on the patient's shoes, and it will keep updating the patient's location to the table wt_patient. We also set a safezone to the table wt_safezone, and the location of caregiver will be updated from caregiver tablet to the table wt_caregiver. We can calculate the distance between the distance between caregiver and the patient using the data in the database wt_patient and wt_caregiver. We also use the data from wt_safezone and the patient location data to make sure if the patient is in the safezone. Besides, watch team will update if the patient is on wandering status to the database. Thus, A caregiver will be notified by text message if the patient is wandering. Notification serves to provide safety to the patient when the caregiver is not around.

Sleep incident

Sleep incidents include patient waking up overnight, getting off bed, and using the bathroom, and involve real-time analysis, notification and database update.

To calculate how many times the patient wake up and get off the bed at night can help us to know how the sleeping trend the patient has. There is a pressure sensor on the patient's pillow, and the sensor will send data to the table hs_events. If the patient leaves the pillow, the pressure sensor on the pillow, the name in the database is “pillow”, will send “HEAD_NOT_DETECTED” message every 10 seconds and store in the database. Thus, when we pull the data from the database. We will count as the same event if the data less than 10 seconds. As soon as the leave-pillow count update, we will immediately update the num_wake_up field of the table ca_sleep_trend. After that, when the doctor's and the caregiver's dashboard will extract the calculated wakeup number and visualize it.

About the get off the bed calculation, we also have a pressure sensor on the mat. If the patient leaves the bed, and steps on the mat. The pressure sensor on the mat, the name in the database is "pressure_mat", will send " FEET_DETECTED" message also every 10 seconds and store in the same hs_events database as pillow sensor database. When we poll the data from the database, we will count as the same detection if the data less than 10 seconds. However, the idea is a little different from the pillow detect data. Because we count a get of bed event from two detections on the mat. One is leaving the bed and the other is coming back. As soon as we detect two detections, we will set the num_get_of_bed field of table ca_sleep_trend in time. Moreover, to ensure the security of the patient, we will keep tracing the coming back time. If we didn't detect the coming back data for more than half an hour, we will send notification to the caregiver.

Overnight bathroom usage is also considered a metric for sleep quality evaluation. Raw data is retrieved from home sensors, particularly rfid sensor and pir sensor. Rfid sensor is installed on the door of bathroom to detect activity around the door, whereas pir sensor is installed under the sink to detect motion in the bathroom. Only when rfid sensor posts 'check entry' data and pir sensor posts 'motion detected' within 10 seconds, the system will identify a bathroom entry. Once an entry is identified, the system will not analyze data in 3 minutes to avoid repetitive text notifications for one incident.

3.3.3.2 Data processing

Walking distance calculation

To record the distance that patient has walked in a day. We will update the walking distance entity in our core analysis table every 24 hours. Our input data is given from wandering team's patient table. In patient table, the entity including patient id and the location of patient. They will update the patient location every 2 mins. Thus, our calculate method of walking distance is to collect the different locations of the patient in a day, then sum the difference between each location. The function Wt_distance_analysis including extracting location method and the distance calculation. To calculate the distance between different longitude and latitude precisely, we have imported the geopy library. Additionally, the update period of the walking distance is 24 hours (a day). Thus, we can get the new update of the walking distance in core analysis' incident summary table every 24 hours.

Data synchronization

In this part, we implement the functionality of data synchronization for caregiver. It could be helpful and convenient for caregiver directly get the patient's detail health information in there dashboard without going to check the health info from the watch or sensor.

When the caregiver enter the new incident in caregiver's incident table and upload to the database, we would immediately trigger the database system to insert pulse rate and respiratory rate in the new incident row in the incident table, which is the real-time update mechanism. The data synchronization for caregiver has been implemented in the function of fetch_watch_rate. We collect the patient's pulse rate and respiratory rate from the watch team's event table and automatically insert it into caregiver's table.

3.3.3.3 Real-time notification

Text messages are considered more "real-time" than emails. While emails can being checked manually by the receiver regularly, text messages would always pop-up on the receiver's cell-phone immediately upon sending, cell-phone being an personal belonging also make the important messages more reachable. Texting however, does come at a cost compare to emails, but the pricing of most text services uses the pay as you go model, and we only use texts for emergencies, it is not very frequently expected and makes the cost negligible. Therefore, for all the emergency events, we have integrated real-time text messaging notifications to the caregiving instead of emails.

To enable the server to send texts, a SMS gateway must be used. After exploring different options including red oxygen, AWS, Twilio and several popular ones, we have selected Twilio due to its low pricing and easy to use API. It also comes with a bonus of 2000 messages of free-trial which made our testing a lot easier. The API for Twilio is very easy to use. We have selected to use the simplest one for our application, each HTTP post request with the receiver's phone number and content is transferred to the gateway and sent to the user. The Twilio's API is also future proof with 2 factor authentication and phone verification functions available. It can also support text reply, and this would be a nice feature to have for our system in the future, for example, the caregiver could reply to an emergency text to initiate some follow up actions to be taken on the patient, such as calling 911 and report patient location automatically on the server.

3.3.4 Software Architecture

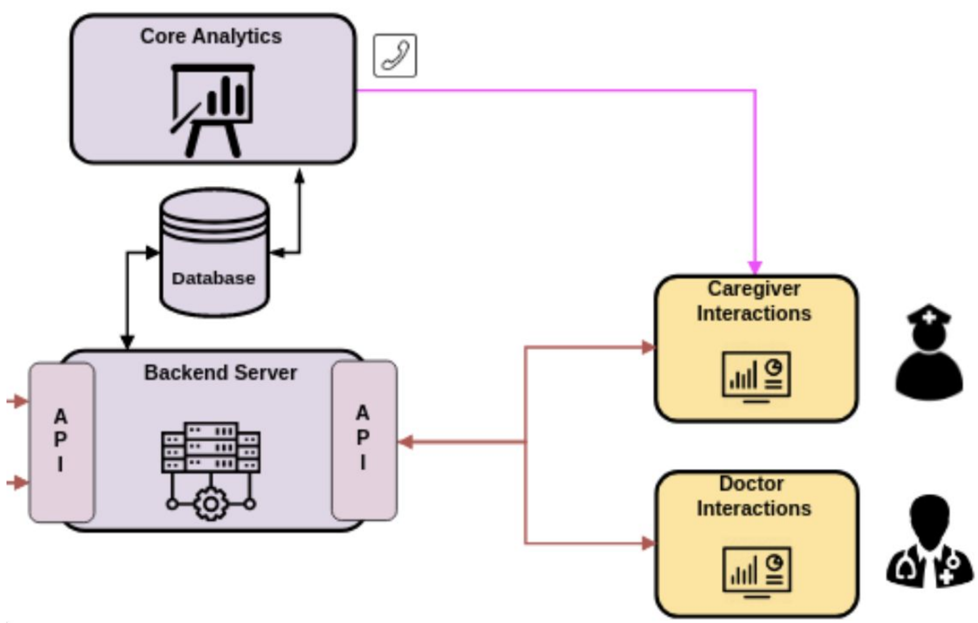


Figure 3.3.7 Software Architecture

3.3.5 Software Modules and Status

Modules	Status
Data Acquisition	Finished
Data Analysis	Finished
Data Visualization	Finished
Text Notification	Finished

Table 3.3.2 Software Modules and Status

3.4 HOME SENSORS

3.4.1 Functionality

3.4.1.1 Kitchen Monitoring

We connect each of the sensors to an Adafruit Huzzah32 controller. The controllers get readings from the sensor and are connected to the Internet via Wifi. The stove sensor, or the GridEye infrared camera, can be mounted above the stove. The sensor is able to monitor the entire stove area. It sends an event when the temperature of any pixel is higher than a preset threshold. It also updates the infrared image every ten seconds for users to monitor the stove in near real time. The motion (PIR) sensor detects any motions in the kitchen area and sends the event to the server. The sensor controllers connect to the database server using an HTTP connection. Data is sent in a JSON format.

When an event is detected by the server, algorithm implemented by the Core Analytics team will analyze the events send by both sensors and decide if it is critical enough that the system should notify the caregiver.

3.4.1.2 Restroom and Main Door Monitoring

Main Door:

Two sensors were used for the detecting if the patient has left the main door. In order to do so the VCNL4010 Proximity sensor and the RFID reader were used in conjunction. Both the sensors were integrated to the same Huzzah featherboard given that they would be installed at the same location.
The algorithm was designed for three different cases namely, when the proximity and the RFID sensor would work together, only the RFID would work and only when the proximity sensor would work. The detection activity from both the sensors are taken in account to accordingly alert the caregiver and generate the log that the patient left the main door.

Restroom:

The detection of the patient in the restroom is dependent on both the PIR motion sensor and the RFID sensor as discussed from Phase 2. Accordingly both the sensors were made to work individually and accordingly send data to the server.
The RFID reader and the PIR sensor were both connected on two different Adafruit Huzzah boards. The algorithm analyzed the messages received from both these feather boards. From successful prompts sent by the sensor the patient was appropriately detected in the restroom. The alert was thereon generated for the caregiver and the activity was logged.

3.4.1.3 Sleep Monitoring

Sleep detection and monitoring was done in 2 parts: monitoring the presence of the patient's head on the pillow and detecting the presence of the patient's feet on the bedside mat. Both the pillow and the mat contained a pressure sensitive resistive sheet that would be triggered when pressure is applied. The data from both the devices was sent to the backend server and the core analytics team combined the data from both servers to make inferences.

3.4.2 Tests and Pictures

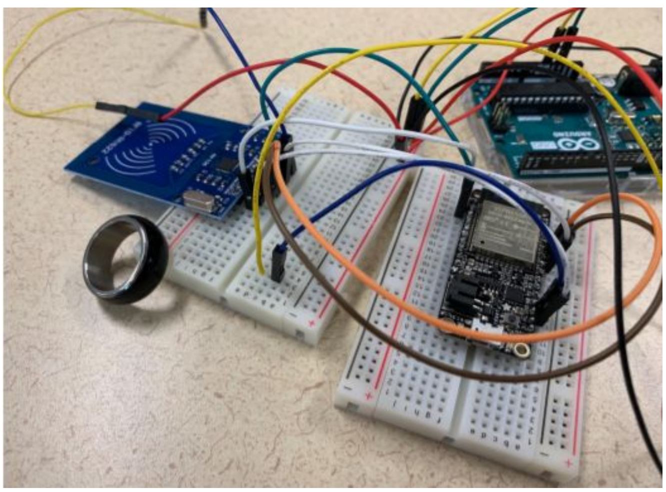
Main Door:

Steps:

- As soon as the patient opens the main door, the RFID reader on the door frame detects the RFID ring in the hand of the patient which confirms that it's the patient who is going out.

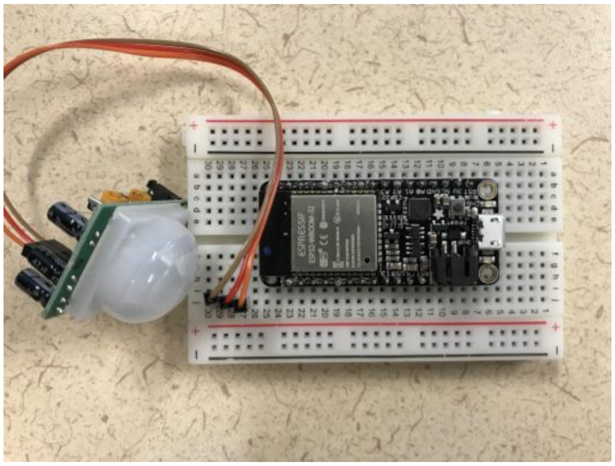
- Once the door is opened, the patient exits through the door, the proximity sensor on the frame of the door detects the exit.
- Signal from both these sensors consecutively confirms that the patient has exited the main door.
- The signal then goes to the server over the Wifi using the wifi module in the Adafruit Feather board.
- The server then sends a text alert to the caregiver on his/her mobile phone.

Steps to install the system:
- Make sure that your Wifi is always on, especially during the night time.
- Install the sensor modules on the frame of the main door and connect them to a power outlet.
- Make sure that the patient wears the RFID ring all the time.
- Keep checking the text alerts you receive on your mobile phone.



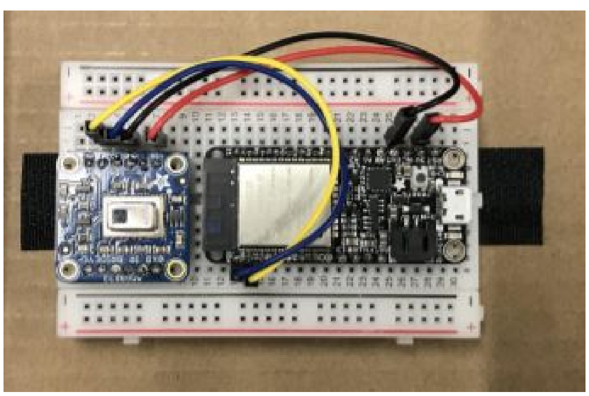
Restroom:

- Patient enters the restroom
- Install the RFID reader at the frame of the restroom door
- The RFID ring records the patient being near the door and sends the detection to the server
- Install the PIR motion sensor inside the restroom (maybe somewhere near the wash basin)
- The PIR motion sensor detects the motion of the patient and sends ‘Motion detected’ to the server
- The analytics team processes both these messages and generates the alert to the caregiver
- You will receive the alert on your caregiver dashboard.



Kitchen:

When the patient enters the kitchen the motion sensors attached on the frame of the door of the kitchen senses some activity and sends a “Motion detected” message to the server. In addition, if the patient tries to switch on the stove, the heat from the stove would be detected by the IR thermal camera which would be setup on the ceiling just above the stove. As soon as the temperature sensed by the stove sensor is above a threshold a message is sent to the server- “Stove is switched on”. Only after both these messages are received in the server does the caregiver be sent an alert notification-”Stove is switched ON without supervision”. This is to avoid any fire related mishap in the kitchen.



Bedside Mat:

The bedside mat needs to be placed at the side of the bed where the patient lies.

It should be switched on by the caregiver once the patient is on bed lying down.

When the patient gets off the bed, their feet will land on the mat. As soon as feet are detected on the mat, a message will be sent to the server, recording that the person got off the bed.

When the person returns to the bed, their feet will again be detected on the mat, which will help in telling us the duration for which the patient was out of the bed.

The mat should be turned off by the caregiver in the morning once the patient has had their sleep.

Pillow:

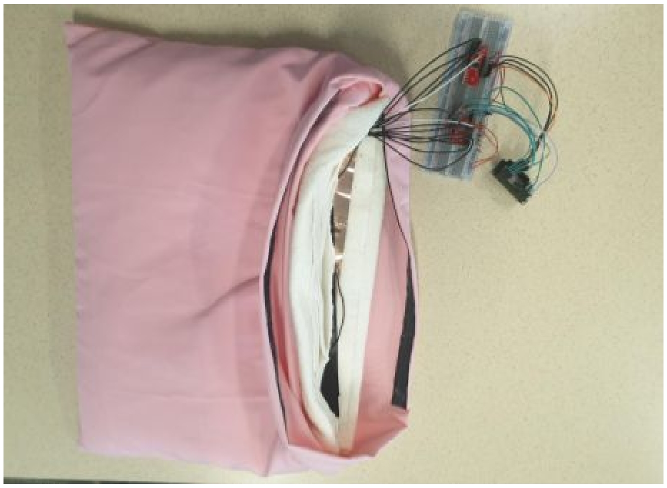
The pillow should be placed on the bed of the patient at the side where their head is expected to be while lying down for sleep.

Once the patient is lying on the bed ready to sleep, the pillow should be activated by the caregiver.

When the patient lifts their head off the pillow, a message is sent to the server, thus recording that the person is not lying down on the bed anymore.

If the patient lies down again, the pillow will stop sending the message to the server regarding absence of the patient’s head on the pillow. This will help us record the time duration for which the patient was not lying down.

The pillow should be deactivated by the caregiver in the morning once the patient has had their sleep.



Inference:

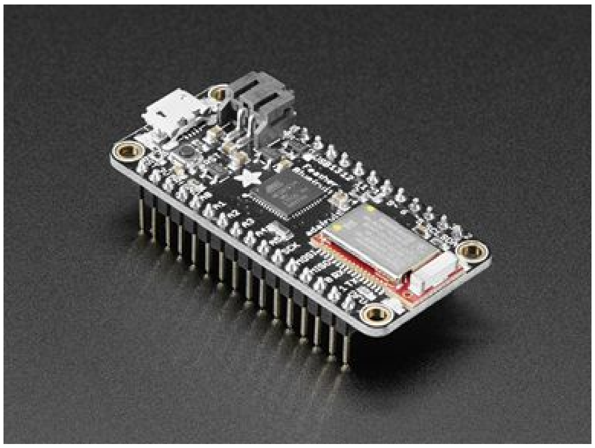
The messages received by the server from both the mat and the pillow help in detecting where the patient is.

If no events are recorded then that means that the patient is lying down on the bed. If the pillow detects absence of the patient's head on it and the mat also detects patient's feet, then that would mean that the patient got off the bed.

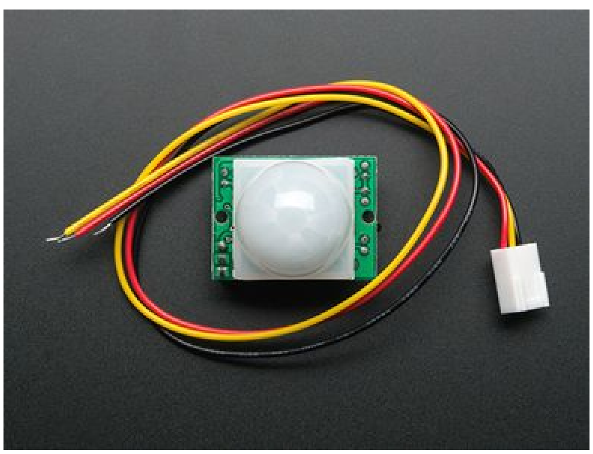
If the pillow detects absence of the patient's head on it, but the mat does not detect any feet, then that would infer that the person is not lying down but did not get off the bed (for instance, just sitting on the bed).

3.4.3 Components and Pictures

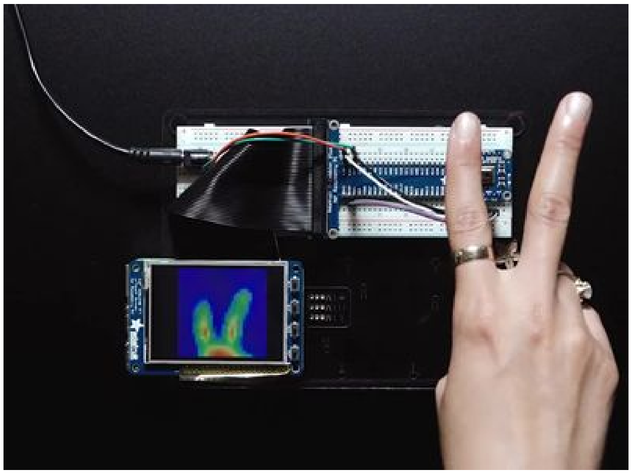
1. Adafruit Huzzah Feather Bluefruit LE
Controller with in-built Wifi and BLE module



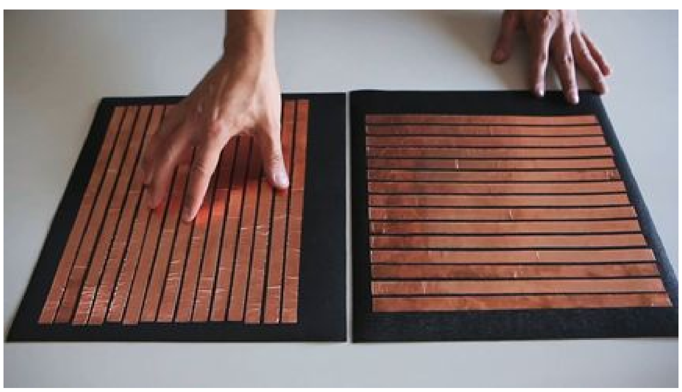
2. PIR motion sensor
To detect the motion of the patient in the restroom



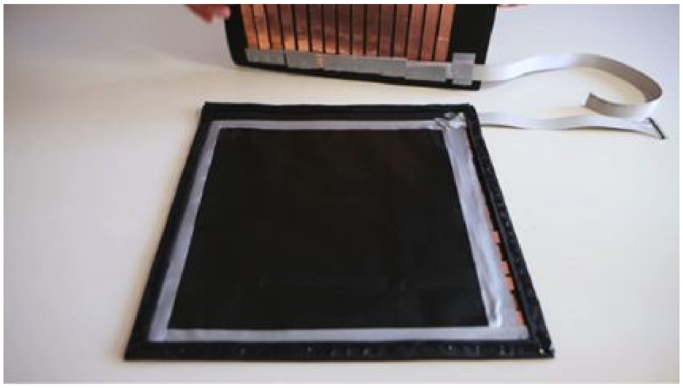
3. Adafruit AMG8833 IR Thermal Camera Breakout
To detect the flame of the burner in the kitchen



4. Copper strips grid



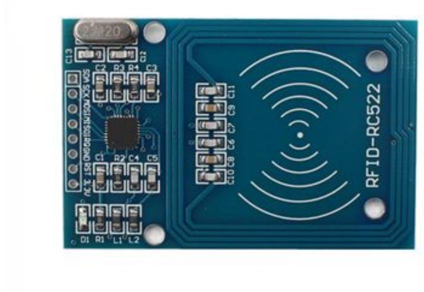
5. Sheet of Velostat



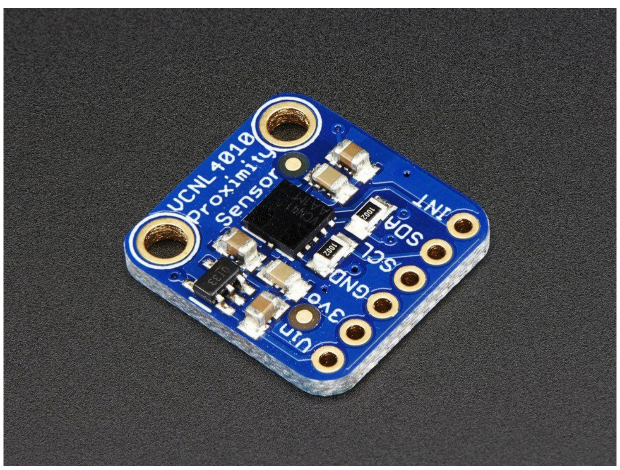
6. Pressure detecting pillow

7. RFID ring by Adafruit
To identify the patient across different doors

8. RFID reader RC522



9. VCNL 4010 Proximity/Light Sensor



3.4.4 Experimental Evaluation

3.4.3.1 Kitchen Monitoring

The experimental measurements of the kitchen sensors is given as follows:
IR thermal camera results: (Temperature grid 8X8)

[25 25 23 22 21 25 25 26
24 24 25 24 28 32 31 33
60 66 63 31 31 61 60 59
61 62 62 31 32 66 63 64
32 31 29 22 23 24 25 27
25 25 27 25 25 24 26 25
24 24 23 25 26 25 24 24

25 25 26 25 24 25 23 25]

The PIR motion sensor would detect the presence of a person in the kitchen. It detects a moving object upto 15 feet and (110° X 70°) detection range. It would be kept somewhere in the kitchen close to the stove for accurate detection of activity.
The primary idea of the IR thermal camera is to detect the temperature around the stove region. For temperature readings in the 8x8 thermal grid above a threshold of 60℃ would send a message to the caregiver, notifying that the stove is ON. The above 2 sensor readings can be used to decide if the message has to be sent.

Conditions and decision based on the results of above 2 sensors:

Scenario	Motion sensor output	IR thermal camera output	Decision
1	No motion detected	Stove is off	No message sent
2	No motion detected	Stove is switched on	Message sent to caregiver
3	Motion detected	Stove is off	No message sent
4	Motion detected	Stove is switched on	Message sent to caregiver

3.4.3.2 Restroom and Main Door Monitoring

We implemented an RFID sensor with a reader on the frame of door of the restroom and the main door and the reader RC522 detected the ring correctly from about a distance of approximately 2 cm. The reader was placed at the level of hands of the patients so that the ring detection id dimpler.
The PIR/Motion sensor is used inside the restroom to detect activity of the patients inside the restroom and the algorithm for the restroom considers the status of both the RFID reader and the PIR sensor. The PIR motion sensor accurately detects motion from about a distance of 15-20 cms and would be placed somewhere near the wash basin since that is the most commonly used place inside the restroom. This sensor sometimes gives false positives since it detects smallest activities and hence our algorithm is such that the caregiver is not alerted unless both the RFID and the PIR are detected as positive.
We also used sensor_id and sensor_type as the two fields to send data to the server so that the data analytics team can use these parameters to send the text alerts.

3.4.3.3 Sleep Monitoring

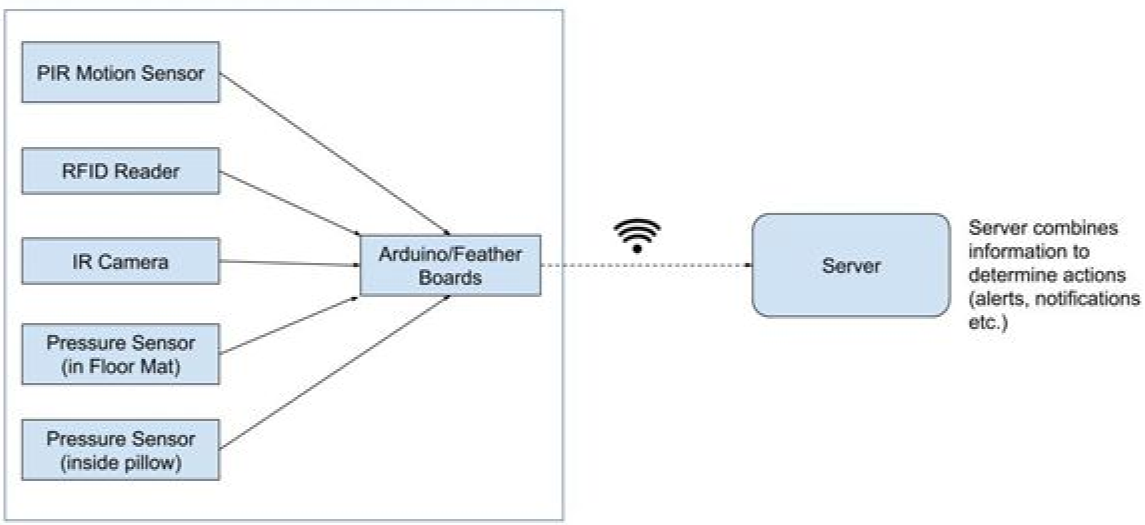
The resistance of the velostat sheet was measured under different weight conditions and the change observed was in the range of 3000 Ohms(no stress) to 7 Ohms(10kg weight). (Probably make a table with people of different weights standing on it?)

