

# Software Design Document

for  
**Want2Remember**

Version 2.0 approved

## Prepared by

David Pablo	Project Co-Lead
Jesse Gonzalez	Project Co-Lead
Harshil Kotamreddy	Documentation Co-Lead
Roger Ramirez	Documentation Co-Lead
Salvador Cornejo	Engineer
Angelo Esmeralda	Engineer
Victor Flores	Engineer
Marco Gonzalez	Engineer
Jonah Kim	Engineer
Ryan Mckean	Engineer
Zilong Ye	Faculty Advisor
Ricardo Marroquin	We2Link Liaison
Michael Malone	We2Link CEO

9 December 2022

Table of Contents	2
Revision History	4
1. Introduction	5
1.1. Purpose	5
1.2. Document Conventions	5
1.3. Intended Audience and Reading Suggestions	5
1.4. System Overview	5
2. Design Considerations	6
2.1. Assumptions and dependencies	6
2.2. General Constraints	6
2.3. Goals and Guidelines	6
2.4. Development Methods	6
3. Architectural Strategies	8
4. System Architecture	11
4.1. Level 0 DFD	11
4.2. Level 1 DFD	11
5. Policies and Tactics	13
5.1. Specific Products Used	13
5.2. Requirements traceability	13
5.3. Testing the software	13
5.4. Engineering trade-offs	13
5.5. Guidelines and conventions	13
5.6. Protocols	13
5.7. Maintaining the software	13
5.8. Interfaces	13
5.9. System's deliverables	13
5.10. Abstraction	13
6. Detailed System Design	14
6.x Name of Module	14
6.x.1 Responsibilities	14
6.x.2 Constraints	14
6.x.3 Composition	14
6.x.4 Uses/Interactions	14
6.x.5 Resources	14
6.x.6 Interface/Exports	15
7. Detailed Lower level Component Design	16
7.x Name of Class or File	16
7.x.1 Classification	16
7.x.2 Processing Narrative(PSPEC)	16
7.x.3 Interface Description	16
7.x.4 Processing Detail	16
7.x.4.1 Design Class Hierarchy	16
7.x.4.2 Restrictions/Limitations	16
7.x.4.3 Performance Issues	16
7.x.4.4 Design Constraints	16

7.x.4.5	Processing Detail For Each Operation	16
8.	Database Design	17
9.	User Interface	18
9.1.	Overview of User Interface	18
9.2.	Screen Frameworks or Images	18
9.3.	User Interface Flow Model	23
	Requirements Validation and Verification	24
	Glossary	25
	References	26

## Revision History

Name	Date	Reason For Changes	Version
David Pablo	12/02/2022	Initial Access	1.0
David Pablo	12/02/2022	Cross-Reference with Past SRS	1.1
David Pablo	12/09/2022	First Draft	1.0
David Pablo	05/10/2023	Final Draft	2.0

# 1. Introduction

## 1.1 Purpose

Want2Remember is a web application that shall assist those with cognitive impairments remember day-to-day tasks and memories. This system shall keep track of user-generated data, such as memories, to-do lists, and appointments, as well as contacts and payment information. This document shall explain the functionality, design, architecture, and requirements of the Want2Remember application.

## 1.2 Scope

The purpose of this document is to provide an overview of the design and development of this project, as well as the recommended procedures and available resources for developers and maintainers of Want2Remember. This document is not an expression of formal policy; it contains documentation for the Want2Remember system and generally agreed-upon best practices.

## 1.3 Intended Audience and Reading Suggestions

This document is for the use in the California State University of Los Angeles (CSULA) Computer Science department. This document will be covering the details of the Senior Design Project of the We2Link sponsored group. The intended audience for this project are the professors and students of CSULA, as well as the employees of We2Link.

## 1.4 System Overview

Want2Remember uses a component-based architecture that allows for the reusability of components so that it reduces the size and complexity of our codebase. The app is made using React JavaScript and individual parts that work together form what is displayed on the screen.

- **The User (Developer):**
- **The Framework:**

## **2. Design Considerations**

### **2.1 Assumptions and Dependencies**

Software used to support the Want2Remember application include:

- JavaScript
- ReactJS
- Github
- JIRA & Agile Development Technology
- Firebase
- Test Fairy, Test flight

Additionally, the user is expected to have a mobile device running on the minimum required operating systems .