The bedside mat has a grid of 5x10 with each unit in the grid 2.5 square inches in area. The bedside mat is able to distinguish between random objects and a person’s feet and has a latency of 500ms between detections and 10 seconds between data points sent to the server. This means that if a person’s feet are detected on the mat, the next reading to be communicated to the server will be after 10 seconds.

The pillow has a grid of 5x9 and each unit of the grid is 1.2 square inches in area. The pillow has a latency of 1.5 seconds between complete readings of the entire grid and a latency of 5 seconds between consecutive readings sent to the server.

3.4.5 Hardware Architecture

Restroom, kitchen monitoring, sleep monitoring: Hardware Architecture

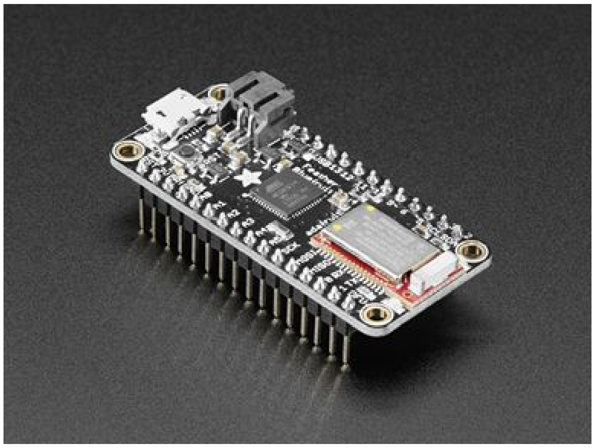


event_type	String. The type of event	"data", "handshake", "message", "error"
sensor_id	String, uuid	"9aa3ffd9-bfa4-4c7e-95ec-8bec12819301"
sensor_type	String. Type of the sensor	"PIR", "IR_CAM", "RFID", "IR_LINE"
data	Object, Data from the sensor	{value: 1, message: "MOTION_STARTED"}
timestamp	String	"2019-04-01T18:25:35+00:00"

Sensor type	Possible message
RIR	"MOTION_STARTED", "MOTION_STOPPED
IR_CAM	"STOVE_ON", "PERSON_DETECTED", "NO_EVENT"
RFID	"TAGGED"
IR_LINE	"BROKEN", "UNBROKEN"

3.4.6 Hardware Modules and Status

Adafruit Huzzah Feather Board :



The Adafruit Huzzah Feather board has an in-built Wifi and BLE. The ESP32 wifi module is used for communication with the server for all the subsystems. The code for this module can be uploaded using the Arduino IDE.

3.4.7 Software Architecture

There is no specific software architecture for our subsystems

3.4.8 Software Modules and Status

There are no specific software modules for our components

3.4.9 List of Components, cost, power

S.No.	Component	Cost	Power

1.	Arduino Mini Pro	$9.95	25mW
2.	Analog/Digital MUX Breakout (2 pcs)	$11.5	-
3.	Velostat (2 pcs)	$9.90	-
4.	Copper strips (tape)	$14.95	-
5.	Adafruit AMG8833 IR Thermal Camera Breakout	$39.95	3.3 Volts
6.	Velostat	$4.95	
7.	Copper strips (tape)	$14.95	
8.	Adafruit Huzzah32 (7 pcs)	$19.95	
9	PIR motion sensor	$9.95	3.3 Volts
10.	RFID ring Adafruit	$44.95	
11.	RFID RC522	$28.99	3.3 Volts
12.	IR Break Beam Sensor - 5mm LEDs	$10.80	3.3 Volts

3.5 Games/Assessments

3.5.1 Functionality

3.5.1.1 Logic Thinking

The games are implemented in Android Studio using Java and can be installed on an Android tablet. They have straightforward layout and/or behavioral guidance for a player to follow smoothly. The games have different difficulty levels to choose from where applicable. They also measure the completion time and accuracy of the player, calculate a numeric score out of 100 based on those two parameters and report the score both on screen and to the backend database through the score reporting utility.

3.5.1.2 Semantic Memory

Our games App is implemented by Java in Android mobile system, and NeuralUp is a web-based game platform, to combine our games App and NeuralUp, we use Webview in our App to open NeuralUp platform. For games related to semantic memory, we provide several different games, so user could choose to play one of them without feeling bored, and there are different levels of them, so they could choose one according to their own ability, and for each play round, we would generate the time that user use while play the game and the score that user achieve on this particular game.

3.5.1.3 Episodic Memory

Episodic memory is the section of Games and Assessment subsystem that deals with testing and recalling experiences related to personal life, that are an important part of the patient’s life as well as with other people who were a part of these events. The games that are a part of this section allows them to “travel back in time” and be consciously aware of a re-experience of important life experiences. They also help the patient to create a sense of personal history and shared history with other individuals of their life. The game is a form of quiz involving the testing of remembering specific events, situations or experiences. The recorded answers, accuracy, that is number of correct answers divided by total number of questions asked and time taken to answer these questions are stored for the doctor’s analysis to keep a track of the patient’s improvement.

3.5.1.4 Procedural Memory

Different from other kind of memory in this project, procedural memory is a subset of implicit memory, sometimes referred to as unconscious memory or automatic memory. Procedural memory stores information on how to perform certain procedures, such as walking, talking and riding a bike. Studies have shown that procedural memory can begin to fail with advanced or severe cases of Alzheimer's. Still, one study showed that Alzheimer's patients even in severe stages could develop new procedural memory and retain it for up to three months. This would indicate that people who have the disease could retain procedural memory for long after they lose other types of memory, but only if they use the skills frequently. So it’s hard to use normal online games to test whether patients are healthy in procedural memory. In this project, we assess patients’ procedural memory by asking them to put steps in correct order. Ordering games can help evaluate whether patients remember how to do a simple tasks without wrong operation theoretically. We can evaluate memory health according to the correctness and timing.

3.5.2 Interactions

For inter-group interaction, our group need to cooperate with the following group: core tablet, short term memory, database, data analytics and doctor dashboard.

We have discussed with core tablet team and confirmed the language and system they use. We built a github workplace to integrate our game on the core tablet. Since we only have one tablet, we need to share it with other group and do the testing.

We will use the same app as short-term memory team so we will also have a same workplace on github with core tablet team. We confirm the front-end layout and the integration of game and short-term memory.

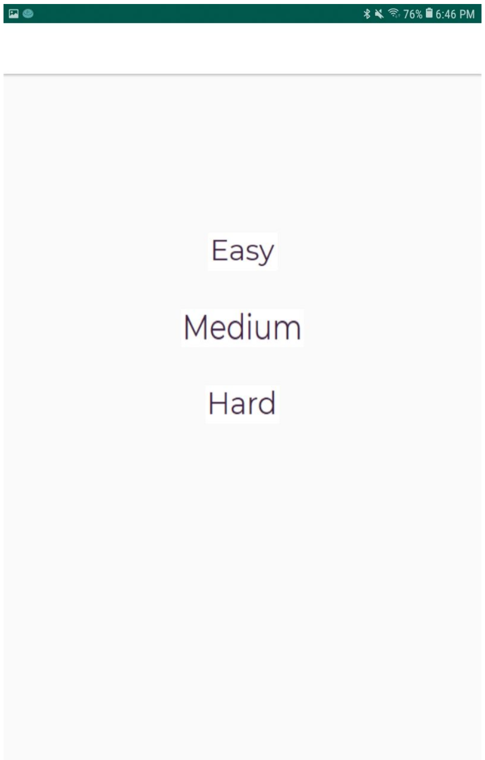
Also, since we know the demand and needs from data analytics team and doctor dashboard team, we need to ask database team to store the data they want, such as: patient id, game score, timestamp etc. Moreover, we need to record the image file and the answer of some games for the doctor use. Data analytics team and doctor dashboard team will get the data they want and generate patient's behavior pattern from the data.

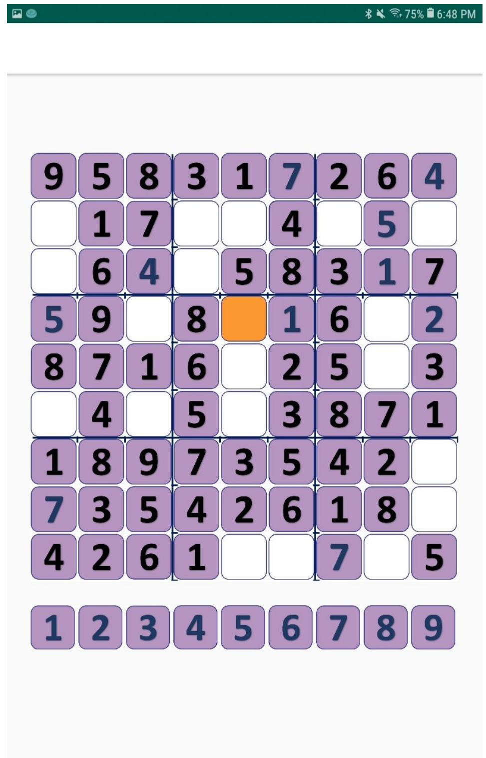
For intra-group, since we need to integrate four categories of games on the core tablet. Everyone will work on github workplace and integrate the game on the core tablet system. Group members need to scale the same output of the games to ensure the data can push to the database.

3.5.3 Type of Games

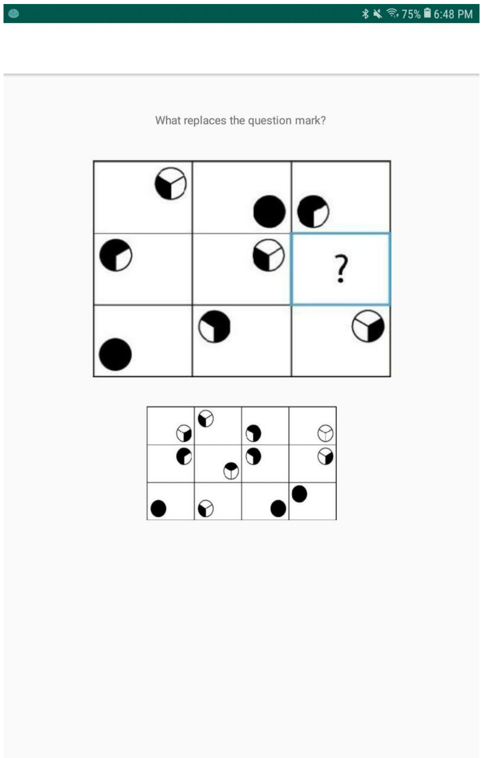
3.5.3.1 Logic Thinking:
The representative games we chose are Sudoku, Finding Patterns and Judging Statements, corresponding to numeric, graphical and verbal representations.

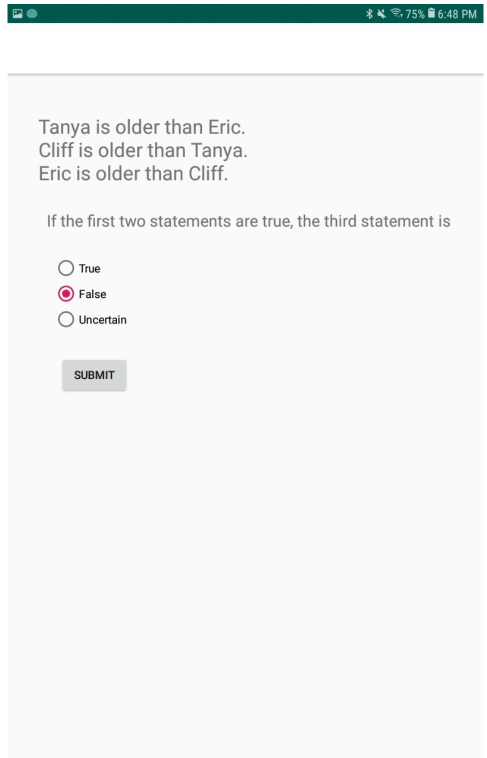
The Sudoku game is built based on open source algorithmic generator, which generates Sudoku puzzles with a specified number of empty cells to differentiate difficulty levels. We designed and constructed the visualization and interaction of the game on top of the generator. The game guides the player's operation by visual cues like color change in game elements and Toast messages.





The patterns and statements games are both quiz-like games with simple instructions on top of the question prompt. The demo questions and answers were all collected from online logical reasoning quizzes.

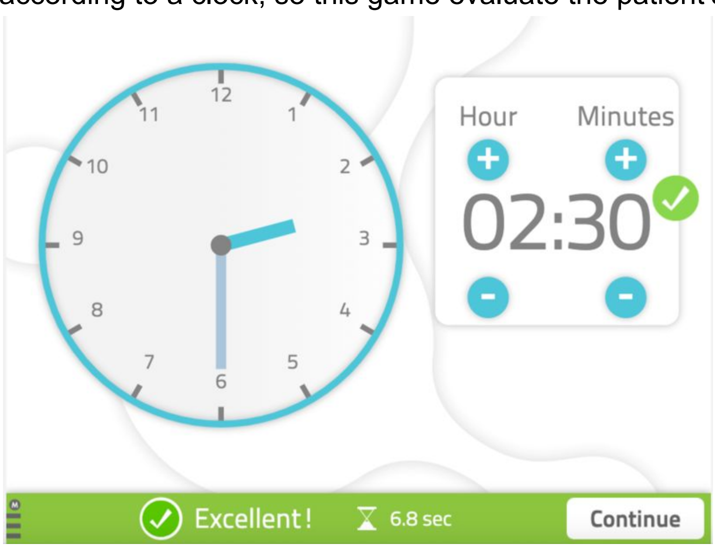




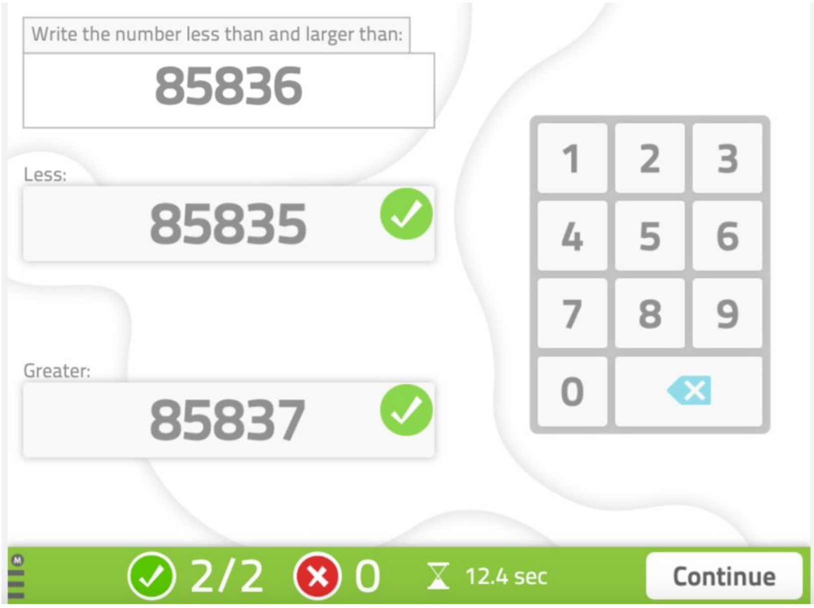
3.5.3.2 Semantic Memory:

There are about 9 different games related to semantic memory in NeuralUp, and we pick two of them to introduce because these two games are typical and easy to understand.

The first one is Telling time, in this activity, patients must type in the digital clock the time they see on the image. Every normal adult should have ability to recognize the correct time according to a clock, so this game evaluate the patient’s semantic memory according directly.



The second one is Greater than and less than, this activity involves writing the smaller and larger numbers immediately preceding and succeeding the reference number; this brain game is ideal for training semantic memory in patients. We could change the difficulty by generate simple number or complex number.



3.5.3.3 Episodic Memory:

Personal quiz is a list of multiple choice questions that the patient is supposed to answer. Following is the sample list of questions to be included in this test :

1. What is your full name?
2. Who lives in your family?
3. Where do you stay?
4. What is your favorite color?
5. What is your favorite pizza place?
6. What is your favorite sport?
7. From where did you pursue your masters?
8. What is the name of your grandchild?
9. What kind of music do you enjoy?
10. What kind of movies do you like?

The questions will be repetitive for a specific number of days to track the response of the patient for these consecutive days. The results of this quiz will enable the doctor to examine the recovery of the patient’s episodic memory and work on it’s improvement in case of deterioration of memory based on the test results.

Also, the questions will be in the form of the multiple choice format. The objective of this test is to recall personal memory and not put a stress on the patient’s memory. So, multiple choice questions are preferable to reduce the strain on the patient to think a lot and for the ease of scoring!

4.5.3.4 Procedural Memory:
For previous two phases, we discussed about Steps Ordering Games and Sorting games, both of which can train and test patients' procedural memory. Due to time limit, in this project the main part of procedural memory test is ordering game. In the demo game, we set cooking pasta as a example simple task. We break it into 6 steps and disordering them. As shown below. In this game we ask patients to put these steps into correct order and record the timing. We provide corresponding pictures to help patients understand and recognize what are these steps so this game is friendly and not frustrating. The pictures are movable and thus they can be dragged until they are in correct order. Patients input this order number and get a result back. The timer will not stop unless the order is correct. The timing and result will be sent to backend server for data analysis. So this game is pretty straightforward and the result is simple. Patients either pass or fail this task according to whether they put steps into correct order within limited time.

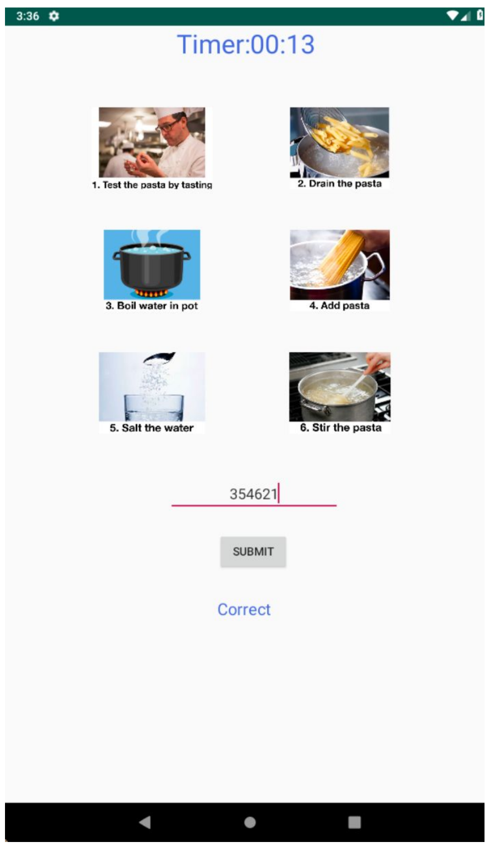


Figure : procedural game - steps ordering of cooking pasta

3.5.4 User Testing/Experimental Evaluation

3.5.4.1 Logic Thinking

The Sudoku has fairly good design on guiding the player’s operations, but could be improved by adding the “eraser” functionality.

The patterns and statements games are easy to follow with simple and straightforward layout, but could be better improved in terms of aesthetics of the layout. Since the questions were manually imported, there were only 5 questions for each game, but for the actual product there should be a larger pool of questions for better assessment.

3.5.4.2 Semantic Memory

The different types of games on NeuralUp have the same method to evaluate, they may both generate the score which reflects the number of correct answers and the time that patient finish the game. Using these two methods, we could easily understand the performance of patient in a particular game.

Because NeuralUp is not an open-source software and we could not get the API of it, it is hard for us to collect the score and time of our games to transfer to other team, we could use screen capture method or generate our own games to avoid this situation if there are more time for us.

3.5.4.3 Episodic Memory

Tests related to personal memory are not available for the desired compatibility of the systems. So the thought of buying these games has been totally discarded. It was then decided to build a personal quiz to implement the memory tests with the help of caregiver. The questions suggested by the patient’s caregiver and their correct answers are stored in database. These questions are displayed on the screen along with other possible redundant options and the patient is asked to select one option out of them. The game is built on Android Studio platform referring this website[1]. Fig 1 displays the Episodic memory quiz game while fig 2 shows the flow of the game. The patient is given about 20 seconds to answer each question. Once the time is up or if the patient gives a wrong answer, the game quits automatically.

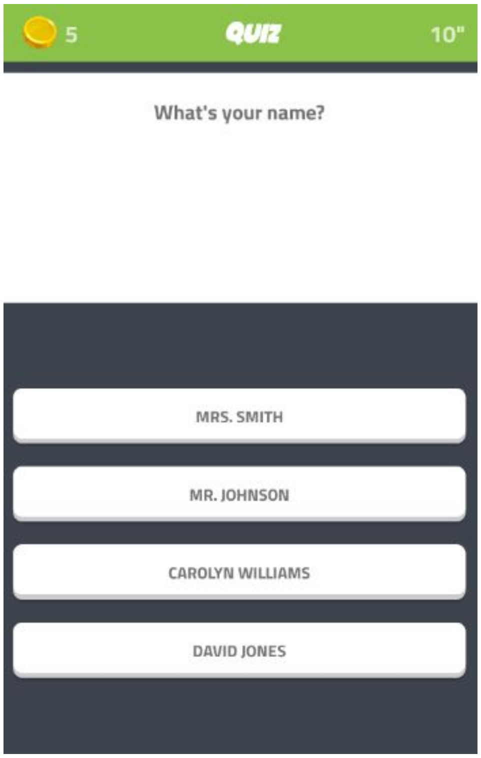


Fig 1. Episodic memory quiz game

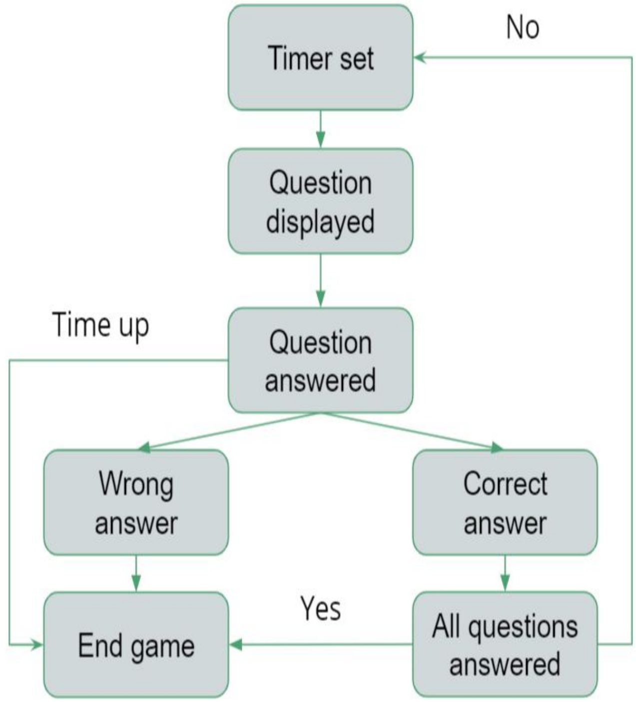


Fig 2. Flow of the quiz game

3.5.4.4 Procedural Memory

This game is a good way to test procedural memory "online". The virtual task is a example to present this idea and more tasks can be added in the future. And to enable more flexibility, questionnaire of user's preference for tasks can be added so that patients have the chance to choose tasks they are more familiar. Movable steps/pictures are good to help patients recognize the tentative order. However, this game can be improved by combining the movable pictures and input order. If moveable steps/pictures is automatically converted to steps order, patients are done once they drag all the steps to the correct order. That will be more straightforward and user-friendly.

3.5.5 Software Architecture

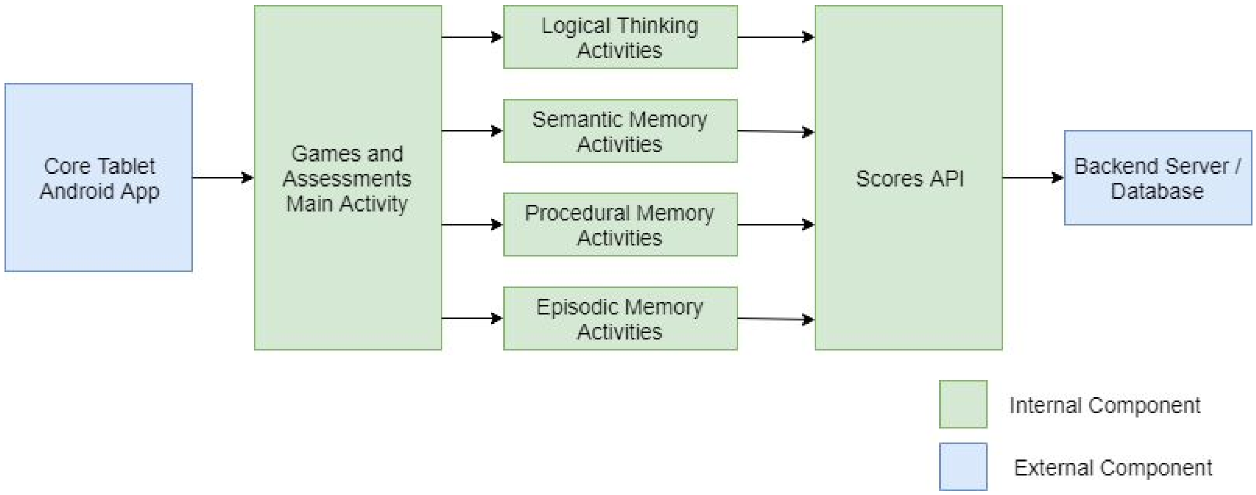


Figure 4.5.5.1 Software Architecture Diagram for Games and Assessments

The final software architecture for Games and Assessments has been revised since the iteration in Phase 2. The Games and Assessments component is now part of the overall patient Android application from the Core Tablet Team, and the Scores API now sends the scores directly to the Backend Server / Database through HTTP POST requests to the backend's REST API. The control flow of the Games and Assessments component is now very straightforward:

1. Patient opens the Core Tablet App.
2. Patient opens the menu to the games and assessments, which is the Games and Assessments Main Activity.
3. Patient selects a game / assessment, which launches an activity from one of the four categories of games and assessments.
4. Patient completes the game / assessment.
5. The activity automatically creates a score entry for the Scores API.
6. The Scores API sends the score to the backend server / database.

3.5.6 Software Modules and Status

The Games and Assessments component is divided into the following modules: Main Activity, Activities for each of the four categories (Logical Thinking, Semantic Memory, Procedural Memory, Episodic Memory), and the Scores API.

The Main Activity serves as the connector between the Core Tablet App and all of the games and assessments. It is implemented as a menu that allows the patient to select from all of the games and assessments. This module has been implemented, although the styling is not finalized.

The Logical Thinking category includes three activities: Sudoku, Patterns, and Statements. Each of the activities tests a different aspect of the patient's logical thinking capabilities. All of these activities are built, although the Sudoku activity also utilizes an open source library to generate puzzles. All three activities have been fully implemented.

The Semantic Memory category includes two activities: Greater Than and Less Than, and Telling Time. Both activities are web applications, bought from NeuronUp. Both activities have been integrated to the Android app. However, these activities currently require the patient to log in to the NeuronUp platform, and the score reporting system is not integrated yet.

The Procedural Memory category includes one activity that tests the patient's ability to recognize the correct sequence of steps to complete a simple task. This activity is built and is fully implemented.

The Episodic Memory category includes one activity that tests the patient's ability to remember facts about his or her life in a multiple choice format. This activity is a free quiz application with custom questions added, and has been integrated to the Android app. However, the score reporting system is not integrated yet.

3.6 Wandering Tracker

In this subsection, we shall describe all the components and the hardware relating to the Wandering Tracker part of our overall design. These components include a custom-made Wandering Tracker device that will be embedded into the patient's shoe. In this section you will find a description of each of the components of the Wandering Tracker team in relation to the overall goal of the system, pictorial representation of the components, experimental analysis of each component, description of chosen architectures and a list of all costs.

3.6.1 Functionality

The Wandering Tracker team is working on a wearable GPS-enabled (or equivalent) tracker to be worn by the patient in order to detect wandering incidents. While passive location detecting is normally a trivial problem (e.g., through a smart watch), it becomes challenging in the context of tracking Alzheimer's patients- who might be forgetful, confused, or otherwise resistant to using a tracker. Furthermore, the purpose of tracking the patient is to detect wandering in such a way that the caregiver and doctor can act on an incident.
The idea to track this activity arose from the simple fact that anyone with dementia is prone to wandering even in the earliest stages of the disease. The risks increase multifold when the

patient is autonomous as he/she may not understand their situation completely and may resist any attempts to be monitored all the time. Not only does this situation put the patient in great danger of facing untoward circumstances, but is also, a cause of great stress to the caregiver and may prove to be detrimental to his/her peace of mind.

Given the above, the idea to alert the caregiver as soon a patient starts wandering could help mitigate the potential risks that a patient could himself/herself under while also proving to be of great help to the caregiver. This would become especially important in public spaces and crowded areas where the chances of the patient and caregiver separating are high. The decision criteria considered for reporting an alert is on the distance separating the patient and caregiver.

Therefore, the functionalities of the Wandering Tracker can be viewed as those of a typical GPS tracker but with critical modifications to accommodate the challenges of Alzheimer's Disease:

1. The wandering tracker should include a battery able to retain sufficient power so as to remain functional during an extended wandering incident; the minimum battery life should be 3 days on a full charge.
2. The caregiver or patient should be able to charge the wandering tracker inductively (e.g., by placing the shoe containing the tracker in a particular spot on a shoe mat or rack).

The scenario envisioned for issuing the alert is as follows a patient has started to wander and is moving further and further away from the caregiver. This results in the distance between the caregiver and patient to increase. Once this is beyond a certain limit, the caregiver must be alerted.

Figure: Patient Wandering Interaction

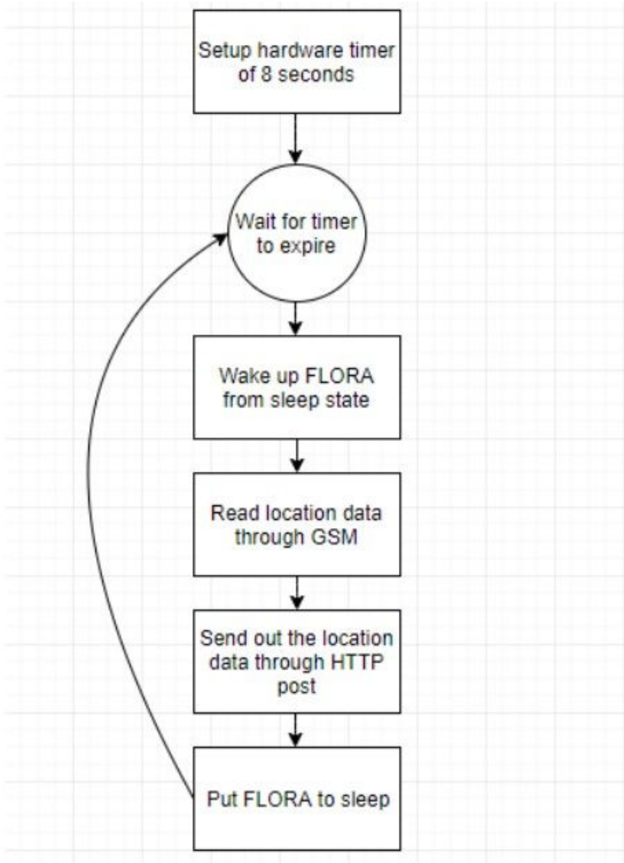


Figure: Architecture Flow Diagram

This would give the caregiver a chance to quickly react and catch up with the patient before it gets out of hand. In order to track the patient down, the caregiver would be aided by knowing the location of the patient.
The functionality of a tracker is not a novel idea. There are many commercial products available. However there is a subtlety that is often neglected. As Alzheimer's disease progresses in a patient, the patient always tends to follow what is referred to as procedural memory. Procedural memory is a part of long-term memory that involves actions that are carried out unconsciously. In simpler terms, a patient tends to do things that they have been doing all their lives. While most of the market provides wearable solutions for tracking in the form of wrist wraps, it is likely that patient will forget to carry a wearable with them as it is not something that they are used to doing. Citing the above reasoning, our team decided to implement the tracker that would fit into a shoe. A patient who is autonomous is never likely to leave the house without his/her shoe as this is something that is likely to be firmly entrenched in his/her procedural memory.

As caregivers of Alzheimer’s patients would be trusting the security of their loved ones to the device, a wandering tracker must be both dependable and reliable. It must be dependable in the sense that the wandering tracker should be ready for use whenever the patient leaves or wanders (or else the tracker’s useless), and reliable in that the tracker must continue functioning

as designed for the entire duration of a wandering incident (or else the caregiver loses the patient). Neither requirement can be met without ensuring that the tracker has sufficient energy to last during long periods of use and the capacity to stay charged at almost all times.
In our design, we had three options: (1) have a battery that lasts a long period of time (on the order of months of years) with mechanisms to ensure the battery can be reliably swapped out by the caregiver; (2) have no battery/a battery that only lasts a short period of time, but rely on energy harvesting or intermittent computing; or (3) have a battery that lasts an intermediate amount of time but can be reliably kept charged. Due to limitations posed by the Adafruit FONA GSM/GPS device and the high-power draw of our prototype circuit, option (1) was not viable for our implementation (but should be revisited in future work). The reliability concerns and software implementation challenges associated with option (3), as well as the FONA specifically requiring a lithium polymer battery with at least 1200mAh capacity, led us to adopt option (2).

3.6.2 Hardware Architecture

The inductive charging kit will be distributed across two physical components- a charging rack (or mat) and the wandering tracker shoe itself. Using a power source in the rack (e.g., a wall outlet), the inductive charging kit will transmit power across the rack into the shoe's battery. This will enable the shoe tracker to get charged wirelessly, the caregiver or the patient does not have to explicitly remember to charge the shoe every day, just placing the shoe at the everyday location i.e. on the mat, will automatically charge it. This battery would then power the microcontroller Adafruit Flora and GPS/GSM module. By using the combination of these two, the location updates will be sent to the server every 8 seconds depending on the requirements.

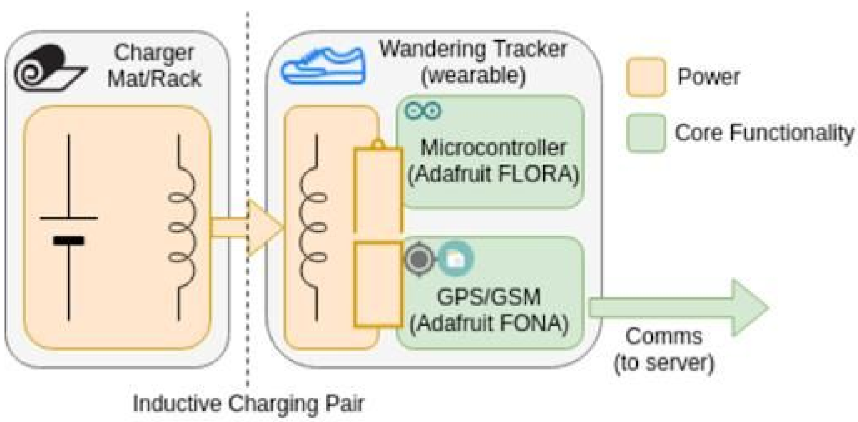


Figure: Wandering Hardware Architecture

In case of an emergency i.e. when the patient is out of the safe zone and the distance between the patient and the caregiver increases a certain preset value, a notification is sent to the caregiver, notifying him of the event so that immediate action can be taken.

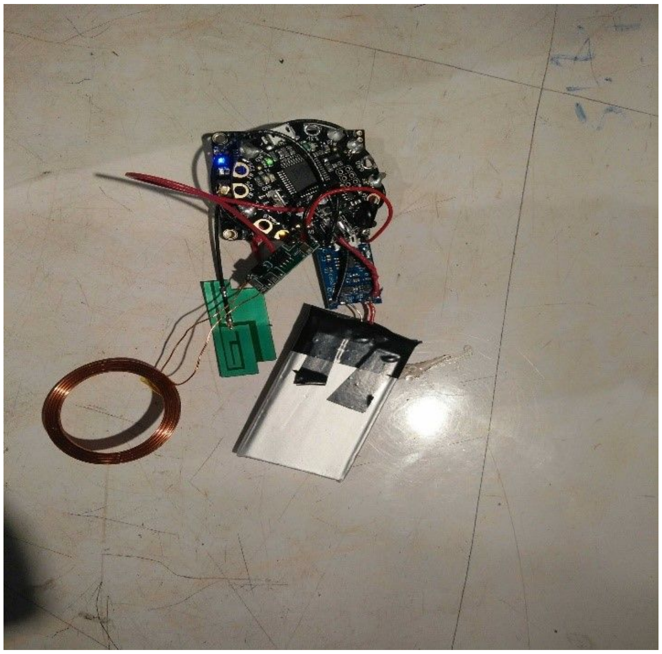


Figure: Finalized circuit

3.6.3 Hardware Modules and Status

Based on our custom design, we explain each module and their working:

1. Adafruit Flora Micro-controller
2. Fona 808 GSM module
3. T Mobile SIM card
4. 3.7V LiPo battery
5. Charging circuit using coils
6. 3D printed case

Microcontroller Adafruit Flora

FLORA is Adafruit's fully-featured wearable electronics platform. It's a round, sewable, Arduino-compatible microcontroller designed for wearable projects which is why we our team thought this would be suitable in order to be able to fit our product into a shoe. There is an onboard polarized 2 JST battery connector with protection schottky diode for use with external battery packs from 3.5v to 16v DC in.

The on-board controller will most likely be the most used component on the board once the code is ready. The Rx and Tx pins will be used to interface with the GSM module. These connections may be soldered in or a customized PCB may also be considered as an option.

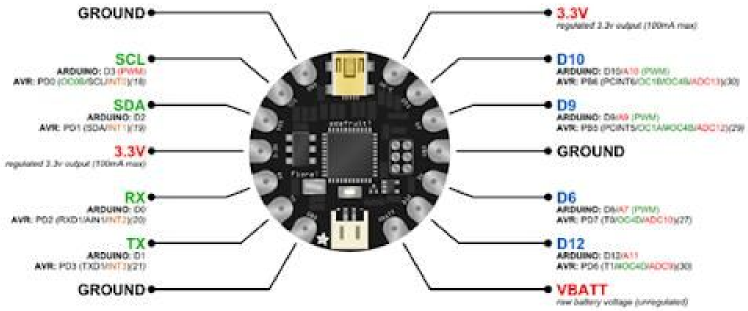


Figure: Adafruit Flora

Figure: FONA 808

GSM Module SIM 808

This GSM module is interfaced with an Arduino which is programmed accordingly. The GSM module uses
a T mobile SIM card which is used to send alerts to the caregiver via text messages. It also uses an antenna
when trying to connect to the nearest cell tower. After making the appropriate pin connections, the module
is tested using a sample code. The different applications of this GSM module include:
_ Make and receive voice calls
_ Send and receive SMS messages
_ Send and receive GPRS data (TCP/IP, HTTP, etc.)
_ PWM/Buzzer vibrational motor control
_ AT command interface with "auto baud" detection

Battery and Inductive Charging Circuit (Charging Circuit)

To implement the inductive charging circuit, the coupled inductors were first characterized to measure their efficiency in relation to distance, using a voltmeter and a series resistor. It was found that the inductors were capable of delivering enough current to charge the battery at a distance of up to 2 cm, but that at 2 cm they were only supplying 100 mA of current. Therefore, in building the actual inductive charging circuit, we focused on minimizing the distance between the coupled inductors- which was ultimately reduced to only a few millimeters (the thickness of the fabric used in the charging mat). It's worth noting that, due to the 13.56MHz frequency of the charging circuit, the material placed in between the inductors was found to have no effect on the current delivered.

The lithium-polymer battery was simply attached to the FONA using its battery and ground input pins. The output of the inductive charging circuit was connected to the FONA's external input pins (5V and ground, connected to USB input) as the FONA's built-in charging circuit was capable of using this power to charge a 3.7 V lithium-polymer battery. Due to the physical constraints of prototyping (e.g., soldering these connections), this charging connection was found to be somewhat unreliable under actual use due to the risk of short-circuits.

Finally, it must be noted that the inductive charging design used here was highly inefficient due to constraints imposed by prototyping (chiefly the limitations of the Flora-FONA circuit and the inefficient power draw of using hardware that vastly exceeded minimum functionality requirements). For future work, we suggest further exploration of battery type (e.g., switching to less-inflexible pouch cell batteries) once the lithium-polymer constraint is lifted, as well as developing a smaller, more efficient wireless charging solution. Based on our design and analysis, we expect a custom-built printed circuit solution to this problem to last significantly longer (by at least an order of magnitude) than the prototype circuit and be simpler/faster to charge than the prototype implementation.

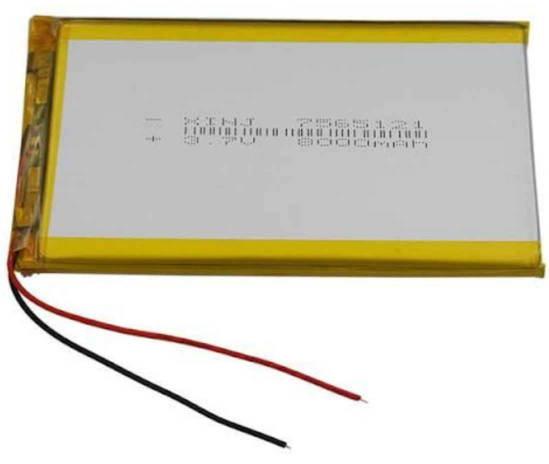


Figure: LiPo Battery

3.6.4 List of Components, Cost, Power

Components	Cost	Power
Whistle Tracker	$199.99	Typically 5 day battery
goTele	$99.99	48hr
Adafruit Flora	$16.24	5V input
SIM 808	$49.95	5V input
3.7V 8000mAh Li Po battery	$45.18	29.6 Watt-hours

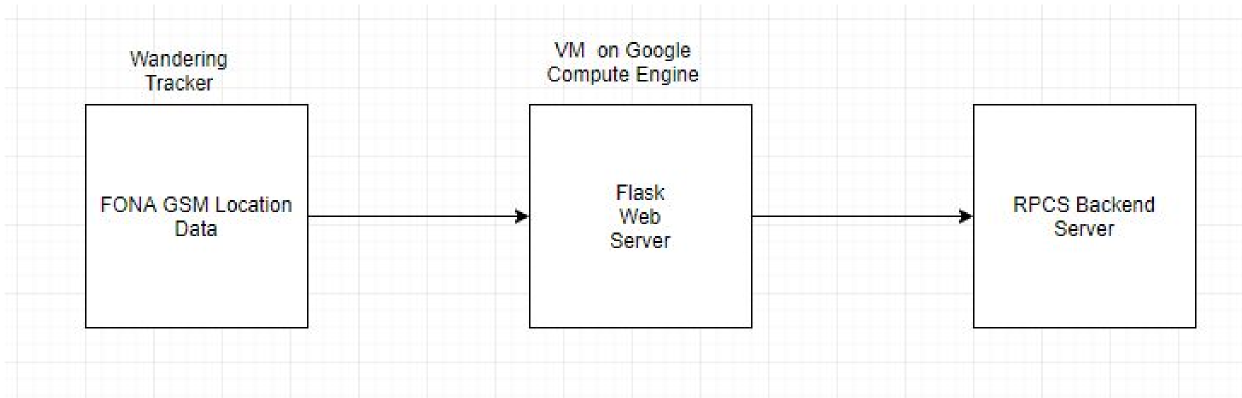
Table: List of Components, Cost, Power for Wandering Tracker

3.6.5 Software Architecture: (Sending data to the server)

Communication between the wandering tracker and RPCS server –

The FONA's HTTP module is capable of sending HTTP post requests with the URL and location data embedded. However, the RPCS backend server also requires the username and password sent within the HTTP request headers as part of authentication before accepting any data sent to it. It was discovered that the FONA is not capable of including these headers in the HTTP request. As a result, as a part of testing, it was discovered that the HTTP requests posted to the test server were recorded but the requests posted to the RPCS server were rejected. As a result, a workaround had to be put in place to facilitate the communication between the wandering tracker and the RPCS server.

The solution used was to set up an intermediate server between the wandering tracker and the RPCS server. A virtual machine was setup using Google Compute Engine's platform. Using the Flask python web development platform, a python script was developed to mimic the functionality of a server and then was run on the virtual machine. The script accepted HTTP POST requests along with the data, added the necessary user name and password request headers to the main request and then forwarded the request to the RPCS server. The attempted solution proved to be successful and the wandering tracker 's location data could be seen reflected on the RPCS backend server.



3.6.6 Experimental Evaluation and Pictures

Custom Design: The larger aim is to have the entire assembly embedded in a shoe. It has been observed that when a person affected by Alzheimer's wanders late at night or essentially at any time during the day, the person makes sure to put on his/her shoes. Since the module is quite small, we plan to fit it inside the shoe and enable it to send regular text alerts to the caregiver. We plan to further expand on this idea so as to track the person wearing this device. For phase 3, we used the Adafruit Flora micro-controller.
The battery's output voltage was measured against specifications using a multimeter (within 0.25V of the 3.7V specified). Similarly, we verified the input/output power and voltage for the inductive charging kit and charge capacity of the battery (using measurement tools on Adafruit's GSM board).
The induction kit was tested at distance to see whether it would still be able to charge the battery, with and without materials (plastic/foam) inserted in between.

We traveled around campus with the entire circuit assembly to check its accuracy and positioning. The circuit was fixed on one of our shoes. The microcontroller was programmed to send the current location coordinates to the backend server at regular intervals. The location updates could be clearly seen on the backend server. We used Google Cloud Platform to constantly send the location updates. We observed that each different location had varying coordinates. The latitudes and longitudes we observed were verified with the actual positions.

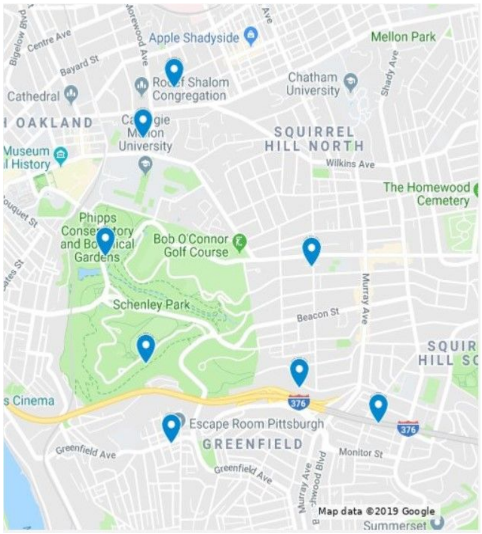


Figure: User Evaluation around campus

Figure: Location coordinates around campus (text alerts)

Figure: Final Shoe Design

3.7 Server/Database

3.7.1 Functionality

In final phase, server/database team implement the versatile server which can storage and request data base on GraphQL grammar. Our team create tutorial for those who is not familiar with GraphQL and create a portal for teams to get familiar with API we created for them. Some functions are created specific to team to let users create record and retrieve record in a certain table. The robustness of the server/database is tested. Also, the API documentation is provided in docs. According to the user feedback, there's no apparent bug in the implementation of server/database. The system architecture is also very easy for developer to extend.

3.7.1.1 Storage

One of the major functionalities of the server/database is to store all the data from sensors and tablet. Data will be pushed to the database from sensors, such as GPS location from the wandering tracker, and then pulled out by the tablet for the patient, caregiver, and doctor.

As all information is centered on the patient, there is a patient table containing basic patient data including name and a unique ID. The unique ID is used to key into other tables in the database and therefore associate sensor data with the patient that the data is being taken from. The database also stores basic information for the doctor and caregiver to allow them to log in to the tablet/web application, and the caregiver location is tracked to detect wandering.

Also, the analytics results are also saved in the database. If doctor or caregiver wants to monitor the advanced analytics result, he/she can just use simple API to retrieve results.

3.7.1.2 General Request, GET and POST

GET and POST request are the two most common request in networks. By sending a GET request, user is able to get certain data for further usage. By sending a POST request, user is expecting to add a new record or change the existing data in the database.

Most push events are by the sensors sending data to the server, and pulling is done by the tablet and web applications to view and display data. Additionally, the Core Analytics group pull data out of the database to do analysis and then push their results back into the database. Their results are displayed on the tablet and web application or potentially trigger alerts to the caregiver and doctor.

For how to use GET and POST request, see 4.7.3 for detail.

3.7.1.3 Security and Authentication

For security, we used the HTTP Authorization: Basic authentication scheme. This means the client needs to provide the server with the username and password with the Headers in the HTTP Request. Specifically, we encoded the username and password of each group with base64. The users should put the base64 encoded username & password pair joined by colon in the header of HTTP Request.

For authorization, we implement the permission for each username so that they can only access their own table and make change to it. This will avoid the users to mistakenly modify the tables which are belong to other groups.

3.7.1.4 Specific Authentication for each team

As mentioned in last part, here's the specific authentication for each team.

AuthN/AuthZ Info for Each Group

Group	Username	Password	Description
Core Analytics	ca_user	rpcs_ca2019	Read only access to /ca/* urls (HTTP GET)
Core Tablet	ct_user	rpcs_ct2019	R/W access to /ct/* urls only
Games & Assessments	ga_user	rpcs_ga2019	R/W access to /ga/* urls only
Home Sensors	hs_user	rpcs_hs2019	R/W access to /hs/* urls only
Interactions	int_user	rpcs_int2019	R/W access to /int/* urls only
Short Term Memory	stm_user	rpcs_stm2019	R/W access to /stm/* urls only
Watch	watch_user	rpcs_watch2019	R/W access to /watch/* urls only
Wandering Tracker	wt_user	rpcs_wt2019	R/W access to /wt/* urls only
READONLY (For anybody)	readonly_user	rpcs_readonly2019	Read only access to any url (HTTP GET)

3.7.2 Database Templates

Below, we list the final version database templates for each group. These templates are based on the phase2 design and are updated to fit the implementation requirements for each group and we update the data type to be consistent with Django models implementation.

There are three tables to store data from Wandering Tracker.
Data template for Wandering Tracker: Patient Table:

Data Name	Data Type	Description
location	CharField	GPS data (latitude, longitude), max_length=500
timestamp	DateTimeField	Date and time at which the packet was received
patient_id	IntegerField	Patient's unique identifier, default=0
wt_patient_id	AutoField	Primary Key

Data template for Wandering Tracker: Caregiver Table:

Data Name	Data Type	Description
location	CharField	GPS data (latitude, longitude), max_length=500
timestamp	DateTimeField	Time at which the packet was received
caregiver_id	IntegerField	Caregiver's unique identifier, default=0
wt_caregiver_id	AutoField	Primary Key

Data template for Wandering Tracker: SafeZone Table:

Data Name	Data Type	Description
location	CharField	Center point of the safe zone, max_length=500
radius	FloatField	The radius of the safe zone, null=True, blank=True, default=None
patient_id	IntegerField	Patient's unique identifier
wt_safezone_id	AutoField	Primary Key

There is one table to store data from Short Term Memory sensor.
Data template for Short Term Memory: Patient

Data Name	Data Type	Description
patient_name	CharField	Patient's name, max_length=50
patient_id	IntegerField	Patient's unique identifier, default=0
scaled_rating1	IntegerField	How the patient feels about activity1, default=0
scaled_rating2	IntegerField	How the patient feels about activity2, default=0
test_results	TextField	Test results descriptions, blank=True

The tables to store data from the Games and Assessments Team are to store one table for each patient per test type. Each table is for the test score for one patient of a given test type.