### **2.2 General Constraints**

The major hurdles associated with this project are:

- Learning the JavaScript web application framework, ReactJS
- Working as a group remotely post the COVID-19 pandemic

We were given access to the Udemy course “React - The Complete Guide (incl Hooks, React Router, Redux)” to guide us through the framework and language. To collaborate, we formed sub-teams to break down the project objectives into manageable pieces. We met virtually twice weekly for status updates and sprint retrospectives and communicated via Slack/discord with our sub-team members.

In addition, there are limitations surrounding local data storage and local security. Future updates shall address these issues once we migrate to cloud storage. Using firebase to emulate the web app.

### **2.3 Goals and Guidelines**

Want2Remember is a web application to help those with memory issues - whether they are from brain injuries, Alzheimer’s, or other cognitive impairments. The Want2Remember web app shall provide the user templates to log memories, passwords, to-do lists, medications, appointments, and other important things the user may want to remember. The user shall be able to record events and reminders from the past, present, and future.

The software features shall help facilitate the user’s ability to live independently, return to work, maintain social interaction, increase work efficiency, maintain personal safety, as well as any

other needs that may come up in development. The web app features shall also help facilitate caregiver needs, as well as improve medical support.

Due to the need for simplicity and reliability, we chose the respective technologies because of the support and modularization they offer. We take the feedback from our beta testers to refactor the code and make improvements. The hope is for Want2Remember to be a viable product and to go to market by Summer 2023.

## **2.4 Development Methods**

Our dedicated team has spent months developing Want2Remember, building off the efforts of last year's Senior Design project. We follow their same methodology; we use the Agile Development Process. We break down tasks and assign them to sub-teams in Sprints of one-to-two-week durations. Agile architecture allows us to pivot and adapt quickly to new changes, which allows us to respond swiftly to any bug fixes or feedback from beta testers. We also visualized the task board using Atlassian's Jira software to improve workflow.

## **3. Architectural Strategies**

### **3.1 Use of Particular Products**

- 3.1.1 JavaScript and ReactJS: Developing with the ReactJS framework gives us the advantage of developing seamlessly for all web compatible devices. The code is written in JavaScript but is then rendered with reactjs code. ReactJS also puts emphasis on creating and reusing components within many screens of the application. This helps save time during development and encourages us to take full advantage of the utilities offered by ReactJS such as a multi-platform codebase, web app development, and fast refresh as soon as new code is saved.
- 3.1.2 Github: We use Github for version control.
- 3.1.3 Firebase: We use firebase for emulation

### **3.2 Reuse of Existing Software Components**

- 3.2.1 This software is built off of last year's Senior Design project for We2Link. We are using their components as a basis to implement them into the web application.

### **3.3 Future plans for software enhancement**

Several features have been marked for future development:

- Registration/Sign on
- Medication
- Customization
- Caregiver support
- Medication Tracker template
- Mobile advertisement
- Premium features

### **3.4 User Interface Paradigms**

- 3.4.1 The user interface shall present the Home Screen upon initial access.
- 3.4.2 The system shall provide a uniform look and feel between all pages. Headings, banners, fonts, and buttons shall follow the same style guide.
- 3.4.3 All screens shall include a dynamic header component and bottom navigation bar.
  - 3.4.3.1 Bottom navigation bar shall include Home, Create, Search, and Contacts.
  - 3.4.3.2 Back button will be in the top right to return the user back to the Home Screen.

- 3.4.4 Home Screen (default screen) shall show brief overview of all existing memories.
  - 3.4.4.1 Help, settings, and filters will be in the top left corner for navigation.
  - 3.4.4.2 The Memory navigation bar shall show “All Memories” by default, with other subcategory options listed in the scroll bar to the left.
  - 3.4.4.3 Below this navigation bar, the preview of the memory shall show with date, title, category, and other relevant summary information.
  - 3.4.4.4 A button at the bottom right shall allow the user quick access to “Create,” leading them to the Create Memory Screen.
- 3.4.5 Create Screen shall show all memory templates.
  - 3.4.5.1 Memory tiles shall be labeled and color-coded in an accessibility-friendly way.
  - 3.4.5.2 Clicking on a memory tile shall lead the user into a custom template.
- 3.4.6 Search Screen shall let users filter through their memory tiles
  - 3.4.6.1 Search bar.
  - 3.4.6.2 When a user searches, the scrollable results shall display below the search bar.
- 3.4.7 Contacts Screen shall show all contacts in scrollable view.
  - 3.4.7.1 Add Contact shall be in the top right corner
  - 3.4.7.2 Search bar.
  - 3.4.7.3 Quick view shall include name, quick links to contact, and associated memories.
  - 3.4.7.4 Clicking on View More shall allow the user to see full details of that contact.
- 3.4.8 Settings Screen
  - 3.4.8.1 The settings options shall allow the user the ability to import and export JSON format memories.
  - 3.4.8.2 The user shall be able to create and reset their secured PIN.
  - 3.4.8.3 The user shall be able to send feedback.