Data template for Game and Assessments: Logical

Data Name	Data Type	Description
patient_id	IntegerField	Patient unique identifier
logical_score	IntegerField	Patient's score for logical thinking test
timestamp	DateTimeField	Date and time for test score creation.
game_id	IntegerField	Game test unique identifier, default=0

Data template for Game and Assessments: Semantic

Data Name	Data Type	Description
patient_id	IntegerField	Patient unique identifier
semantic_score	IntegerField	Patient's score for semantic test
timestamp	DateTimeField	Date and time for test score creation.

Data template for Game and Assessments: Procedural

Data Name	Data Type	Description
patient_id	IntegerField	Patient unique identifier
procedural_score	IntegerField	Patient's score for procdedural test
timestamp	DateTimeField	Date and time for test score creation.

Data template for Game and Assessments: Episodic

Data Name	Data Type	Description
patient_id	IntegerField	Patient unique identifier
episodic_score	IntegerField	Patient's score for episodic test
timestamp	DateTimeField	Date and time for test score creation.
question	CharField	Episodic test question.
answer_choices	CharField	Episodic test choices.
patient_answer	CharField	Patient answer to the test question.
correct_answer	CharField	The correct answer to the test question.

There are two tables to store data from Home Sensor. For each patient, we store all the relevant events for the patient. Also, we store every sensor information on the patient.

Data template for Home Sensor: Event

Data Name	Data Type	Description

event_type	CharField	Event type
sensor_id	UUIDField	Sensor unique identifier monitoring the event.
sensor_type	CharField	Sensor type
timestamp	DateTimeField	Date and time for the event creation.
data	CharField	Meta information of the event.
event_id	AutoField	Event unique identifier, primary key

Data template for Home Sensor: Event

Data Name	**Data Type**	**Description**
location	CharField	Event location
patient_id	IntegerField	Patient unique identifier
sensor_type	CharField	Sensor type
sensor_id	UUIDField	Sensor unique identifier monitoring the event.

There are two tables to store all the data that Core Tablet: Patient need to visualize the data on the table platform.

Data template for Core Tablet: Incident Table:

Data Name	**Data Type**	**Description**
patient_id	IntegerField	Patient id.
incident_id	IntegerField	Incident id.
timestamp	DateTimeField	Date and time for this incident creation.
pulse_rate	FloatField	Pulse rate
respiratory_rate	FloatField	Respiratory rate
blood_pressure	FloatField	Blood pressure.
incident_type	CharField	Incident Types. (hallucination/frustration/agitation)
recording	CharField	Recordings for this incident from any device
details	CharField	Incident details.

Data template for Core Tablet: Trends Table:

Data Name	**Data Type**	**Description**
patient_id	IntegerField	Patient id.
test_core	IntegerField	Cognitive game and assessment score.

num_falls	IntegerField	Number of falls
num_injuries	IntegerField	Number of injuries
weight	FloatField	Patient weight
body_fat_ercentage	FloatField	Patient body fat percentage

Data template for Core Tablet: Wandering

Data Name	**Data Type**	**Description**
patient_id	int	Patient's unique identifier
caregiver_id	int	Caregiver's unique identifier
is_wandering	boolean	whether the patient is wandering
alerted	boolean	whether the caregiver is alerted

There are two tables for Watch group.

Data Template for Watch: Patient:

Data Name	**Data Type**	**Description**
patient_name	CharField	patient's name
patient_id	IntegerField	patient identifier
event	CharField	the event happens to this patient
event_id	IntegerField	event id associated to the event

Data Template for Watch: Event

Data Name	**Data Type**	**Description**
event_id	IntegerField	event id as primary key
event_discription	TextField	event description
event_category	CharField	event category (ie. log, reminder)
timestamp	DateTimeField	Date and time for event creation.

3.7.3 Testing/Experimental Evaluation

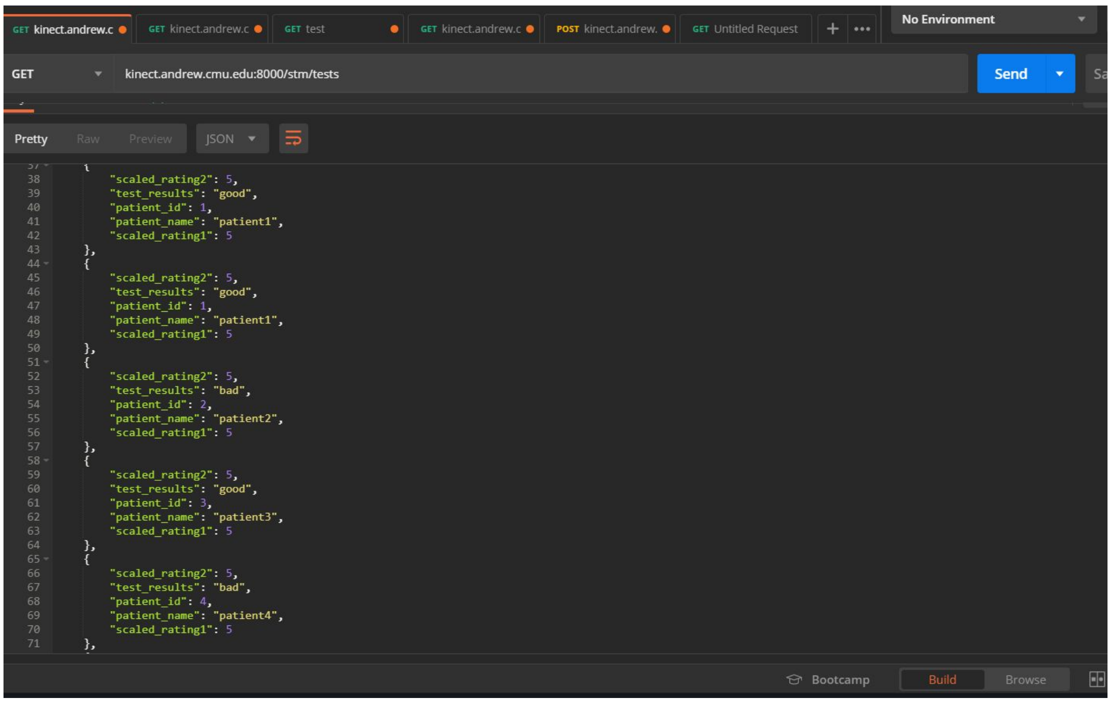
We tested the core database functions using POST and GET requests with Postman, a tool for API development that allows you to send requests and receive responses through a user-friendly GUI. The GraphQL portion was also tested using GraphiQL, the UI endpoint that accepts requests and delivers results in a more readable format.

3.7.3.1 Correct GET Request

Using REST

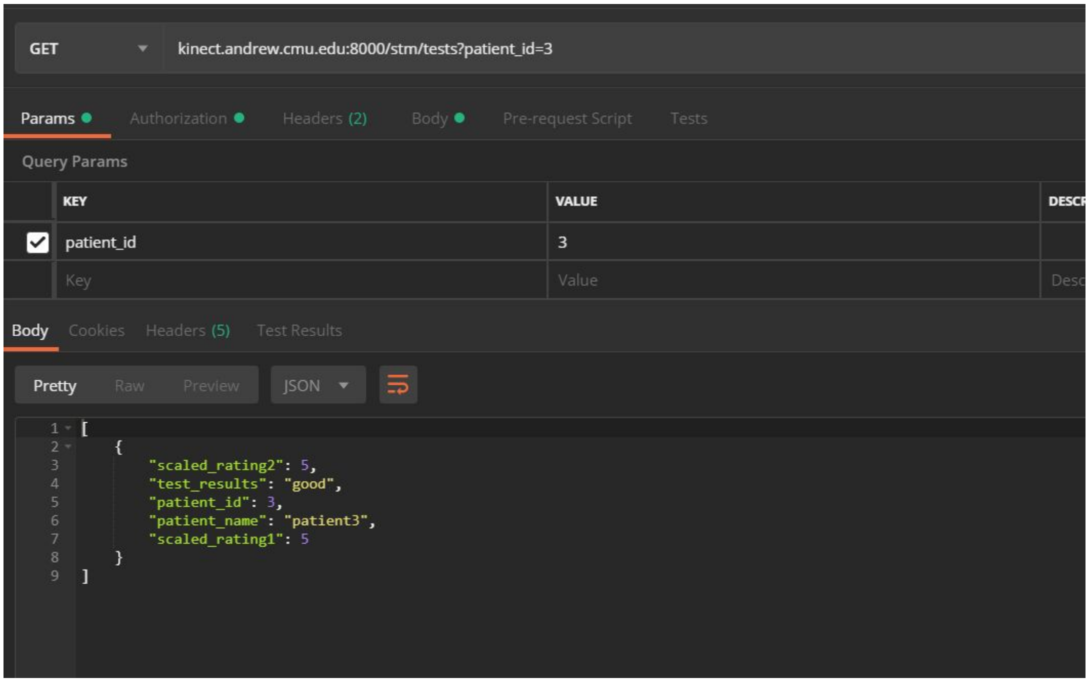
No Query Parameters

A request is sent to the URL kinect.andrew.cmu.edu:8000/[group_name]/[model_name]. Shown below is an example using /stm/tests with no additional query parameters. In this case, the server responds with all the entries in the ‘tests’ table.



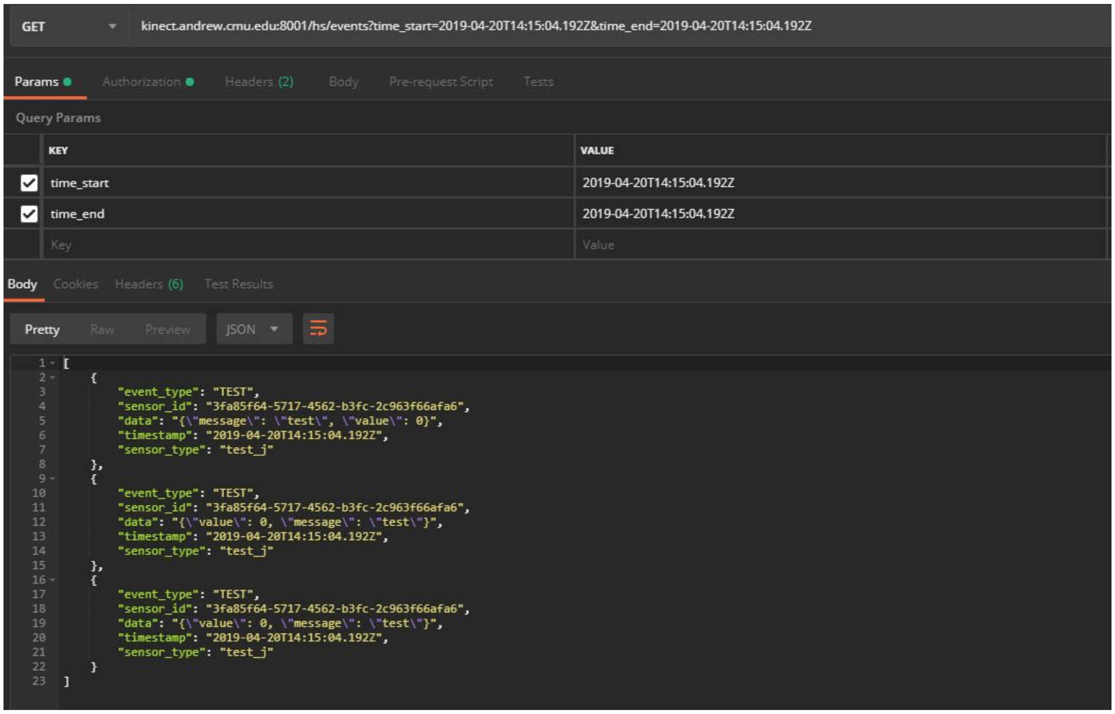
With Basic Query Parameters

In the example below, a query is sent to the endpoint with query parameter: patient_id=3. The server appropriately responds with only the entry with patient_id = 3.



With Timestamp for Data over a Date Range

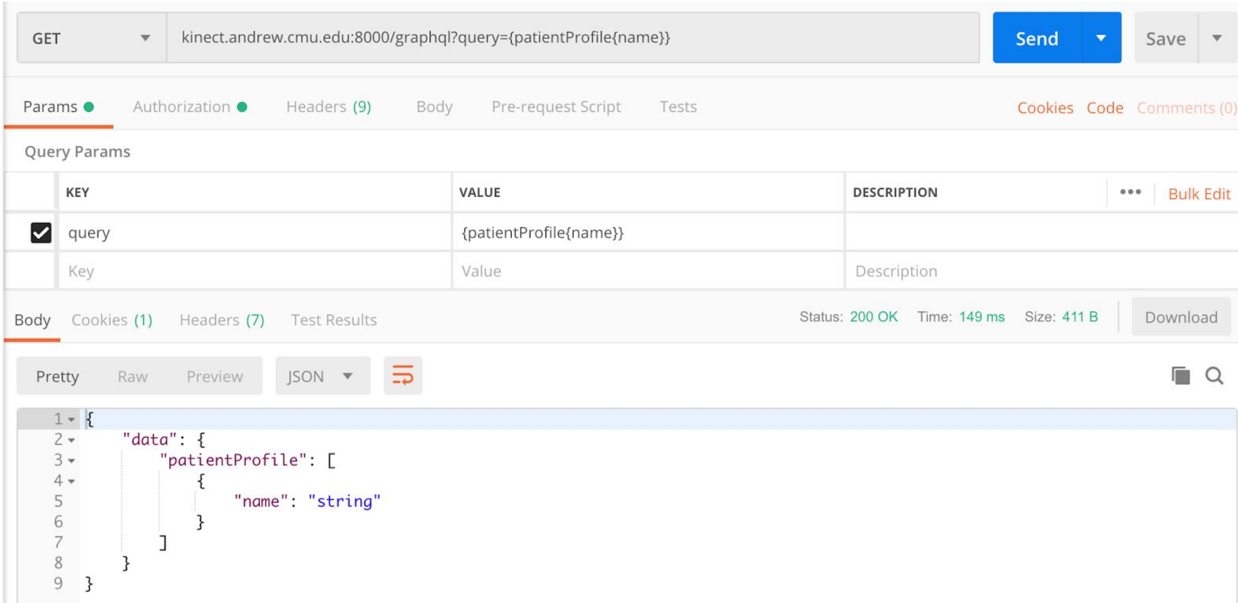
To account for the need to display data only within a certain date range, we added a feature to filter by date range for Home Sensors. The query parameters for this are time_start and time_end, and the server will return the data between those two timestamps, inclusive. If only time_start is entered, it will return all data from that date onward, and if only time_end is entered, it will return all data up till that date. This functionality can be seen in the example below.



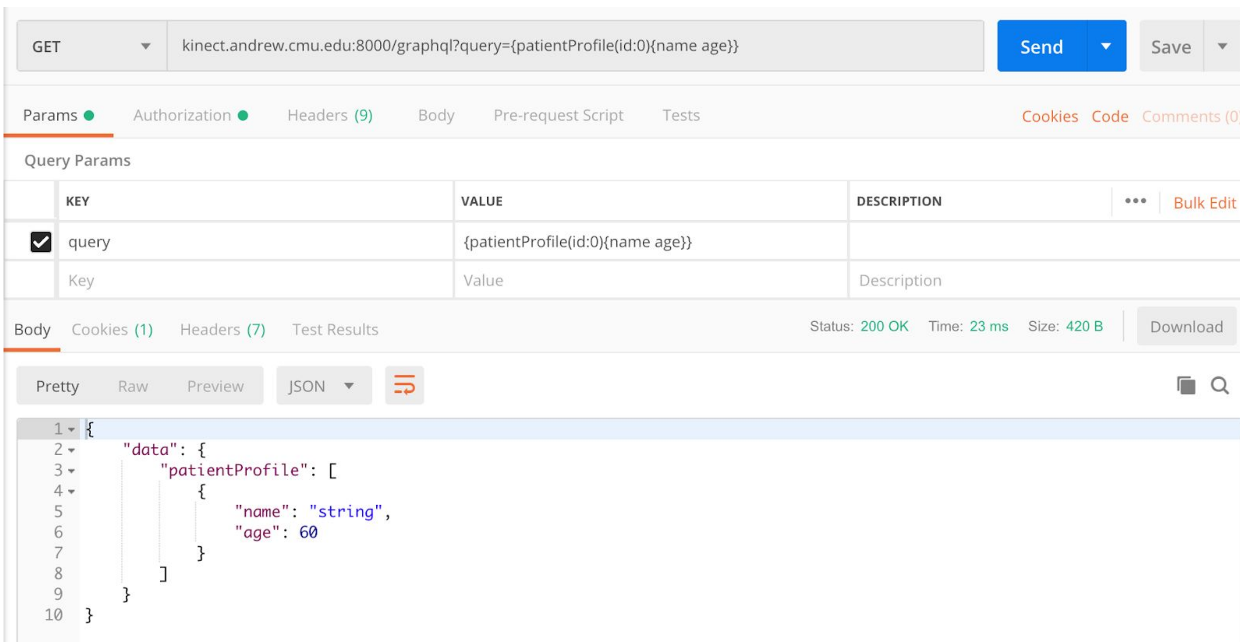
Using GraphQL

Basic Query

A GET request is sent to the GraphQL endpoint, kinect.andrew.cmu.edu:8000/graphql to query for the all the entries in the patientProfile table. Since there is only one endpoint, a query parameter must be included to specify what model/table it should be fetching from. In this example, the query parameter is {patientProfile{name}}, meaning that it will be fetching all the entries in the 'patient profile' table. The second portion of this query, {name}, specifies that it will only be fetching the 'name' field from each of these entries. There is only one entry in the database so it only fetches that one entry, but if there were more entries then this query would return all of them.



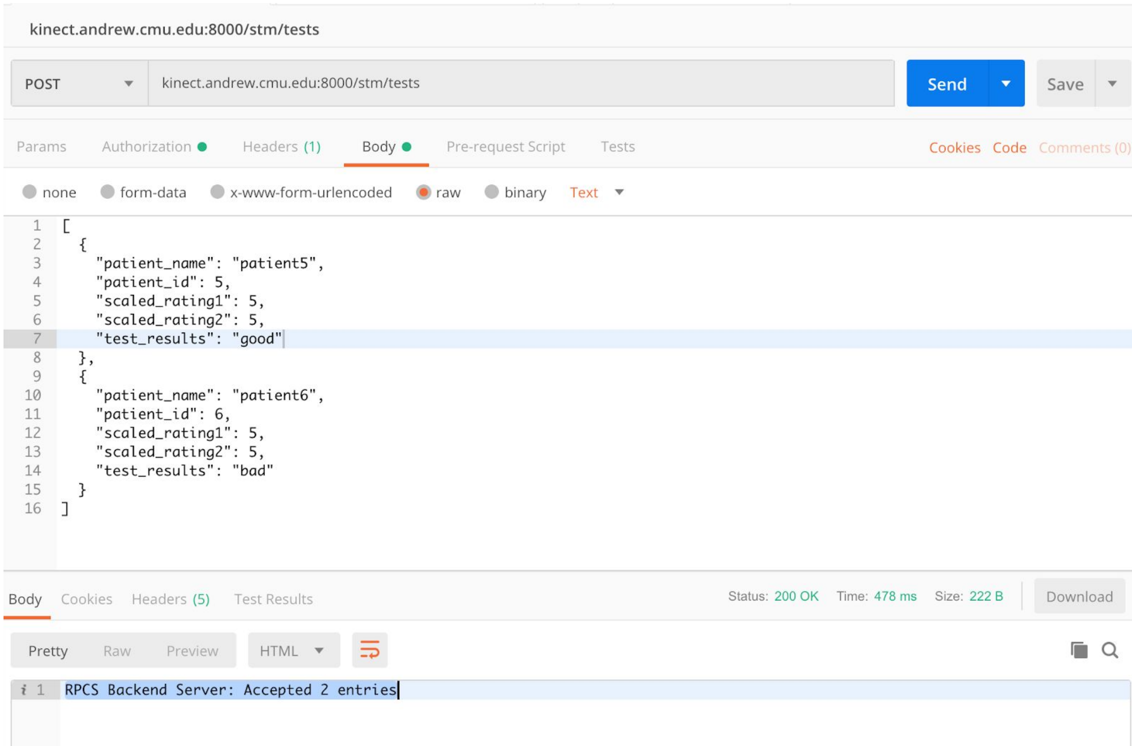
With Query Parameters
GraphQL can also filter by a field in the model. In the example below, the query value is {patientProfile(pid:0){name age}}, so it returns the 'name' and 'age' fields of the patient profile with patient_id=0.



3.7.3.2 Correct POST Request

Using REST

Just as with GET, a request is sent to the URL kinect.andrew.cmu.edu:8000/[group_name]/[model_name], and the example below again uses /stm/tests. The server responds to a valid POST request with how many entries it received, so the database has now ingested the two sets of data that were sent into the 'tests' table.

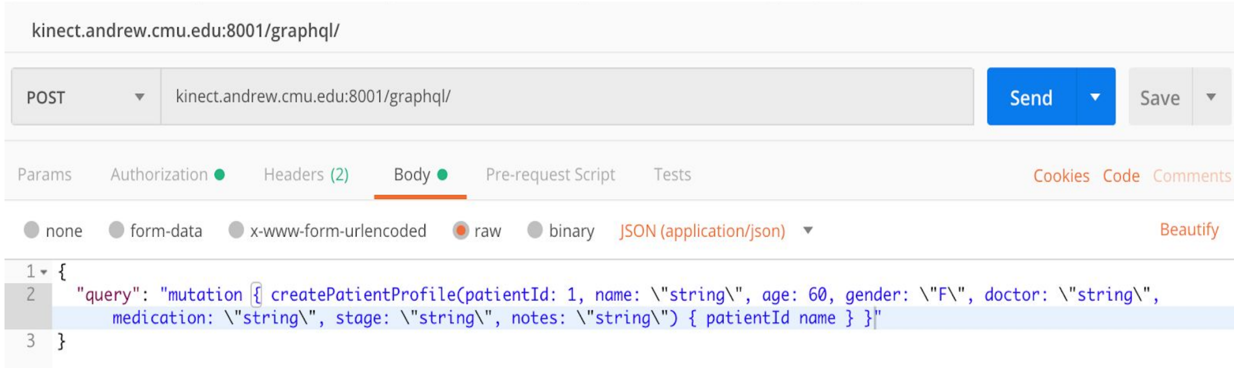


Using GraphQL

In GraphQL, POST requests can be used to both push and pull data.

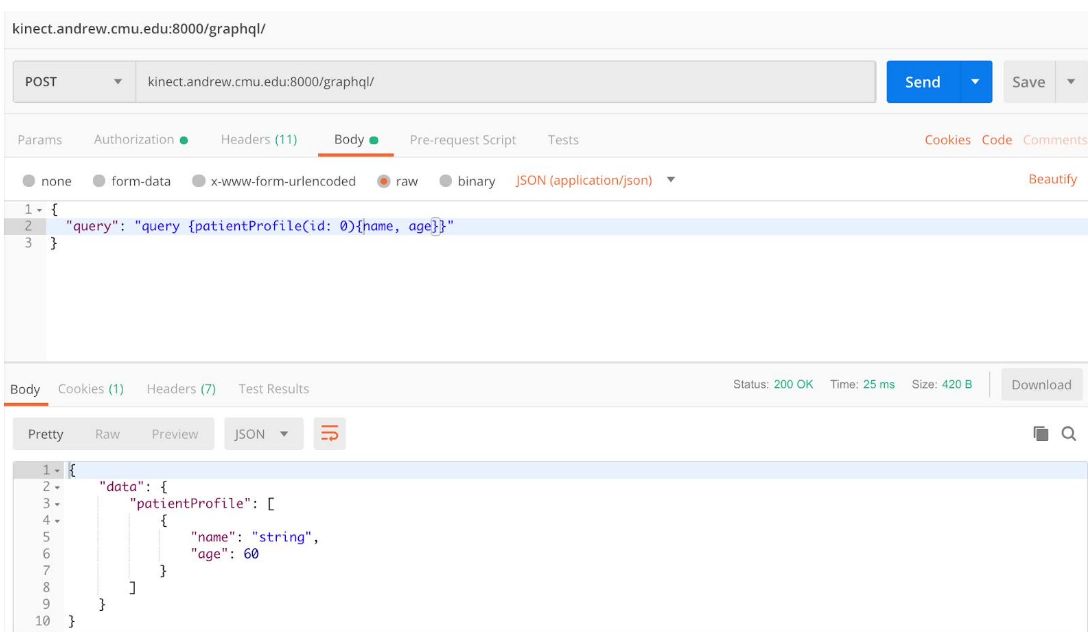
Pushing Data

Pushing data is called a mutation in GraphQL. A mutation is done by sending a POST request with JSON body to the GraphQL endpoint with the "query" field. In the example below, we insert an entry into the 'patient profile' table with all the necessary fields. The server will respond with whatever fields are within the curly braces at the end; in this case it is {patientId name}, so 'patientId' and 'name' will be returned in the response.



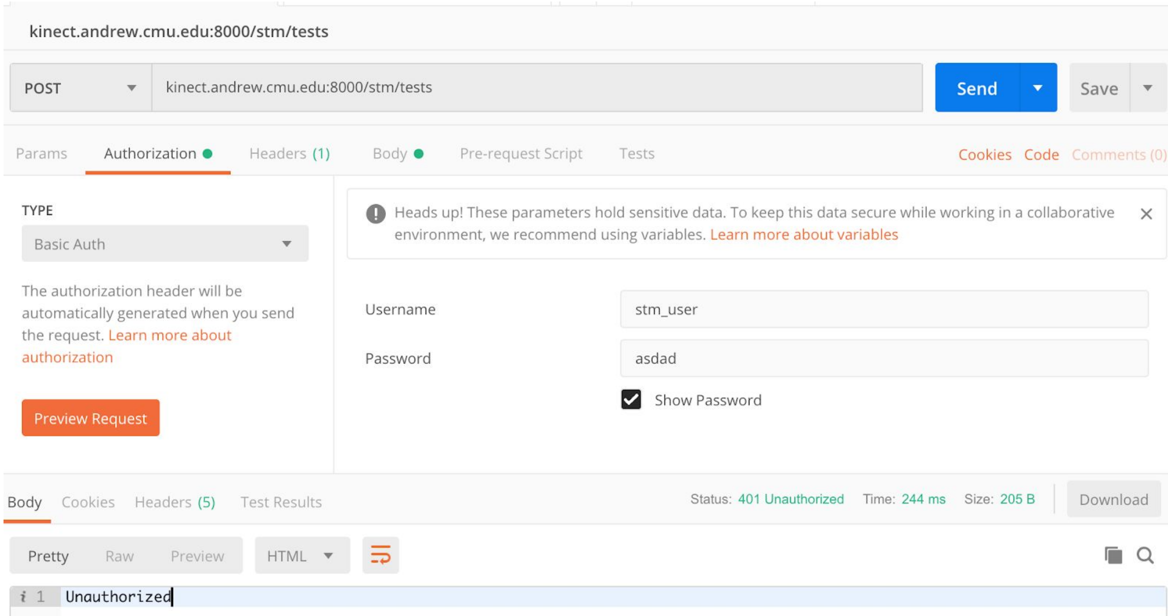
Pulling Data

In the GraphQL part of Section 4.7.3.1, we pulled data using a GET request, and the example below shows the same query using a POST request. The value of “query” in the JSON body is the same as the value sent in the GET request, and the format of the POST request is the same as the one for the mutation above, but this time it sends a query (pull request). This request will return the name and age of the patient profile with patient_id=0, the same as before.



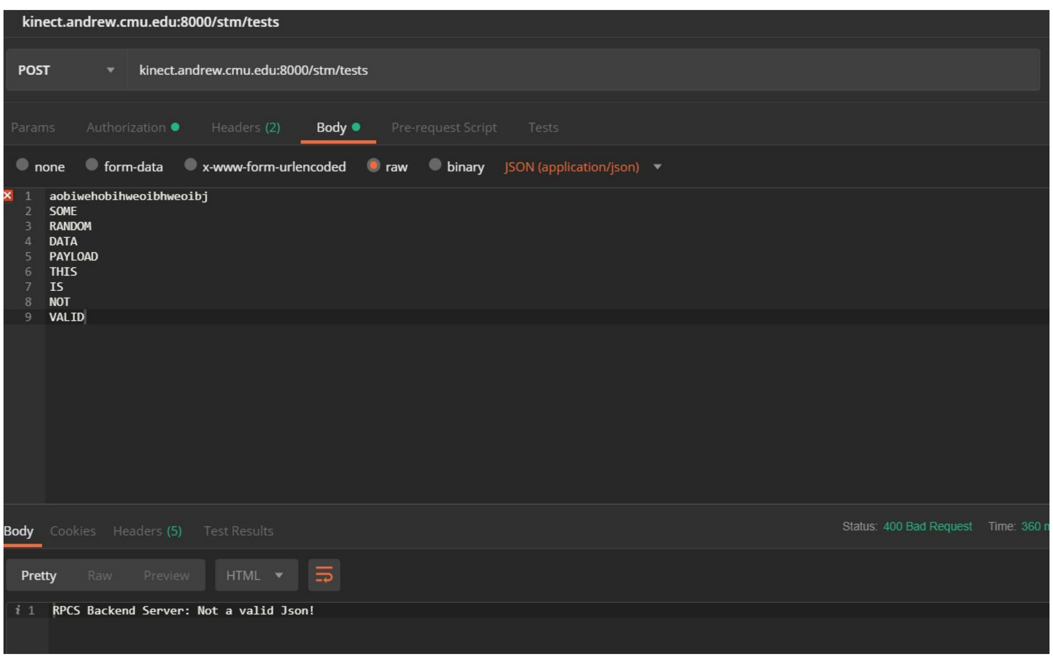
3.7.3.3 Operations Without Passing Auth

The correct username and password must be passed in as authorization in order for requests to succeed. The example below shows a request sent with an unauthorized username/password combination, and the server responds appropriately with “Unauthorized.”



3.7.3.4 Corner Cases

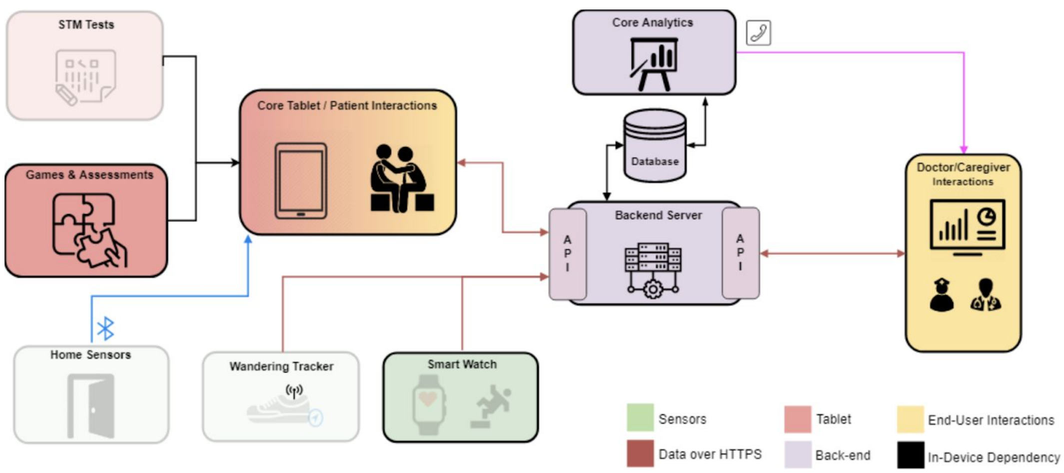
The server is also robust in making sure it is receiving properly formatted data. If it receives non-JSON data, it will respond with an error and not ingest the data. In this example shown below, garbage data was sent to the endpoint, and the server responded with “Not a valid Json!”.



3.7.4 Software Architecture

The system architecture is shown in the following picture. Basically, data would be gathered from several mobile devices, like core tablet, wandering tracking and smartwatch and then pushed automatically to our backend server, which would store all the data into the database. The mobile devices could also automatically pull the data from the server to see the update of data gathered from other mobile devices.

Also, doctors and caregivers would get the real-time situations of the patients and the corresponding analysis by making a request to our backend server through API. The backend server would retrieve the data from the database and deal with the data based on some analysis algorithms and automatically update the analysis result back to the database to keep everything updated.



3.7.5 Software Modules and Status

The backend server's specifications broadly form 5 different software modules. These aren't strictly observed in the codebase itself as different modules, but each functionality can be thought of as different modules. All of these modules were completed in time (and even way before) for the demo/presentation. We were able to have the server up-and-running with almost all of the required functionality about 2 weeks prior to the demo to allow other groups to test/integrate/develop their functionalities against our server.

<u>REST API Module</u>
This is the core communications system that serves as a medium between the data sources (sensing teams) and the data sinks (interaction teams). Each team has their own paths outlined in the API documentation, and used different actions (HTTP GET, HTTP POST) to provide or retrieve data from the server.

Authentication Module
This module was used to authenticate any incoming requests for data pushes/pulls. It outlined the username/password for each team such that each wouldn't stomp on others' data as well as having unauthorized, potentially malicious third parties having read/write access to our data.

GraphQL Module
Generally, front-end software specifications change quite a bit more frequently than back-end specifications. UI/UX changes require more subtlety and dynamism, and as such the software development process reflects that as well. Therefore, this module was used to provide a more flexible interface to the server to help the front-end teams change less of their code per spec change.

Core Analytics Module
The backend server team was not in charge of this (The Core Analytics team was), but it serves as an important facet to the backend server's codebase. The analytics code used the data that was ingested by the GraphQL/REST API modules to realize the phase 1 visions of the sensing teams, whether that be caregiver notifications given anomalous data or data aggregation for better visualization for the doctor.

Core Server Module
To increase developer velocity, we used Django to automate the nitty-gritty/baseline operations that any backend server needs. This includes support for connections between different databases (Postgres, MySQL, SQLite) as well as a standardized software interface for servicing requests. Django allowed us to not have to rewrite the same HTTP parsing code over-and-over again, and provided us with many importable modules that allowed us to continue writing impactful code for the teams that depended on us without us having to spend too many hours writing boilerplate code.

3.8 Core Tablet

3.8.1 Functionality

As core tablet, our main task is to develop interactions with patients. Patients have different scenarios and they need different types of help. For example, they need an app to track objects to show where they are when patients lose them. And they need reminders and even alerts to notify them of their activities. In addition, patients need to do some short memory tests using a tablet and calm down by playing some music or do calm down exercises. We have designed an application based on these scenarios.

When the user opens the app, there is a navigation bar to the left of the screen which can direct users to different pages including the calendar and notification page, games and exercises page and the my info page.

Our application can remind patients about their activities. Patients and caregivers can check and add activities and alerts through the calendar and notification page. The calendar page will show a calendar of the current month and will show a red dot on specific day when there is something important on that day. By clicking on a day in the calendar, users will be directed to the notification page which will show the detailed plan and activities for that day. Patients and caregivers can add notifications like take medicine. When it is time to take medicine, the app will alert and remind the patient in case they may forget to take their medicine.

Patients also can do some tests and exercises through our app. The games group will develop games which will benefit patients' cognitive abilities during the process. Patients can get access to these games by going to the exercises page. This page also has meditations which are useful to calm patients down.

Patients may forget things about themselves, like their family members and where their home is. Because of this, we developed a my info page to show some personal information like locations, name and pictures of family members, age, etc. This page can continuously remind patients about their important personal information.

Function	Detail
Homepage	1. User can user the navigation bar to navigate to different pages. 2. Display some frequently used components
Calendar: Check Schedule	1. User can select the day he wants to see 2. User can see event title, description, time, repeat frequency for each event
Notification: Edit patient's day	1. User can add event to schedule by clicking add icon 2. User can edit events on a pop out window
Games Page	1. User can click 'Game' button to play games designed by Game and Assessment Team 2. User can do some calm down exercise by listening to some music, doing some painting.
Personal Info	1. User and caregiver can check personal

	information about the patient. 2. Caregiver can edit the personal information as well

3.8.2 Tests and Pictures

Iteration 1

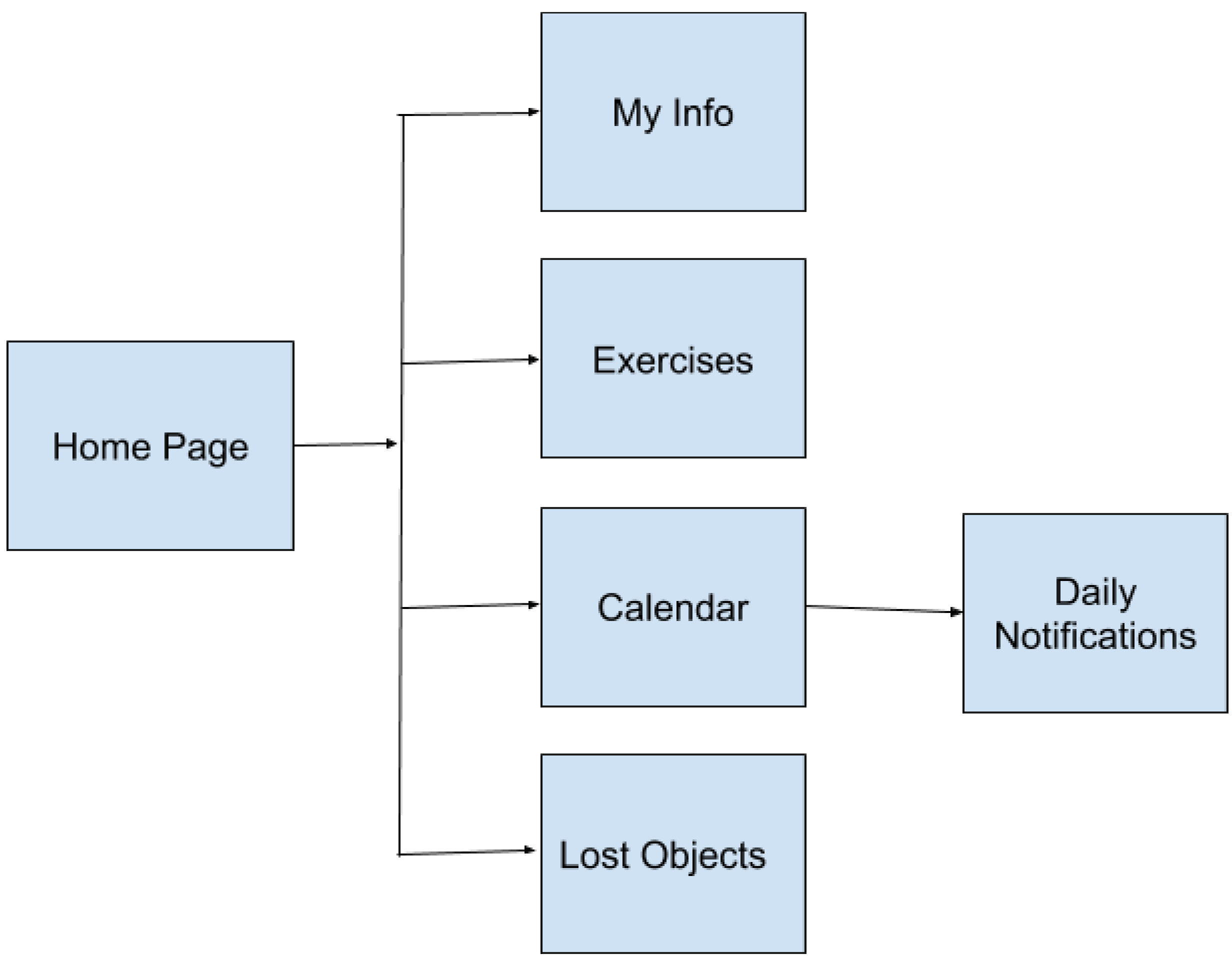


Figure 1 Iterations 1

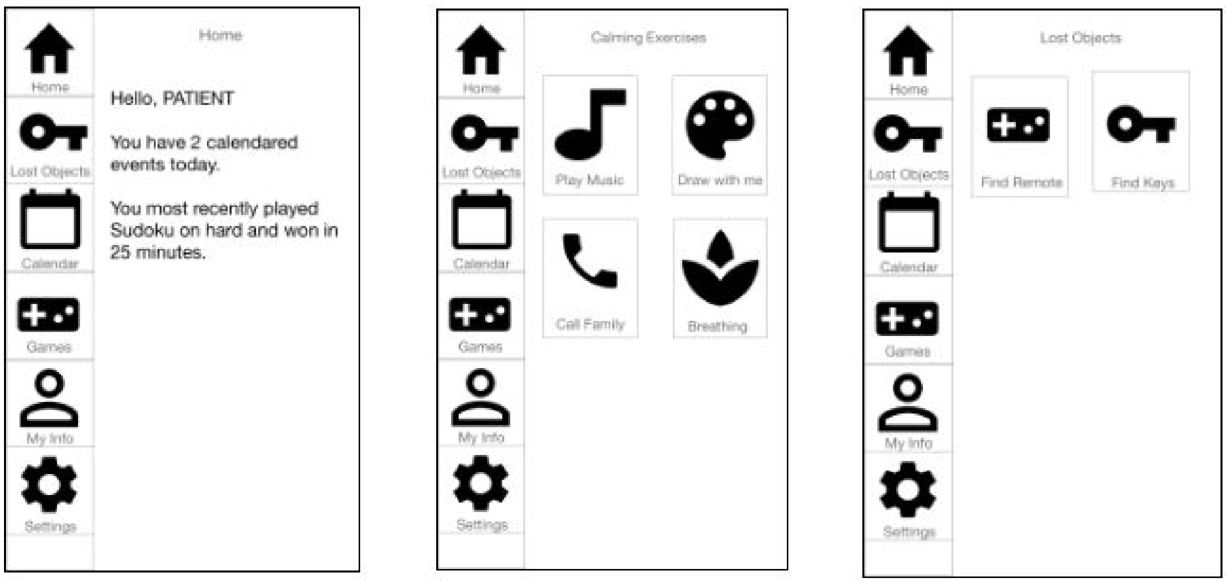


Figure 2 Initial Design

Our initial prototypes focused on getting very basic screens to get user feedback on. We did not build out all screens as we found ourselves desiring to build mid-fidelity prototypes as quickly as possible, as we felt that our color scheme was an essential piece of the interface that our user tests would be missing.

Iteration 1 User Feedback

- The tile based navigation might be a bit overwhelming for a senior, let alone a Alzheimer’s patient
- Can you trust a patient with their own settings?
- There should be more visual information, at least in terms of what is on the home screen
- Maybe include photos of contacts/family when they hit call family?

Iteration 2

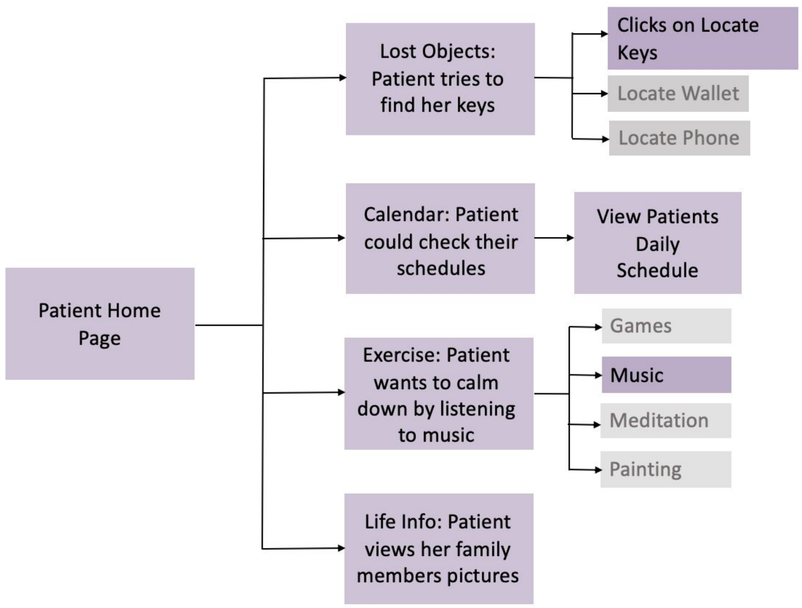


Figure 3 Iterations

Our mid-fidelity prototypes added more visual information, interaction, and color to our wireframes.

- The color scheme was researched to be a non-overwhelming solution to building applications and technologies for senior citizens.
- Got rid of all the navigation bars on all pages but the home page, and implemented back buttons to reduce visual clutter.
- For the calendar page, we modified the wireframe to a more visible and the user will be more clear about the reminders for certain days.
- Added more visuals and reminders to the home page in order to give the patient some immediacy upon opening the application.

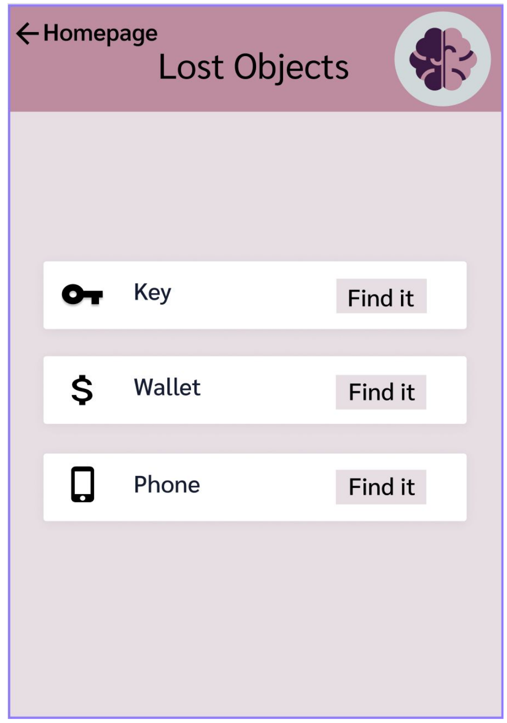
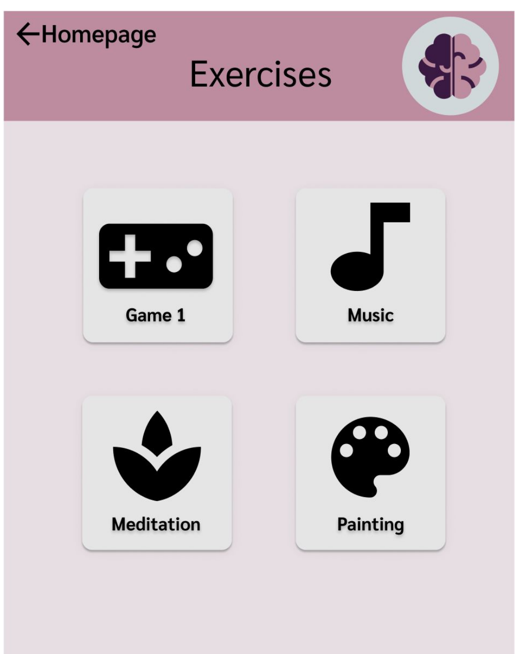


Figure 4 Home Screen V2

Figure 5 Lost Objects V2



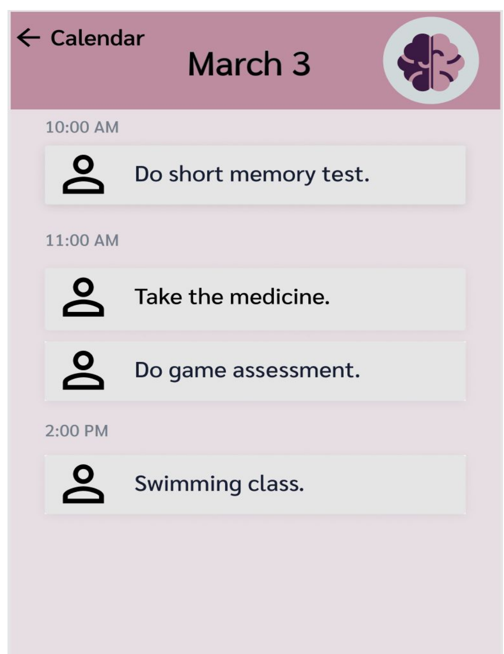


Figure 6 Exercises

Figure 7 Notification

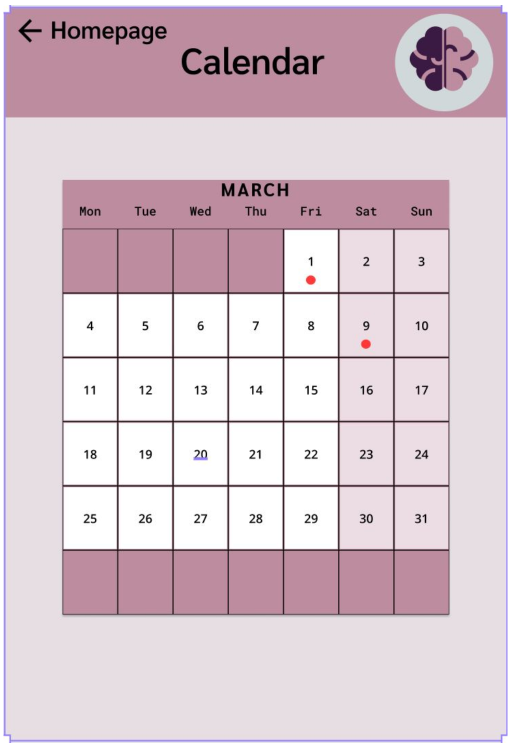


Figure 8 Monthly Calendar

Figure 9 Info Page

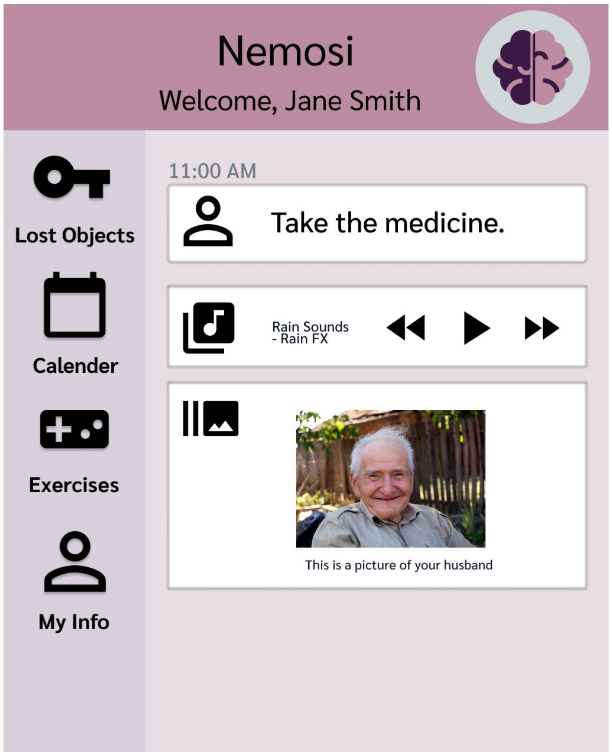
User Feedback

- Dark tiles on backgrounds looks too calm, does not make the text or images within them pop
- Some of the aspect ratios are not consistent with the others
- Would you let a user select which month they see on the calendar? What if they were to get lost?

Iteration 3

Our final prototype frames added more visual fidelity by changing colors slightly.

- This allows for increased perceptual affordance into important tasks.
- The colors of all buttons and notifications was changed to match the “Find it” buttons of the lost objects page.
- After discussing it as a group, it was decided that the patient would not be able to change the calendar month, as this could lead to confusion.



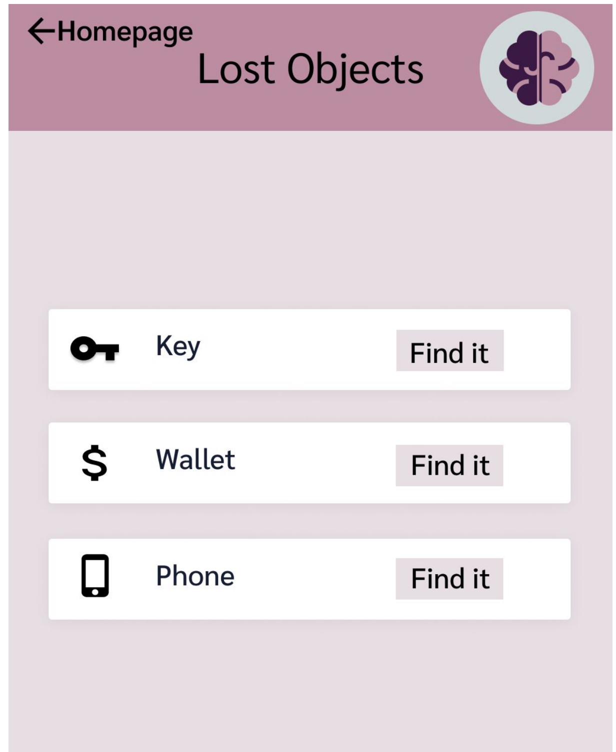


Figure 10 Home Page V3 Figure 11 Lost Objects V3

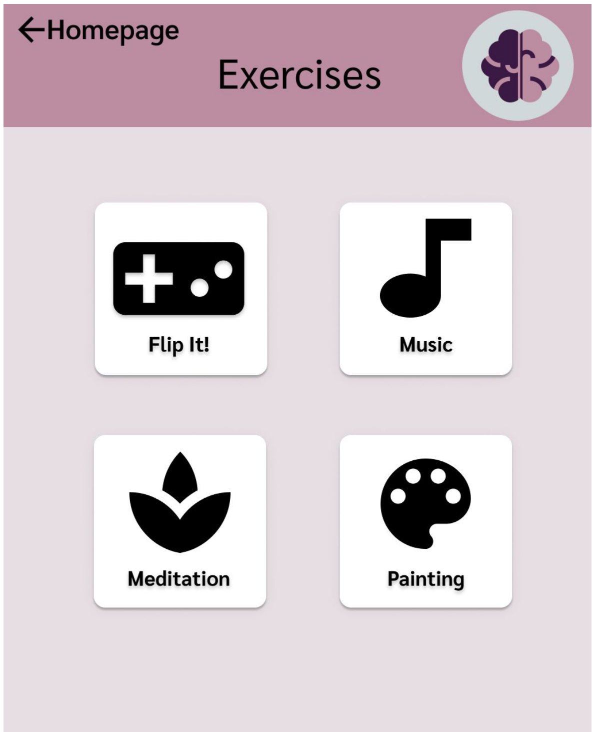
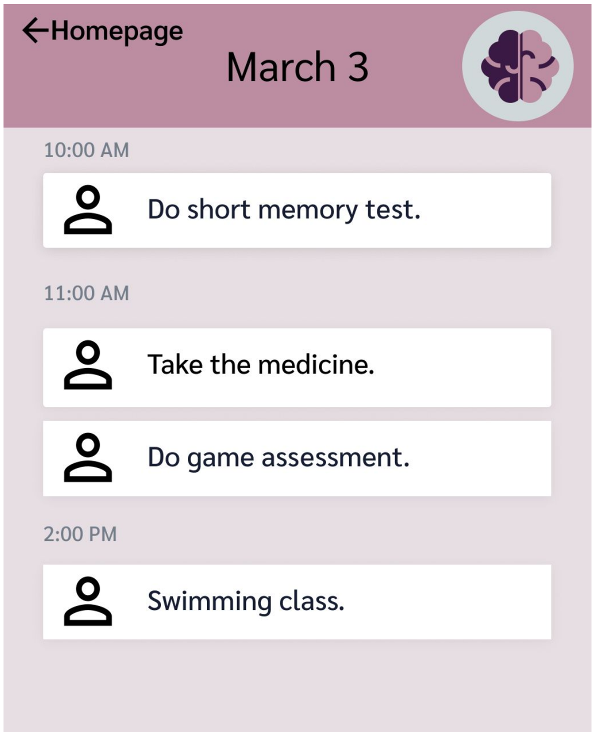


Figure 12 Games V3

Figure 13 Info Page V3



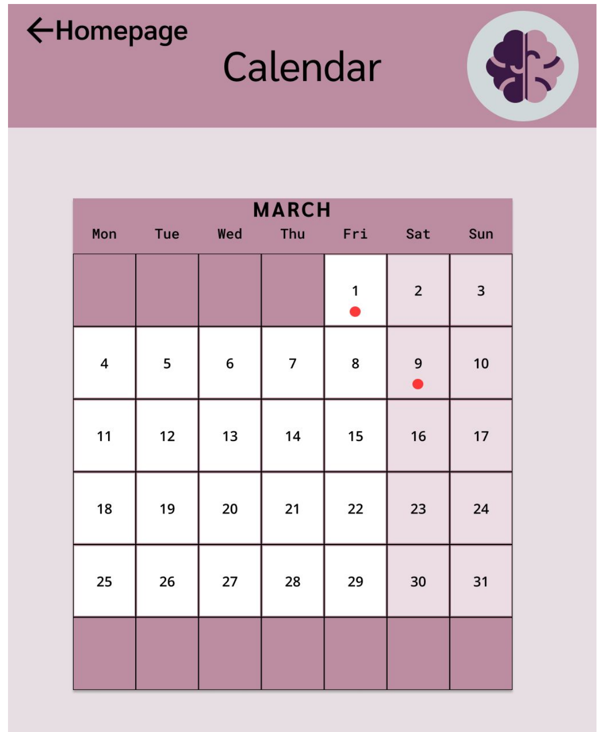


Figure 14 Daily Calendar Page V3

Figure 15 Monthly Calendar Page V3

Iteration 4

We built the Android Application based on the previous iterations, but we were not building the "Lost Objects" page, which would be built by the Short Term Memory group. This is a baseline application and future work will be focused on the styling of the app and useability.

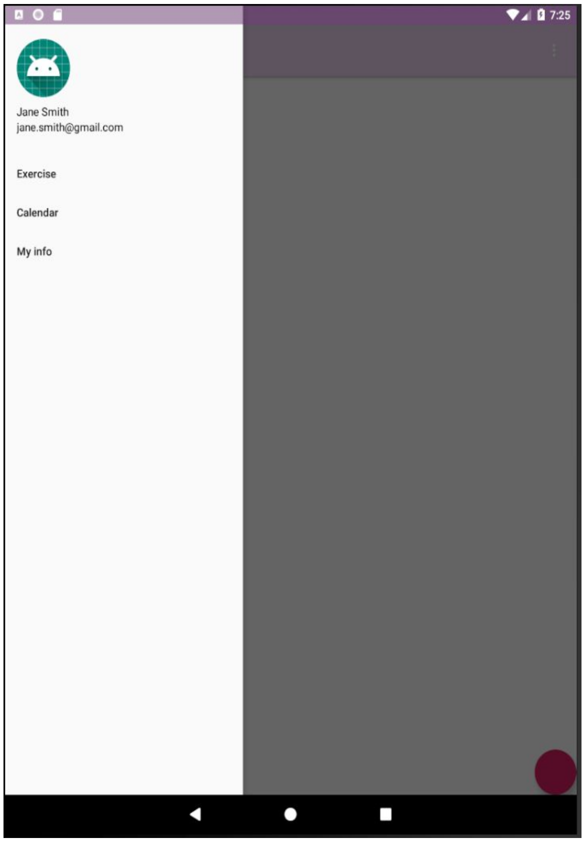
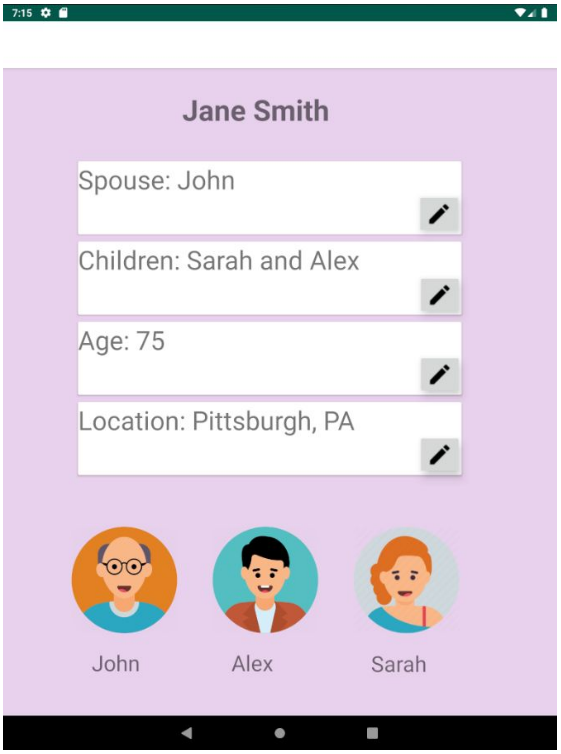


Figure 16 Home page w/ navigation bar V4

Figure 17 Games V4



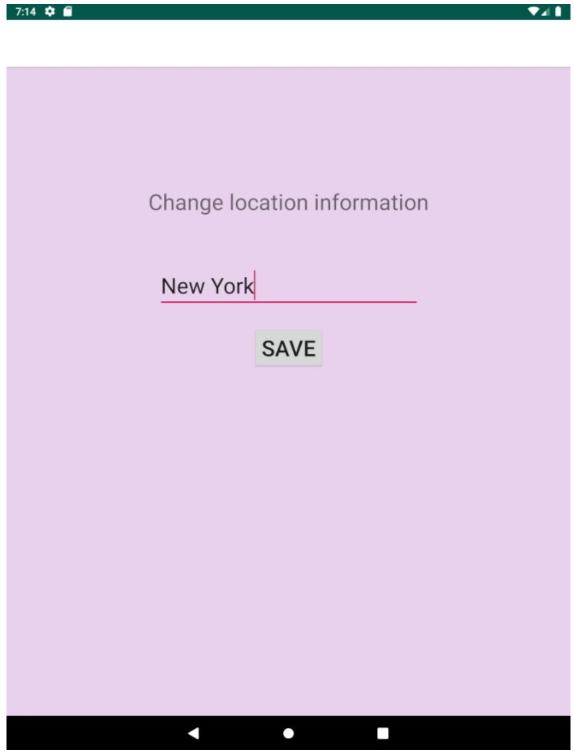
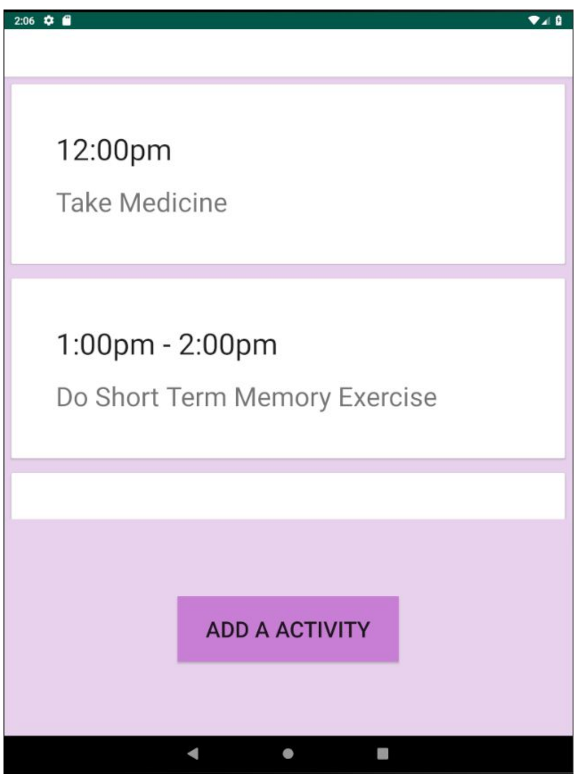


Figure 18 Info Page V4

Figure 19 Edit Info V4



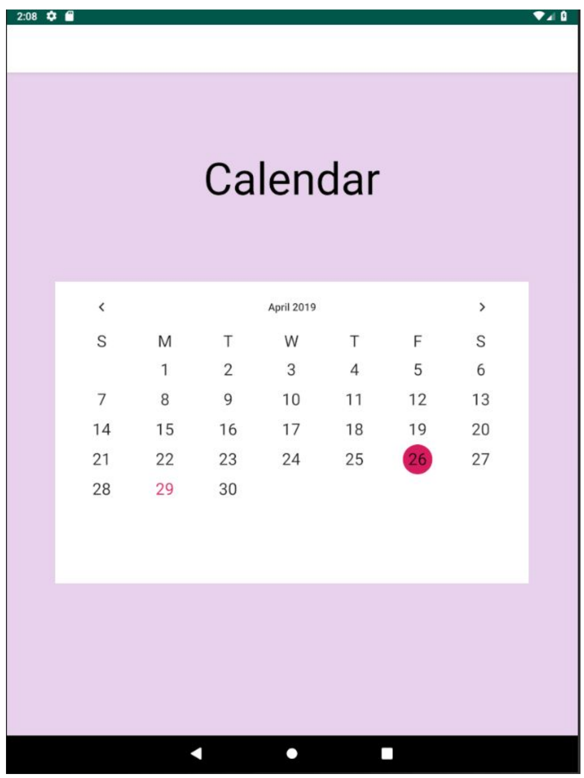


Figure 20 Daily Calendar Page V4 Figure 21 Monthly Calendar Page V4

3.8.3 Hardware Architecture

There is only one hardware component required to perform our functionality: an Android Tablet device. The device we have selected is the Samsung Galaxy Tab.

3.8.4 Hardware Modules and Status

There are no specific hardware sub-units or modules for our team.

3.8.5 Software Architecture

In concert with the Short Term Memory Tests group, we are developing an Android application that will run on 8.0, “Oreo”. Our portion of the development will focus on the user interface, with a few database stores for patient information. STM Tests and games will attach functionality to our interface, and will in turn interface with the server via RESTful APIS and Bluetooth.

3.8.6 Software Modules and Status

Our core tablet application mainly consists of the the following components: Home page, Calendar and Notification Service, Personal Info page and Exercises.

Home Page is a direction page which shows some frequently use functionalities and will direct user to a specific service.

Calendar and Notification Service will mainly be implemented by our group which can display patients daily activities and remind patients in case they may forget.

Exercises will be implemented in cooperation with Games and Assessment Team. They will design and develop several game or assessment app to help patients, and our exercises page will direct patients to the right application.

Personal Info Page will act as a continuously reminder to patients about their personal information like who is his/her wife/husband, who are his/her children, where is his/her house, etc.

3.8.7 List of Components, cost, power

Tablet device: Samsung Galaxy Tab A 8.0”, Black (Verizon)

Cost: $249.99

Power: Music Play Time: Up to 136 hours

Video Play Time: Up to 13 hours

Internet Use Time: Up to 12 hours

3.9 Watch

3.9.1 Functionality

3.9.1.1 Built-in GPS and continuous heart rate

This watch could monitor patient heart rate and track his workout routes without his phone. Persistent heart rate monitor works continuously with high precision. The patient can view his heart rate graph for the last 6 hours. Resting heart rate monitoring is also supported. HUAWEI WATCH 2 motivates user all the way during a workout, showing user goal achieving progress, lap/time/distance reminders, and even speed guidance. Enjoy wandering or cycling with live GPS tracking. The patient can view a real-time map and trail on the high-resolution display. At its core, the Watch 2 comes with a gyroscope, accelerometer, GPS, barometer, digital compass. So, when the user falls down from the bed or the stair, Watch 2 could sense this condition by gyroscope and send the notification to the caregiver.

3.9.1.2 Scientific sleep tracking

HUAWEI TruSleep keeps track of light, deep and rapid eye movement (REM) sleep stages to make your sleep truly restful. This sleep tracking technologies are used for analyzing user sleep patterns and are considered an alternative diagnostics for sleeping pills. Huawei TruSleep can accurately analyze deep sleep, light sleep, REM sleep, and waking to provide patients with the sleep quality evaluations and recommendations authorized by the Center for Dynamical Biomarkers, which is a major teaching hospital of Harvard Medical School. The patient can see in the HUAWEI TruSleep, how accurately it tells the doctor about patients sleeping duration and various modes of our sleep on the daily/weekly/monthly/yearly basis. This data will help the doctor to take necessary measures before it’s too late. The above picture provides you in-depth inquiry of each sleep mode. The doctor can figure it out the problem area of the patient’s sleep.
Moreover, the user can do yoga, exercise, timely sleeping or change user work-life balance to bring back user sleepless nights on the track again.

As for sleep quality detection, the watch could collect light, audio and movement data from the patient every two minutes. We knew that these kinds of information and parameter are closely related to sleep quality. Every the application read these data from the watch, it will write these bunch of data into a local file. In the application we built, it will run a sleep algorithm that calculates sleep quality based on the accumulated data on this file. In regard to the level of sleep quality, it will evaluate the sleep quality in three different levels, including -1, 0, 1. In these various levels, they are related to specific actual sleep time. The actual sleep time means the time of the patient in a deep sleeping status per day. If the actual sleep time is longer than 7 hours, the application would determine the level of sleep quality is 1.

3.9.1.3 Power and notification

The user can optimize power consumption settings to get the most out of his device. The 4G model of HUAWEI WATCH 2 supports either SIM or eSIM card, making patient's 4G-connected even without his phone. Get calls and messages wherever he goes. Double 4G, GPS, NFC, Wi-Fi, and a Bluetooth antenna are integrated under the watch bezel. This reduces signal interference from the user's body and optimizes signal reception. As a result, this watch could help to notify the patient to communicate with the caregiver.

3.9.1.4 Fall detection

As for fall detection, there are accelerator and gyroscope installed in the Huawei watch. Based on the data collected from the accelerator and gyroscope, the application could get real-time patient movement data from the watch. If a falling signal from the watch was detected, the watch will automatically run the fall detection algorithm. After that, the application dynamically transfers the falling signal to the server side. As a result, the doctor could notice the patient's falling condition by the monitor.

3.9.2 Tests and Pictures

Figure 3.9.2.1 Huawei Watch 2.

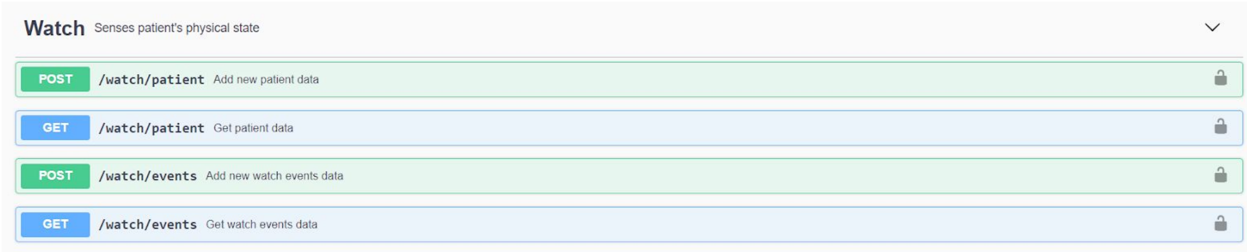


Figure 3.9.2.2 API of the Huawei Watch 2.

Figure 3.9.2.3 Huawei Watch 2 data Json form.

3.9.3 User Testing/Experimental Evaluation

To meet our needs, the Huawei Watch 2 must collect four main types of data:

1. Heart rate
2. Sleep
3. Accelerometer and gyroscope
4. GPS

We have tested the watch for all four of these requirements by wearing it for one day and observing the data it collects. Below is a summary of the tests we have conducted.

3.9.3.1 Heart rate

The Huawei Watch 2 tracks minute-to-minute heart rate data. The user can view his or her current heart rate directly through the watch as shown in Figure 1 below.

Figure 3.9.3.1 Instantaneous heart rate reading as shown on the Huawei Watch 2.

The watch's companion app, Huawei Health, allows the user to see overall trends in the heart rate data collected. Although we do not plan to use Huawei Health in the final version of the Nemosi system, we have used it in the evaluation phase to display collected data. Figure 2 shows a screenshot of the heart rate data collected in one morning by the Huawei Watch 2.

Figure 3.9.3.2 Heart rate trend data as shown in the Huawei Health app.

3.9.3.2 Sleep

The watch uses Huawei's own TruSleep technology to track a user's sleep patterns. Sleep data can also be viewed through the Huawei Health app. These data include a time series of the stages of the user's sleep (Figure 3), Huawei's sleep score for a given night (Figure 4), and a classification of each stage of sleep as normal, high, or low (Figure 5).

Figure 3.9.3.3 Time series of user's sleep stages.

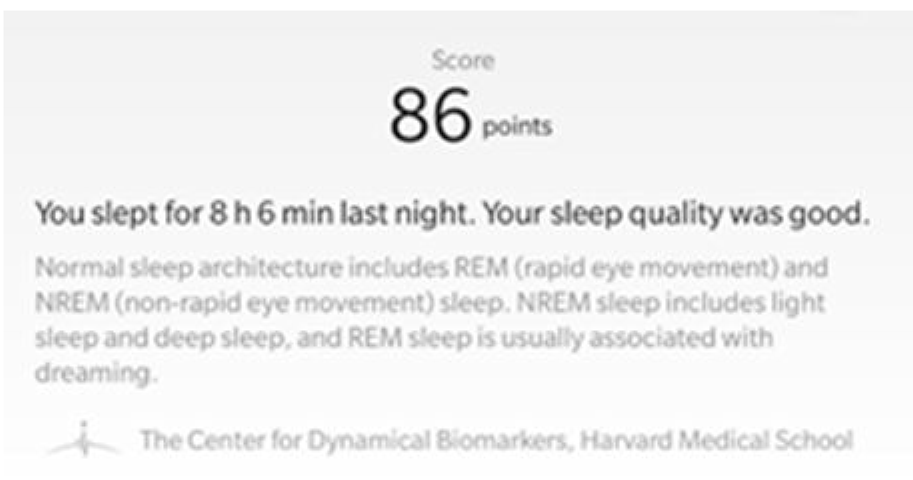


Figure 3.9.3.4 Huawei's TruSleep sleep score.

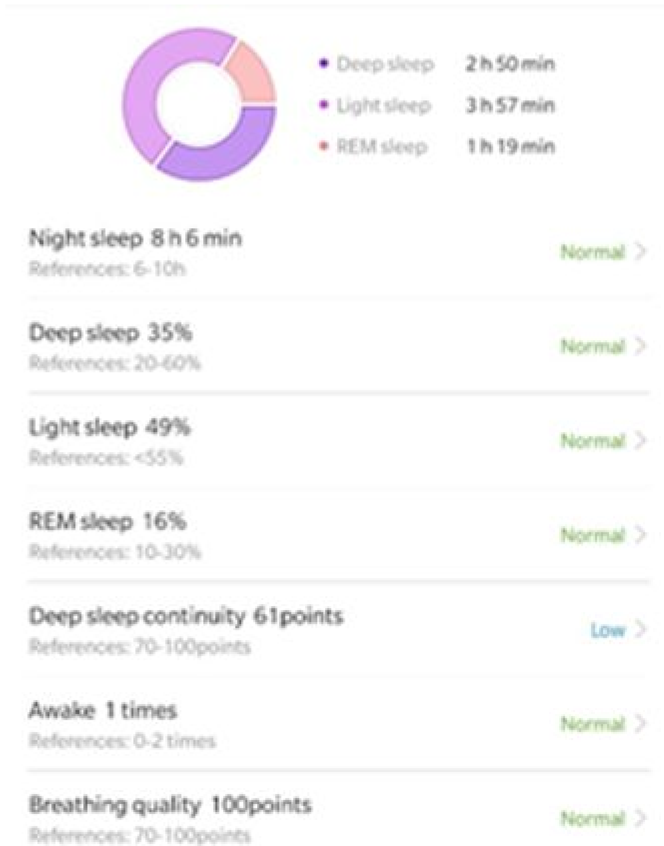


Figure 3.9.3.5 Sleep stage breakdown.

3.9.3.3 Accelerometer and Gyroscope

To collect accelerometer and gyroscope measurements from the watch, we used the Sensor Box app available for Wear OS from the Google Play Store. This app visualizes accelerometer and gyroscope measurements in real time.

Figure 6 shows the accelerometer reading as the watch is rotated toward, away from, and again toward the user. Figure 7 shows the watch being rotated left, right, left, then right.

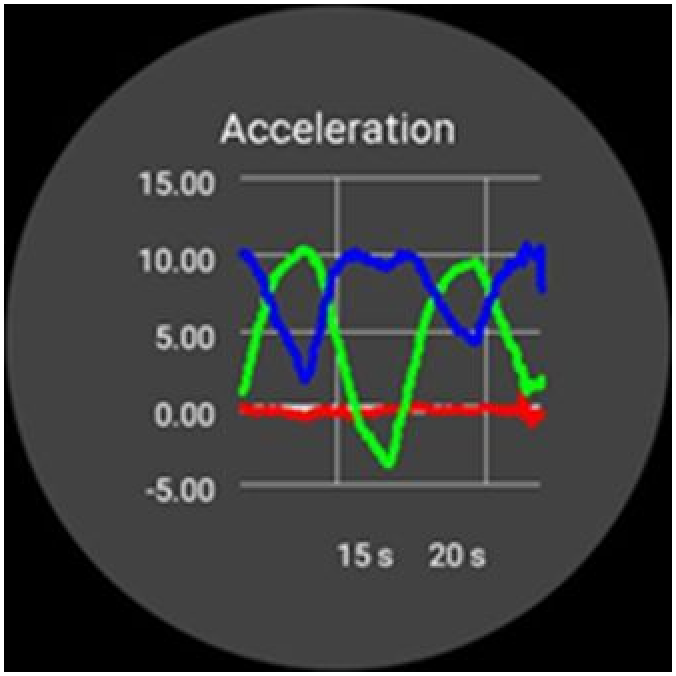


Figure 3.9.3.6 Accelerometer reading as watch is rotated toward and away.

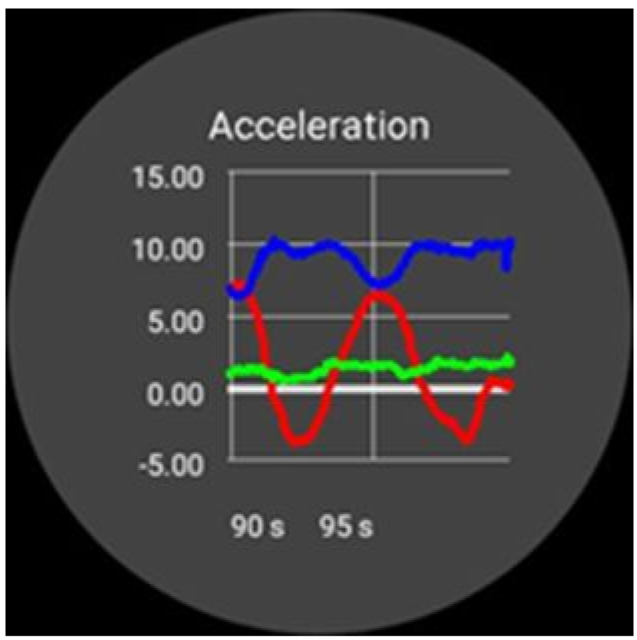


Figure 3.9.3.7 Accelerometer reading as watch is rotated left and right.

Figure 8 shows the gyroscope reading as the watch is tilted along each of its three axes. Roll is shown in green, pitch in red, and yaw in blue.

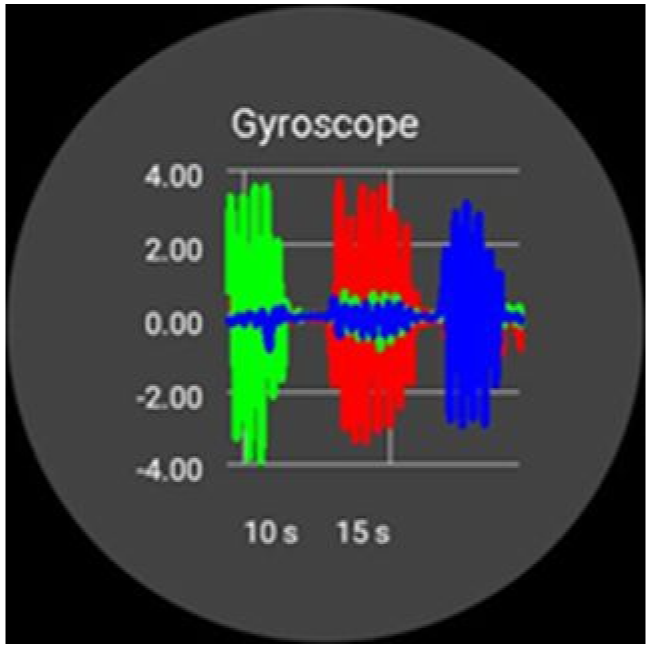


Figure 3.9.3.8 Gyroscope reading for roll (green), pitch (red), and yaw (blue).

To more accurately simulate the data from our expected use cases, accelerometer and gyroscope readings were also taken during a fall. These readings are given in Figure 9 and Figure 10 below.

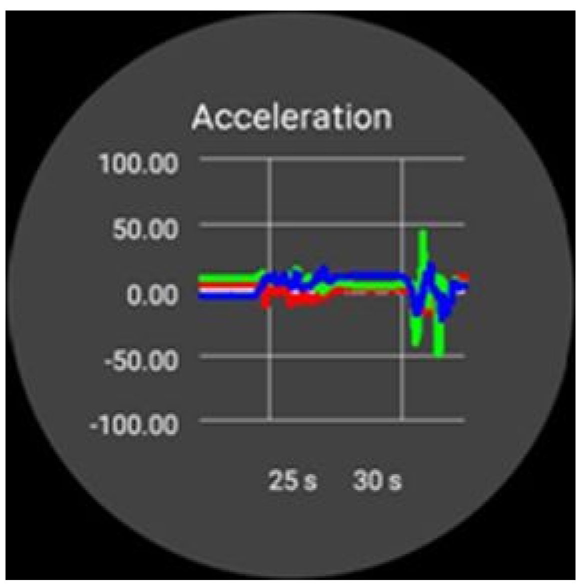


Figure 3.9.3.9 Accelerometer reading during a fall.

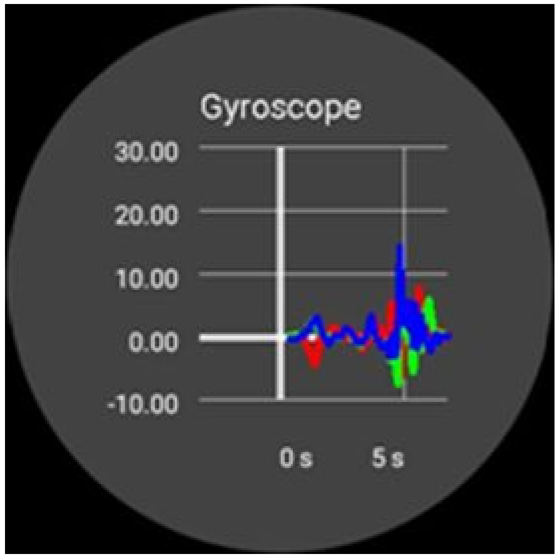


Figure 3.9.3.10 Gyroscope reading during a fall.

3.9.3.4 GPS

Finally, the watch's GPS tracking capabilities were tested. Figure 11 below is a screenshot from the Google Fit app, another health-tracking app native to the watch. It shows a walk from Carnegie Mellon's Hunt Library to its East Campus parking garage.

Figure 3.9.3.11 GPS tracking data displayed in Google Fit.

3.9.4 Software Architecture

We want to implement the following features in the watch:

1. Stress detection.
2. Sleep evaluation

3. Fall detection
4. Implement push notifications.
5. SOS signal

We shall be programming in WatchOS which is a variant of Android. For our software architecture, we shall be following an **MVP pattern**.

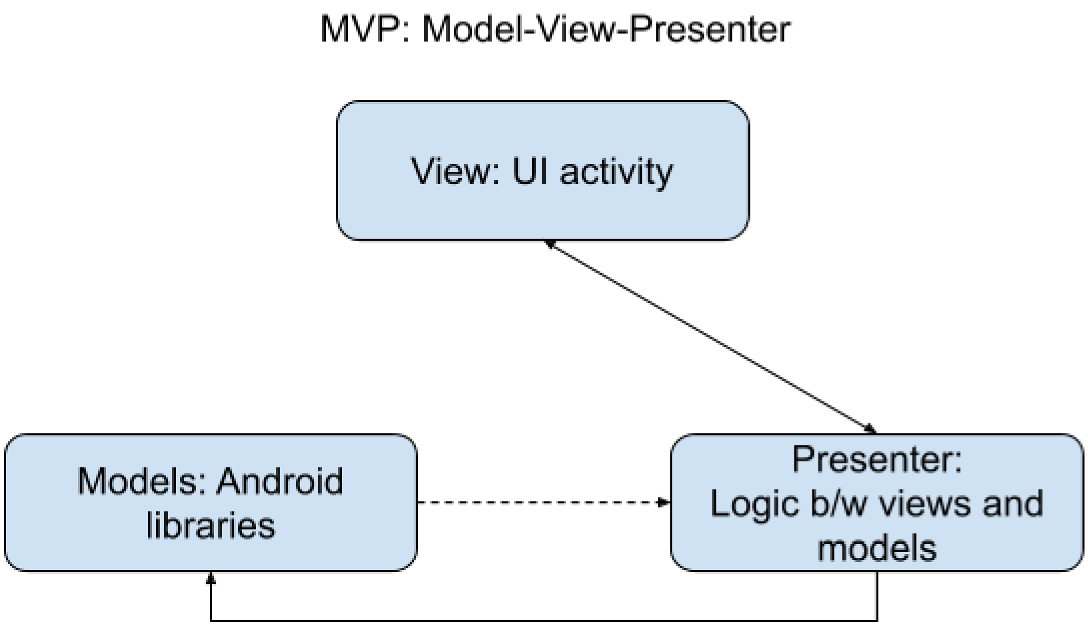


Figure 3.9.4.1 The Model-View-Presenter

While the presenter and view are pretty trivial to understand, the models necessitates a deeper dive. The following diagram illustrates the models we shall be using.

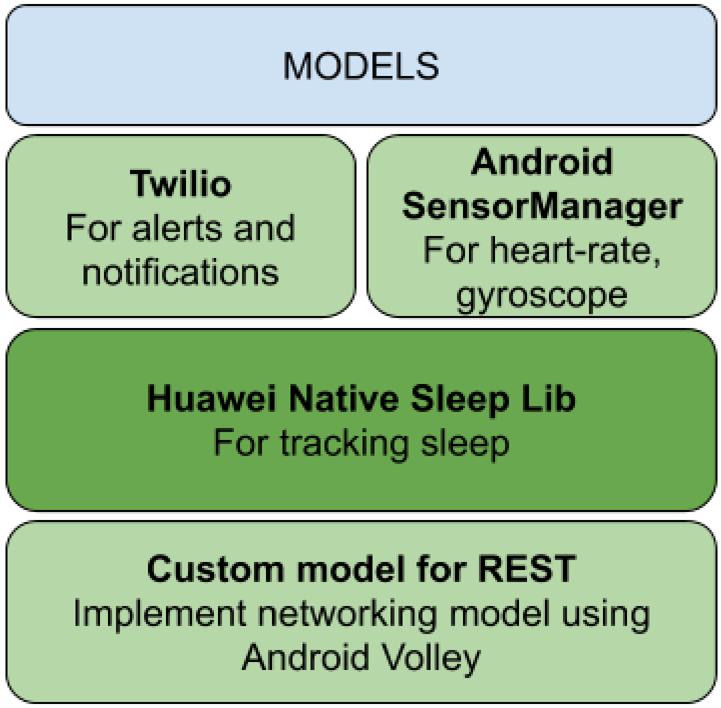


Figure 3.9.4.2 MVP details

3.9.5 Software Modules and Status

We now have access to the specific APIs and libraries we need to implement the desired features. We have communicated with the OS/DB team to determine the API and the data format. We have received the watch and begun development of the Watch software. We are currently on schedule to meeting all of our goals for the project. We have not encountered any substantial issues that might delay our delivery of the Watch system. I know which specific APIs we will need to use to gather the sensory data. I also found code examples that use those APIs to gather the appropriate data. We also realized that we need to be able to “fake” some data for the simulation since sleeping needs to be simulated for a part of the presentation. Additionally, being able to “fake” data will help make debugging easier since it is a form of unit testing. We implemented sleep detection by looking at the watch’s light and movement level. We found an algorithm that would take this data and calculated the sleep quality of the user. We implemented fall detection by looking at the accelerometer and measuring the acceleration of the watch. If it meets a certain threshold, the watch will make a network request to the sever with the fall time with JSON. The watch also periodically sends the wearer’s heart rate to the server.

3.9.6 List of components, cost, power

Components	Material	Cost	Power
Watch Case	Plastic	$300	420 mAh* 410 mAh* (minimum value) *Typical value. Actual capacity may vary slightly. This capacity is the nominal battery capacity. The actual battery capacity for each individual product depends on network configuration and many other factors. Actual results may vary.
Watch Strap	Strap is replaceable with a lug width of 20mm and suitable for a wrist size of 140-210mm		
Display	1.2-inch circular AMOLED display 390x390 pixels with a PPI of 326 Corning Gorilla Glass		
CPU	Qualcomm Snapdragon 2100		
Operating System	Android Wear 2.0 Supported Mobile OS: Android 4.4+, OS 9.0+		
Memory	4 GB Flash+768 MB RAM		
Specaker, NFC	Supported		
Sensors	6-axis A+G sensor 3-axis Compass Heart Rate Sensor (PPG) Barometer Capacitive Sensor Ambient Light Sensor		
Water and Dust Resistance Rating	IP68		

Since there is no specific cost and power consumption for each components, above list just give the overall cost and power.

3.10 Conclusions

3.10.1 Summary of Key Design Issues

Caregiver & Doctor Interactions Team:

- Due to the nature of this class, the design of the interactions and screens had to constantly be changed due to changes in the work of the sensor teams. Although the initial design informed the work of the sensor teams, the success and progress of the sensor teams ultimately determined what features would be available and how they would be interacted with. Many features had to be dropped due to time limitations while others due to other resource limitations.

Core Tablet Team:

- One of the major design issues faced by the team, involved having the entire team work on a single UX/UI Platform. It was hard to find a platform where we could incorporate all the members to work simultaneously.
- While working on the UX/UI Design, the team had to work on creating a single template for the pages of the application and ensure all the pages had the same specifications and basic design.
- As we moved towards working on the coding of the application, the team faced design issues related to creating the application on Android studio and writing the code for the application.
- The team had to work towards creating a high fidelity application and refine all the design aspects of the UX/UI Design and create a more accessible and usable application.
- The UX/UI Design had to be refined by adding the proper icons, buttons, interactions, toolbars, and patient information to ensure a more user friendly application.

Core Analytics Team:

- Patient falls data visualization not finished
- Our progress was constantly behind schedule in phase 3 because of we were dependent on other teams for raw data.
- No caregiver or doctor data retrieved from wandering team.
- Many physical measurements and healthcare analysis were researched in phase 2, but very few measurements can be collected from watch team in phase 3.

Short Term Memory Team:

- The database api expects a certain format which isn't ideal for our contribution. As a result, we send data back from our daily assessment survey in a format which isn't space efficient.
- Both the Tile trackers and Flic button use bluetooth to connect to the tablet. Bluetooth has limited range which makes finding objects far away impossible. Our setup only works well for an in home scenario.

- The Tile api is closed and we are not able to directly interface with the Tile trackers. As a result, we instead make calls to the official Tile app within our app. This requires the patient to also have the Tile app installed on the tablet.
- The Flic button is dependent on the Flic API. Changes or issues in this API could affect our product.

Watch Team:

- The first issue we encountered is first how to set up the connection between watch and the Android PC without having line connection, so to solve this, we use the WiFi connection to make sure that data capturing from watch can be directly sent to the PC and database for further uses.
- The second issue we had to deal with is how the falling detection that the sensor data type may be too complex to be readable. So we have done many experiments such as falling down from left, right, head and back to get a threshold of falling down comparing to sitting or laying down.
- The third one would be the sleep quality sensing that needs to use light, audio and time parameters. At first we managed to store those data together in a file but then we found out that the data usage may be too large for every seconds, so we tried to use every one or five minutes, which worked for further analysis.

3.10.2 Lesson Learned

Caregiver & Doctor Interactions Team:

- In the last phase of this project, I learned not only how to program frontend interfaces, but also how to operate a technical team which involves frontend, backend, and sensors (data source). The communication between teams becomes really essential when the size of the development group exceeds ten people. How to recruit more developers when our team is short of coders (originally only 3 people in my team can code), how to divide up the work to fit with everyone's expertise, how to maintain the communication with other backend, analytics, and sensors team to make sure their data schema aligns with ours. The total amount of work actually way exceeded the hours I spent on coding. This project experience lets me start to think what it is like to work in an industry setting where many engineering teams need to collaborate on the same project.
- In this phase, I learned lots of technical details, including how to get access to git, how to pull, push or commit my own document, how to implement responsive website (html, css, javascript) using react.js library and using bootstrap to simplify css structures. Moreover, I learned how to cooperate with other teams. Previously, our collaborations with other teams are more logical, but now we have to merge our code together which often requires lots of patients and communication between groups. For example, we

relied on core analytics group to help us implement the data visualization graphs, then both of us relied on functional backend server to provide us with useful data. Server/database relied on sensor teams to provide data as well.

- In phase 3, I was the front end developer as well as coordinator for caregiver's app. On one hand, I learnt how to cooperate with developers within our team (some of them are from other groups in phase 2). For example, I wrote static website and handed it to my teammates to transform it into React JS. On the other hand, as I participated in app design in last phase and familiar with feature details, I helped coordinate the progress of demo implementation. For instance, before the first demo time, there were lots of work gap between front end team and server/database team, but the communication on kiva is of low efficient. I formed a Messenger group for both group in order to accelerate the sprint and it worked.

Core Analytics Team:

- Annan Ma: Getting the individual parts working is easy, but integrating all working parts and keep them working correctly is hard.
- Jennifer Liu: Team work, communication skills, database programming skills, prototyping in a short period of time, learning what the patient really needs.
- Luo Yu: Team work, database using skills, some data processing methods. Knowledge about hardware platform.
- Wei Ling Chang: Team work, data analysis programming skills, connect engineering with clients, prototyping a system or product in short time.
- Xiaoyu Qiu: Team work, task management skills among teams, communication, linux and database programming, rapid prototyping skills.
- Yuling Liu: Team work, working with clients, rapid prototyping skills.

Game assessment team:

- Between combining work from individuals from the team and integrating software components across teams, proper usage of Git is vital to successfully build a large scale group project.
- Actual implementation of designs from Phase 2 can change rapidly due to unforeseen problems and new solutions. It is important to keep track of any architectural changes within the project through communication with other teams.
- The ability of learning a new technique is significant and necessary for group members. Since time is limited, everyone should set about implementing as soon as possible to keep pace with other groups. Though Android Studio is new for our team, we put efforts into studying it and finished the application on time. The communication with other groups is also essential. We have to make sure that all our implementation fulfills other groups' requirements.
- The games were not tested by enough users due to time limit and thus have some problems in user experience. The process could have been: demo page of the game →

user experience feedback on design → improved design → full game. This way the feedback on user testing could be included in the final product.

- The communications among our team, the Core Tablet and Short Term Memory teams were not sufficient enough to have a smooth integration of our work. Each team is dependent on each other and we should have reached a consensus on the work hierarchy early on in the implementation phase.
- Merging a platform into our games App without knowledge of its API is not an easy task, and by completing this, I learned not only techniques such as Java, Android development and git, but also the method to fix problems and the courage to get rid of difficulties.
- Working with group members could save a lot of time to complete our work, different people have different strengths, and it may be much better to combine our power together rather than do it alone.
- Due to time constraints, buying games appeared to be the most feasible option and we were convinced to do so but after researching on score evaluation and merging of all games, the disadvantages of this option outweighed its advantages. Building games can be easier for the sake of obtaining and storing scores, adding personalized components to these assessments and integrating the various games related to different assessments into one interface. Buying them wouldn't have sufficed these goals.
- As game and assessment group, we need frequently communicate with a lot of groups. Due to limited time the communication among the teams is still not enough. There will still be some mismatch especially in database and data analytics team. The data they need and the data we can create are very messy and every team spent more time on working their own work.
- Also, our output data and its format will be important to analytics team to generate the analysis and pattern of patients. After confirming the system, language and output, we can move to next stage to research on the game or assessment we will build or buy. We should continue keep other teams updated on what stage we are and corporate more for the future testing and implementing.

Wandering Tracker:

- In this iteration, a well-designed CAD could not be completed as learning CAD was slow and there were time constraints for designing and testing. If this design (with these boards) were to be used, a decent case that could fit into the shoe would be necessary to manufacture. The much smaller CAD circuit could essentially fit inside the heel of a shoe or beneath the sole. This would make our design more compact and sturdier. Compared to our current design, the entire architecture would just disappear within the shoe and would not be visible to the outside world.
- The Global System for Mobile (GSM) we used for positioning of this design iteration was highly lacking in accuracy. In future models, we would like to be able to use the actual GPS module that is also included with the FONA but that we had problems accessing. This would give us more accurate data for tracking when the patient is wandering.

- This design, however, was just a prototype. Ideally, we would eliminate the Commercial Off-the-Shelf (COTS) microcontroller boards and custom design a single board computer (SBC) with all the same functionality.
- Since we used open-source COTS products for this design, the schematics could be used to create our own Printed Circuit Board (PCB). We could use the microcontrollers onboard the Adafruit Flora and FONA boards and the individual I/O connections to make an SBC of minimal size.
- Going further, we could use a smaller microcontroller or even nano-technology that only has very basic functionality to get the circuit down to the size of a sticker. This creates the potential to attach the device to anything the patient takes with them, whether it be a purse, a jacket, etc.
- There is also current research at Carnegie Mellon University currently, that looks to make small electronics into tattoos. If we can make this product into a tattoo, we could guarantee the patient never leaves it behind, like they may if they wear different shoes one day or forget their watch.
- The hardest technological challenge with either of these two options is power. How can we standardize charging if the product is not a shoe that has a specific mat? How can we charge a tattoo? Or can we somehow make the product cheap enough that the tattoo can be replaced every week or so? These are questions that would need to be answered if the wandering device were to be shrunk for alternate placement.

Core Tablet Team:

- Working as a team to create the patient interface, we learned how to effectively collaborate through a design process.
- The team began by working separately and then realized it was much easier to create the wireframes together to keep the vision consistent.
- Our team also learned more about visual design principles, including consistent navigation, fonts, and color schemes. To have a uniform visual design among our pages and with the doctor/caregiver interactions team, we learned to follow a style guide.
- The members of our team who developed the Hello World prototype learned how to create an Android app with Java.
- The team had to work on splitting the code amongst the members and coding separately and coordinating among ourselves to ensure the application was consistent through its flow.
- We all learned how to collaborate with other teams to determine how to divide tasks and integrate separate parts into one product.
- The team also learned about how different groups interact and how to effectively present our queries and concerns with the other groups. We had to collaborate with other groups and integrate the work into one cohesive application.

Short Term Memory Team:

- Between combining work from individuals from the team and integrating software components across teams, proper usage of Git is vital to successfully build a large scale group project.
- Public APIs do not always have the full functionality that a company has access to
    - For the Tile, the public API had the ability to find locations but not the ability to make the Tile devices ring
    - This prevents control of the Tile from outside the app
- Companies restrict cross-program functionality for security reasons
    - Google allows search queries from other programs, but it won't allow assistant queries due to security reasons
    - This also prevents control of the Tile from outside the app

Home Sensors Team:

- A stronger knowledge about data analytics will allow us to provide more comprehensive information on patient activities and thus better help the patients.
- A good workflow will allow teams to integrate fast and often, and will also allow teams to discover bugs early.
- Integration with other teams and effective communication was one of the most important lessons learned in the design of Nemosi.
- Hardware can run into unexpected issues anytime and the ones working with them must be ready to make alternate provisions.
- While working to implement a huge system, one needs to ensure that every link is working correctly.

Watch Team:

- Communication between different teams is important. What we got is the data from the watch sensors which needs to send to the database team so that we need to communicate with them what kind of the data and how they are represented.
- The Git tool is really useful for the group work since we divided our implementation in different parts and for the sleep quality analysis, we need data from light, audio and time data.
- The data type should be readable after some data transformation rather than the raw data from the sensor itself.

4. Project Management

4.1 Task Dependency Chart

- Patient and Doctor interaction

Groups	Provide to us	Request from us
Core analytics	- Average heart rate, blood pressure during each incident - Visualization of patient trends	- List of incidents uploaded by caregiver - Time range of each incident
Server/Database	Doctor’s dashboard: - List of appointments (date, time, patient) - List of patients - Patient profile (name, gender, age, doctor’s note, medication, appointments) - Patient trends data (exercise, cognitive assessment result, sleep quality, night movement, injuries) - Incident log (time, type, heart rate, blood pressure, caregiver notes) Caregiver’s dashboard: - Patient schedule - Patient real-time location - Real-time alert (when detects falls, wandering outside, mood swings) - Patient exercise - Patient sleep (hours of deep sleep, hours of light sleep, # of times woke up, # of times used bathroom, # of times got out off bed)	- Patient schedule (uploaded by caregiver) - Family members and photos - Incident log and notes

	- Patient’s night movement (frequency went to each room)	
Core tablet		- Patient schedule - Family member info and photos
STM test	- Memory assessment results	
Watch	- Sleep quality (hours of deep sleep, hours of light sleep) - Vitals (heart rate, blood pressure) - Falls	
Home sensor	- Night movement (# of times got out of bed, # of times used bathroom, # of times exit the main door during night) -Sleep Monitoring	
Games/Assessments	- Cognitive assessment results	
Wandering tracker	- Real time location tracking	

- Core Analytics

Groups	Provide to us	Request from us
Patient and doctor interaction		- Data analysis results
Server/Database	Raw data including - Physical data - Mental data - Wandering data - Action data	
Core tablet		

STM test	- Short term memory test results (mental data)	
Watch	- Physical data - Action data	
Home sensor	- Physical data	
Games/Assessments	- Cognitive assessment results (mental data)	
Wandering tracker	- Real time location tracking (Wandering data)	

- Core Tablet

Groups	Provide to us	Request from us
STM test	APIs to control Tile Bluetooth Tracker.	Database group gets Daily Assessment Survey results
Games/Assessments	Game and Assessment app.	Directions to their app.
Doctor and Caregiver interaction	Patient Schedule and Patient Information entered by the caregiver.	Data from the tablet is sent to the doctor dashboard.

4.2 Issue tracking

Group Name	Issue Status (Resolved/Open/Not Valid Anymore)	Issue Category (if any)	Issue Description	Members Responsible	Any Comments
Wandering	Open	Architecture	No existing method	Eli Yu, Rajat	need to work with

Tracker			in architecture to get Caregiver location	Mehndiratta	Caregiver Interactions team
Home Sensors	Resolved		Problem in using MUX	Priyanshi Jain, Jhanak Johri	
Wandering Tracker	Resolved	Parts	Need to order new SIM card	Rajat Mehndiratta	priority lowered because current solution works
Wandering Tracker	Resolved	Parts	Need to order new battery	Rajat Mehndiratta	priority lowered because current solution works
Wandering Tracker	Resolved	Hardware	Unable to load code onto Adafruit Flora	Sharan Turlapati	Figured out a hack to upload the code. Pressing the reset button twice during upload does the trick
Wandering Tracker	Resolved	Software	Adafruit FONA network initialization issues		can just throw in a workaround in our code
Wandering Tracker	Resolved	Software	Code needs to go on the git server	Rajat Mehndiratta, Sharan Turlapati	
Wandering Tracker	Open	Software	No interface or method exists for storing geo-zones		can just hardcode on the server
Wandering Tracker	Resolved	Software	Integrate GSM code with timer and sleep code	Sharan Turlapati	WIP, has some compile-time errors (per Sharan)
Caregiver Interactions	Open	Frontend Web	Migrating frontend caregiver Django code to React	Eli Yu, Xuliang	
Wandering Tracker	Resolved	Hardware	Build the circuit inside of a shoe	Raewyn Duvall, entire team	Raewyn has figured out a good bit of this
Home Sensors	Resolved	Software	Build a json/http module to communicate w/ the DB server	Justin Chu	No encryption for now
Wandering Tracker	Resolved	Software	Unable to link FONA library codes	Sharan Turlapati	The library package wasnt installed on the local machine
Short Term Memory	Resolved	Software	Unable to ring tile using classwide	Manu Gopakumar	Limits integration with classwide app and

Sensors			app		prevents functionality for scheduled ringing and physical button ringing
Doctor & Caregiver Interactions	Open	Software	Need GraphQL endpoint from Server/DB team (We cannot start connecting the frontend with real data without the specifications of data type from GraphQL endpoint)	Anyi Diao	We need to know the format of data we will get from server in order to implement the fetch functions. Core Analytics team is also requesting the format of visualization related data in order to create graphics.
Wandering Tracker	Resolved	Hardware	We apparently can't get an ammeter to work; would be useful to be able to test inductive charging with an ammeter...	Rajat Mehndiratta, Eli Yu	
Server/Database	Resolved	Software	Working on GraphQL endpoints	Wenting Chang	
Wandering Tracker	Resolved	Hardware	TING SIM card not yet verified working	Rajat Mehndiratta, Sharan Turlapati	backup: just use rajat's sim card
Wandering Tracker	Resolved	Software	HTTP/server request support		
Wandering Tracker	Open	Software	SMS messages adjusting polling frequency		Semi-resolved but not fully implemented (SMS RX pin interrupts)
Watch	Open	Software	Sleep data cannot be extracted from Huawei watch natively	Eryn Hopps	Currently researching alternate sleep data collection methods, including Sleep As Android API
Wandering Tracker	Open	Software	Is there any logic on the server to analyze wandering?		
Core Analytics	Open	Data	No caregiver or doctor data retrieved from	Xiaoyu Qiu	

			wandering team.		

4.3 Suggestions for Improving Class

Over the course of this project, we have came up with the follow suggestions for improving the class:

- Development strategies were disjoint, group specific and unguided. To improve work across groups a unified development methodology should be adopted for the entire class from the beginning.
- Technical work could begin earlier. Phase 1 planning could be more specific to cover many aspects of Phase 2, and Phase 2 could be technical testing and replanning as needed.
- Since Alzheimer is a complicated topic in medical research, it is better to let the doctor show requirements or testing methods, whereby we build a system. Otherwise our implementation is not valuable in practical use.
- The time for implementation and optimization is too short. The key phase is phase 3 while currently time for all phases are almost the same, which is not practical for implementing a good design.
- Since Alzheimer is a complicated topic in medical research, it is better to let the doctor show requirements or testing methods, whereby we build a system. Otherwise our implementation is not valuable in practical use.
- The time for implementation and optimization is too short. The key phase is phase 3 while currently time for all phases are almost the same, which is not practical for implementing a good design.
- The class is so big that the group definitions and inter-group collaborations were very difficult, so some extra monitor and help on communications would be beneficial
- It is important for us to pay more attention on each group's cooperation, it may be better if each group's interaction guy could meet together usually to talk about their own groups' work.
- The scope of this year's project is too big that every team started by going out and experimenting different things, and ended up hard to integrate. I think if we were to try such a big topic in future semesters, the instructors need to make it clear to the HCI team in the beginning that we need a narrowed scope of product features.
- It will be better if the semester can start by everyone filling out a questionnaire about their background (major, fields of expertise -- hardware, frontend, backend, etc.) and top 3 team preferences. The instructors can assign the members of teams based on both personal preference and expertise. This will avoid the situation where one team runs extremely short of people, and some other teams get too many. The questionnaire can also be useful later when there needs to be team shuffling, and instructors can assign students with previous experience to specific teams.

- The in-class discussion seems to be inefficient. Often lots of students are absent, then neither the group meeting nor inter-group communication can work because people that are responsible for certain parts cannot be found. So it's better if attendance is recorded, then everyone can discuss more efficiently.
- I feel it would be better to allocate more time to phase 3 than phase 2. In phase 2, some teams had few work to do, their time was wasted in this period before they learnt the availability from implementation. You don't know whether the device works until you try it. If these hardware teams can start implementing earlier, they would knew the limit of each chosen device and had time to solve it.
- Also, the workload is not fair for everyone, which is resulted from too many people in a group with not much work. For example, 2-3 people in one hardware team would be more efficient than 5-6 people. Students would have enough responsibility for their part in a small team.
- Should introduce more evaluation metrics in team contribution, e.g. peer evaluation.

4.3.1 Alternative Communication Platforms

- Kiva is a nice platform but a file sharing system similar to Google Drive would be great for collaborations on work products
- Kiva is old and obsolete compared to its competition. It is reminiscent of internet forums from the early 2000s which are now dead. We suggest dropping Kiva entirely and replacing it with a more modern system that also has a mobile app.
- Kiva is not the most user-friendly platform for communication, and is prone to failing. I don't know if Slack has all the features in Kiva, but I believe Slack to be a far more well built communication platform for group work.
- Kiva has broken for several days, it is not convenient to get information before in these days, if there are more method to share our works or record our previous works, it will help us a lot.

Throughout all three phases, adoption of Kiva remained a persistent challenge. Due to the perceived user-unfriendliness, unreliability, and insecurity of Kiva, most teams appear to have gravitated toward other platforms. This created a fragmentation of communication in the class, making it challenging to locate and coordinate with other teams or to make activity in the class visible to staff for insights, feedback, and grading. These communication challenges not only made work more difficult but likely hampered the quality of the final product (e.g., by making it hard for the Demo Master to keep track of teams' progress due to a lack of practical avenues for interfacing with teams). The lack of platform adoption fostered communication barriers throughout the entire semester, with critical class communication occurring not only on Kiva but also on at least Slack, Messenger, and WeChat.

Contrary to design, the vast majority of communication- including formal communication- appears to have occurred off-Kiva, much of it on Slack. Moreover, students persistently used Google Drive (and similar platforms) for file-sharing instead of uploading and revising documents on Kiva; these platforms, unlike Kiva, provide source control and facilitate concurrent editing. All phase reports were compiled on Google Docs, as were most documents and diagrams in the course. During Phase 1, we raised intellectual property concerns regarding third-party file-sharing applications; however, Google Drive's Terms of Service indicate our intellectual property will be safe (see Figure 4.3.1) and most widely-adopted file-sharing/collaboration platforms have similar terms, alleviating our concerns.

Your Content. Google Drive allows you to upload, submit, store, send and receive content. You retain ownership of any intellectual property rights that you hold in that content. In short, what belongs to you stays yours.

Figure 4.3.1: Google Drive Terms of Service, with intellectual property protection highlighted

While Kiva evidently represents the culmination of significant research by Dr. Finger and others in conjunction with course staff, Kiva's analytics lose much of their benefit when most critical communication occurs off-Kiva. In contrast, while shifting to Slack would require ceding some platform design decisions, these losses may be mitigated by the benefits of being able to reuse existing, tested, and popular platforms. As with Kiva, Slack administrators can even restrict users' ability to edit or delete messages. However, additional work would be required to incorporate the analytics aspect of Kiva.

As evidenced by Figures 4.3.2-4, standard network analytics (e.g., identification of structural holes and brokers) can already be achieved through existing Slack integrations like Compass. While New Noun Phrases and other analyses don't presently exist in any Slack integration, these tools could themselves be built out as an addendum to the Kiva Project to ensure that these analyses can be performed on more complete data sets. With Slack and Google Drive, not only will communication be easier but staff will be better positioned to monitor the class and supply their insights to make sure future rapid prototyping projects succeed. As evidenced by the relatively large proportion of posts to Everyone during Phase 1, much of the communication between teams (and especially within teams) was hidden from course staff; this challenge of Kiva adoption may get even worse as the course scales.

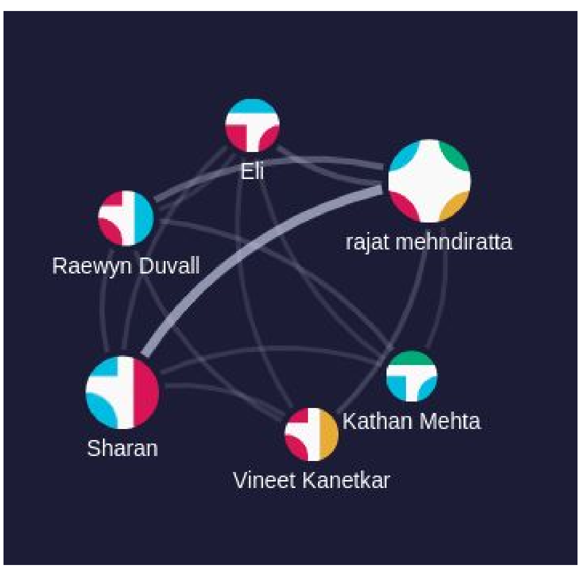


Figure 4.3.2: A network graph generated from the communications of the Wandering Tracker team over Phases 2 and 3, using the Compass Slack integration

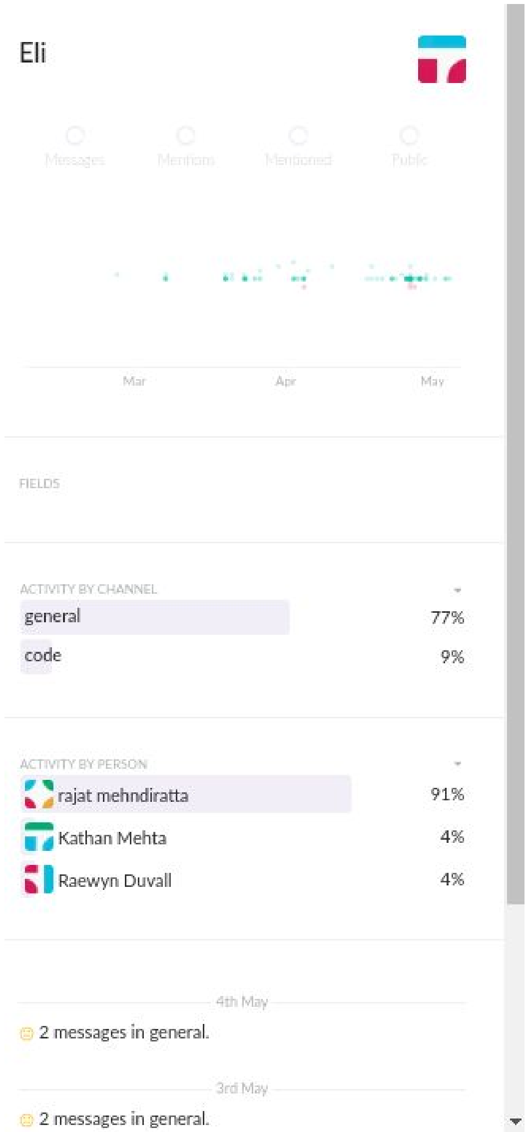


Figure 4.3.3: Sample analyses of a single member of the Wandering Tracker team, including sentiment analysis (top), the channels they posted on most frequently (middle), and the people with whom they interacted the most (bottom)

Figure 4.3.4: Rankings of the Wandering Tracker team by Compass based on network interactions

4.4 Contribution and Work Log Hours Summary for Phase 3

Work Log Hours Summary					
	Team Members	Roles	Member Contributions	Team contribution	Phase 3 Work Hours
Caregiver / Doctor Interaction	Anyi Diao	Editor	- Led the development of doctor and caregiver dashboard, set timeline, maintained communication with other teams - Implemented the doctor dashboard including home page (schedule, search patient), integrated data visualization charts, implemented data fetch from database - Acted as the doctor in the final demo - As the editor, splitted the work of phase 3 report to each team member and integrated into the final report	- Implemented the frontend interactions of doctor and caregiver dashboard - Coordinated with server / core analytics team on the data schema required for the dashboards - Two team members Ritu and Daniela created the final demo script, coordinated with all the sensors team on features that can be showcased, created the back-up demo video, and led the practice of the final demo	48 hr
	Daniela Marmolejos	Leader / Member in Demo Master team	- Led the Caregiver & Dashboard team in Phase III, attended team leader meetings - Communicated with all sub functional teams regarding functionalities to show in the demo - Created the script for the final demo - Filmed and edited the backup video for final demo - Organized and helped facilitate the demo practice - Acted as the caregiver in the final demo		25 hr 45 min

	Ritu Parekh	Member / Member in Demo Master team	- Redesigned caregiver dashboard (trends, incident summary) - Communicated with all teams on features to include in the final demo - Created scenarios and general flow for final demo - Contributed to demo video shooting - Designed class T-shirt - Helped run practice demo - Acted as narrator in the final demo		35 hr 35 min
	Xuan Xu	Presentor	- Implemented the doctor dashboard including patient information page (basic info, patient incident log, next appointment) - Prepared and gave the phase 3 report as the presenter for the team - Wrote up the doctor dashboard part for phase 3 report		41 hr 20 min
	Zhe Zhao	Member	- Implemented caregiver dashboard (incident log, integrate patient trends graph) - Coordinated with demo master, core analytics, server, and other sensor teams on finalizing the features that should be included in caregiver dashboard		27 hr 30 min
Core Analytics	Wei Ling Chang	Presenter	Mat and pillow data analysis		
	Jennifer Liu	Member	Fall, pulse rate, respiratory rate, and wandering data analysis		
	Yuling Liu	Editor	Doctor and Caregiver	Core analytics team	

			Dashboard Data Visualization	detected patient incidents and updated analytical results in the server for front-end to display, as well as sent real-time text notifications to the caregiver and helped with data visualization. We acted as a mediator in data pipeline, transforming raw data into human-readable and valuable result. We also helped the interactions team with graph visualization in dashboard.	182 hours
	Annan Ma	Member	Text notification for caregivers, react JS for sleeping chart.		
	Xiaoyu Qiu	Member	Setup and implemented analytics script in server. Communicated with home sensors group, watch group as well as server/database group, and updated software as data schema changes. Enabled text notification service for free. Represented the group in demo meetings.		
	Luo Yu	Leader	Attend the leader meeting, and demo meeting. Organize the group meeting. Analyze the stove data. Calculate the heart rate attributable risk.		
Short Term Memory Tests	Sooyoung Ahn	Presenter	- Developed software for performing short term memory assessments within tablet app. - Created final presentation for STM group. - Prepared and presented all work completed throughout course for STM group.	-Created the Daily Assessment Survey in the tablet app -Added functionality in the tablet app to support Tile trackers -Implemented Flic button system to locate patient tablet. -Communicated with other groups about the data structure visual methods and any other details.	45 hr
	Zachary Armendariz	Leader	- Assisted with integration work between hardware and software with Manu Gopakumar.		

			- Attended all leaders meetings and provided recaps and important details to team. - Acted as demo master for project and ensured integration of teams work into final demo for class.		43 hr
	Joe Fifty	Editor	- Contributed to phase 3 report - Built daily assessment survey in android tablet with Sooyoung - Helped prepare slides for presentation - Provided integral role in creating a common android project across groups as well helping people navigate android and git		46 hr
	Manu Gopakumar	Executive Contributor	- Tested and found boundaries on Tile tracker functionalities - Developed link between class app and Tile app - Connected Flic smart buttons to class app - Developed software to respond to Flic button presses and		37 hr

			sound find alert through lock screen		
Home Sensors	Justin Chu	Leader	Leaded the Home Sensor’s team during phase three. Organized team meetings to work on implementing the sensors, communicated with the OS team about endpoint query parameters, and had continued discussions with the Core Analytics team on data analysis strategies. Managed the teams Git repository and wrote the JSON http connector module that was responsible for communicating with the server for all sensors.	-Implemented the monitoring of restroom and main door which logs the number of times the patient enters the restroom or goes out of the main door by uniquely identifying the patient using an RFID ring. -Also implemented monitoring inside the kitchen which detects the presence of a person inside the kitchen and alerts the caregiver when the stove is left on for more than a predefined amount of time. -Monitored sleep by deploying sensors in the bedside mat and the pillow and logged the number of hours of sleep in the caregiver’s dashboard.	244 hours
	Karan Vasant Hebbar	Member	Worked on implementing the threshold for stove IR thermal sensor. Worked on syncing the logic of motion sensor and the stove sensor to avoid false positives. Worked on the JSON http connector module for communicating sensor data with server.		
	Priyanshi Jain	Presenter	Worked on designing the circuit, soldering and making connectors. Also did experimental measurements and helped with testing and debugging the code.		
	Jhanak Johri	Open Issues Tracker	Contributed in creating the design of the pillow		

			and the bedsite-mat, including fabricating the circuits and writing code for the required implementation. Also worked on sending the sensor values and derived useful information to the server.		
	Chinmayee Purohit	Editor	Co owned the implementation of RFID sensor and PIR sensor for the restroom and main door monitoring. Also implemented the connection of these sensors with the server and coordinated with the analytics team for the implementation of text alerts. Worked as an editor and compiled the report for her team.		
	Rohit Sreedhar	Member	Worked to implement the RFID detection of the restroom.Wrote the algorithm for the detection activity of the proximity sensor and the RFID reader. Co owned the responsibility for communicating the algorithm with the core analytics team for message alerting.		
Games / Assessments	Mo Shi	member	-Liaison with other teams including core tablet team, data analytic team, database team, doctor dashboard team and short-term memory team to discuss interactions across groups and make	-Built games related to 4 different abilities of patients, including logical thing, semantic memory, episodic memory and procedural memory. -Provide the score API for core tablet and core	260hr

			sure we meet the requirement for each group’s demand. -Evaluate feasibility analysis and implementing method for the games and assessments -Discuss and confirm the data other teams need and implement our game and assessment. -Discuss and confirm the data output for the core analytics team and doctor dashboard team. -Create slides for presentation about interaction and schema.	analysis group and work on the integration of the software system -Considered the whole development process and platform and other technical details of each game. -Communicated with other groups about the data structure visual methods and any other details.	
	Yuyan Sun	presenter	-Created 3 games for assessing logical thinking skills -Coordinated and collaborated with team members on integrating the games and score reporting utility -Communicated with the Core Tablet and Short Term Memory team on integrating the application -Led creation of final presentation slides and presented them -Worked with the Dashboard team on end-to-end real-time score update for the final demo		
	Guikang Xu	member	-Be responsible for the game platform that we bought on Phase II		

			-Researched games and assessment for semantic memory -Studied Java and Android Studio to merged NeuralUp into our Android App so we could play and run games of NeuralUp on our tablet -Tested the performance of NeuralUp and generated different plans to improve it -Generate final presentation slides and phase III report related to semantic memory and our game platform NeuralUp -Worked with members in our group to debug the games App		
	Derek Yan	member	-Built the Scores API, which includes adapters for each category and sending HTTP POST requests to the backend server REST API -Built the skeleton for Games and Assessments Main Activity for other members to build upon -Recorded Demo videos, which contains playing through each game once -Helped Caregiver convert their web application framework from Django to React		
	Yi He	editor	-Researched games and assessment tools for procedural memory testing -Designed the module for procedural memory testing		

			-Studied Android Studio and implemented the procedural module -Merged game applications of different types of memory testing -Contribution in iteration of ideation -Recorded contents of group meeting and post it on Kiva -Summarized the work of our team in final report		
	Bhumi Bhanushali	leader	-Researched on quiz game options available online -Built the quiz game -Communicated with Derek to integrate the game into the final application -Updated the group weekly of the professor's expectation -Prepared slide on episodic memory games -Attended extra demo meetings in studio B -Attended leader's meetings		
Wandering Tracker	Raewyn Duvall		- Identified multiple form factor solutions for the wandering tracker, including exploration of CAD as an alternative to hand-soldering - Contributed to final assembly of circuit and troubleshooted unreliable soldering connections, leading to first working soldered version of core Flora-FONA circuit	Designed, programmed, and assembled a shoe-based wandering tracker (circuit and case positioned on a pair of shoes) that connected to the server and periodically pushed the patient's latest location. Also set up an inductive charging mat for said tracker.	35
	Vineet Kanetkar	Editor	- Worked on the hardware integration for the entire wandering		62

			tracker project. Interfaced Adafruit Flora Microcontroller and tested several codes by programming the microcontroller. - Contributed to the server (software) part by working on PyGoogleVoice and Google Cloud Platform to send data to the backend server. - Worked on the 3D model using Autodesk Fusion 360 software so as to fit the circuit on the shoe. - Being the editor, made a report draft and attended editor's meetings to decide the final layout and outline for the project.		
	Rajat Mehndiratta	Leader	- Characterized power and charging solutions and set up the final charging circuit; soldered together the circuit and helped physically assemble the final tracker. - Helped resolve authentication/HTTP challenges by using an intermediary server/worker to convert wandering tracker coordinates to requests the RPCS server could process.		53:55
	Kathan Nilesh Mehta		- Continued the implementation of Adafruit and GSM modules where the aim was to finalize a way to send alert messages - Coded the FONA for its integration with the SIM		42

			808 along with other team members - Worked on the hardware connections of the circuit using inductive wires. - Worked on the code to resolve a shorting issue - Worked on sending coordinates to the server in the required format		
	Sharan Turlapati		-Developed code for Adafruit Flora - to trigger location reads from FONA and send it across an SMS -Helped modify the code developed to send HTTP requests instead of SMS - Incorporated sleeper functionality in the FLORA code to help save power. - Helped with soldering the circuit together - Helped with getting GCP server up and running to act as intermediary between FONA and RPCS server -Contributed to report and presentation		45hr
	Yi Lun Yu	Presenter	-Helped solder inductive circuit of tracker -Coordinated with caregiver dashboard to send location in the right format -Implemented caregiver dashboard page -converted caregiver incidents page from Django to React and modified UI -modified caregiver sleep trend page's UI -Implemented all Caregiver App's API calls with server (get patient real time location		55.5hr

			updates, post new schedule, post new incident, get schedule, get incident, get real time stove on/off event, get sleep trends) -presented our tracker		
Server / Database	Wenting Chang	Member	Implemented the GraphQL endpoint of the database and created a tutorial for other groups' use. The GraphQL API endpoint involved setting up the schemas to integrate with the Django models alongside the existing REST API and implementing queries (resolver functions) and mutations for each model. Tested each GraphQL model and debugged any issues, and assisted in testing the REST API. Contributed to group meetings and class discussions, helped write and edit the final report		270h
	Justin Ahn	Member	Architected and implemented the RESTful API server as well as the authentication system. The REST server implementation included query filtering and JSON data customization per each group. Outlined and wrote most of the REST API tutorial. Acted as the point of contact for leaders of other groups (including the core analytics team) to help integrate their software with the server. Set up both the production and the test server for external stakeholders,		

			making sure that they were both up-and-running at all times. Continuously updated and managed the API documentation. Represented the group in the demo meetings. Wrote a portion of the final report.		
	Xuliang Sun	Member	Contributed to the RESTful API codebase, converted data templates to data models in Django, assisted in testing and debugging the RESTful API server using Postman, attended the final integration test, cooperated with other groups to collect their feedbacks and updated accordingly the data models in the backend codebase. Assisted Caregiver/Doctor Interaction group with frontend development, participated in caregiver frontend meetings and completed the pages related to patient trends viewing. Contributed to the final report - data template part and participated in group meetings.		
	Lue Li	Leader	As a leader, I attended leader meetings and distributed missions to team members in group meetings. I also helped with Restful API of the database and finished one part of report.		
	Dingze Wang	Editor	As Editor. Participated in editors meeting. Wrote the outline for the report and distributed the report		

			task to each team member. Communicated with other teams to make sure the data schema meets their most recent needs. Helped with some implementation of the RESTful API and test the robustness of the server. Did research on how to combine GraphQL with Django. Edit the GraphQL tutorial and the final report.		
	Lizhi Zhao	Presenter	As a presenter, participate in presenters' meeting, made the OS/DB part slides for final presentation. Presented the OS/DB part in the final presentation. Demo our REST api using Postman in one lecture. Helped in implementing the REST api server in the basic architecture of django server. Wrote part of REST api tutorial. Participated in the final report writing on one part.		
Core Tablet	Allen Chao	Editor	-Implemented navigation bar of the core tablet application to link to different pages -Utilized Android Studio to implement the core tablet application -Responsible for editing and integrating our part to phase 3 final report -Attend meeting with the editors to discuss about the format and content of the phase 3 final report	Designed and implemented the interface of the core tablet for patients	31:40
	Warren Glasner	Member	-Implemented home page of the core tablet application		28:40

			-Debugging core tablet app -Coordinated technologies with other groups -Designed and iterated upon case designs for the wandering tracker team		
	Cong Liao	Member	-Implemented the calendar part of core tablet, including calendar page, notification page, and add activity page. -Helped Allen to debug some issues about navigation bar. -Assisted in the final report.		27:00
	Abigail McManus	Member	- Researched appropriate meditations for calming exercises for Alzheimer's patients - Implemented audio players in the exercises page to play the calming exercises - Styled the exercises page - Helped edit the final report		26.5
	Junyan Shi	Presenter	-Implemented the patient info page and the patient info edit page -Helped in merging each parts together -Fixed the styling of each pages -Led creation of final presentation slides and presented them		24.5
	Esha Tandon	Leader	-Attended the leaders meeting and presented the core tablet teams progress. -Assisted in the report and final phase 3 presentation.		22:30

Watch	Yunqian Chen	Presenter	-Studied and practised the basic Android Studio functionality and create some small demo projects. -Contributed to the slides and presentation of phase III. -Contributed to the report content building.	Detect users' heart rate, falling situation and transferring data to the database. Detect environment conditions such as light and audio.	220h
	Eryn Hopps	Member	-Collected sleep, stress, and fall detection libraries Took the lead on integrating Twilio text message notifications (although we ended up not using them) -Helped team brainstorm sleep algorithm -Represented Watch team in integration meeting; communicated with other teams on Kiva and in person -Organized team meeting; posted schedules to organize class-wide meetings -Presented Watch team's work at Meeting of the Minds symposium		
	Mingyu Hu	Editor	-Studied and practised the basic Android Studio functionality and create some small demo projects. -Implemented the file handler class that is used to store reading data from sensors and write data to other team. -Participated in the falling, heartbeat rating and sleeping experiments. -Contributed to the report content building. -Meet with other editor and contributed to the		

			phase III report editing.		
	Akshat Prakash	Member	-Studied and practised the basic Android Studio functionality and create some small demo projects. -Designed the draft algorithm about heartbeat and sleep quality coding. -Contributed to the report content building.		
	Nithin Kumar Sathish	Leader	-Studied and practised the basic Android Studio functionality and create some small demo projects. -Contributed to the report content building.		
	Abel Tesfaye	Member	-Studied and practised the basic Android Studio functionality and create some small demo projects. -Built demo project for presentation. -Communicated with database groups about the data transformation. -Mainly implemented the falling detection and heartbeat coding part. -Contributed to the report content building.		