## **3.5 Hardware and/or Software interface paradigms**

- 3.5.1 Hardware interface paradigm
  - Device must have a display for user input /interaction.

## **3.6 Error detection and recovery**

- 3.6.1 Implement exception handling in the codes
- 3.6.2 Debugging the software interfaces/programs
- 3.6.3 To recover the issue, discussion must be made within the group.

## **3.7 Memory management policy**

- 3.7.1 Memory management of the software shall be managed by the framework

## **3.8 External databases**

- 3.8.1 Cloud Firestore: We use cloud firestore to allow for compatibility with the mobile app. Cloud Firestore is a NoSQL database hosted by Google that stores data in a document-based model and allows for scalability and expansion.

## **3.9 Distributed data/ control over a network**

N/A

## **3.10 Generalized approaches to control**

TBD

## **3.11 Concurrences and Synchronization**

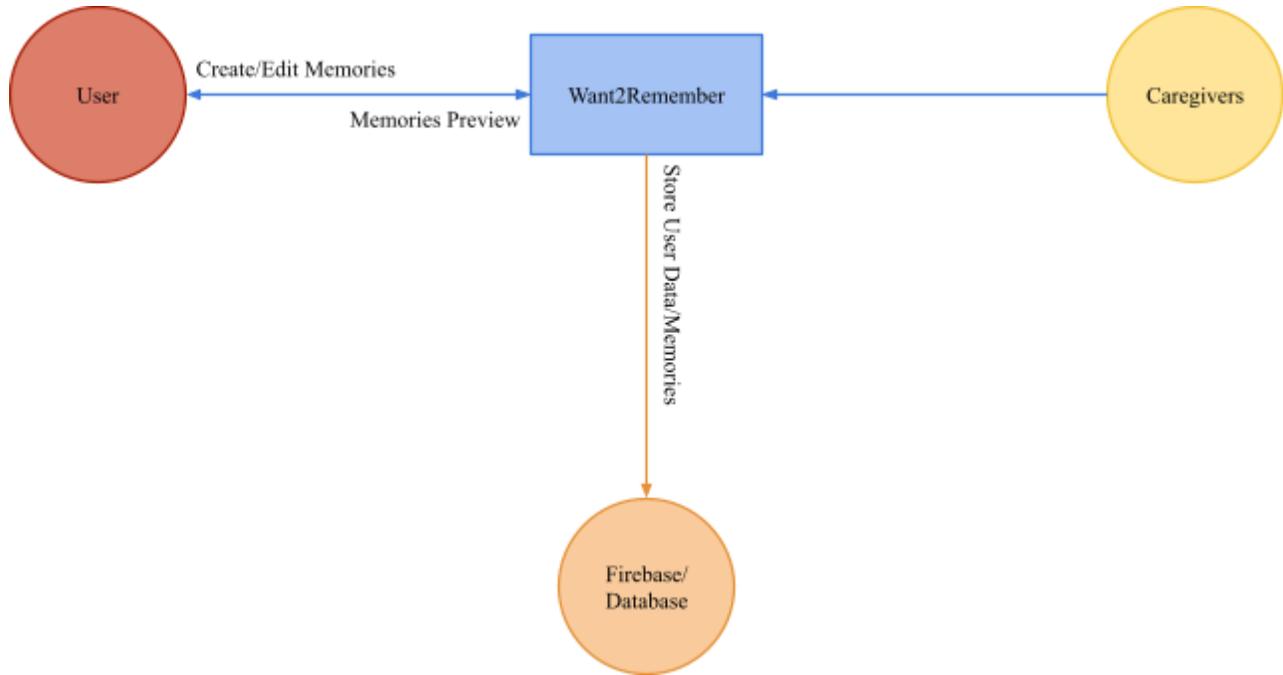
TBD

## **3.12 Communication mechanisms**

- 3.12.1 The software shall allow the user to contact the administrators through help screen in a web application.

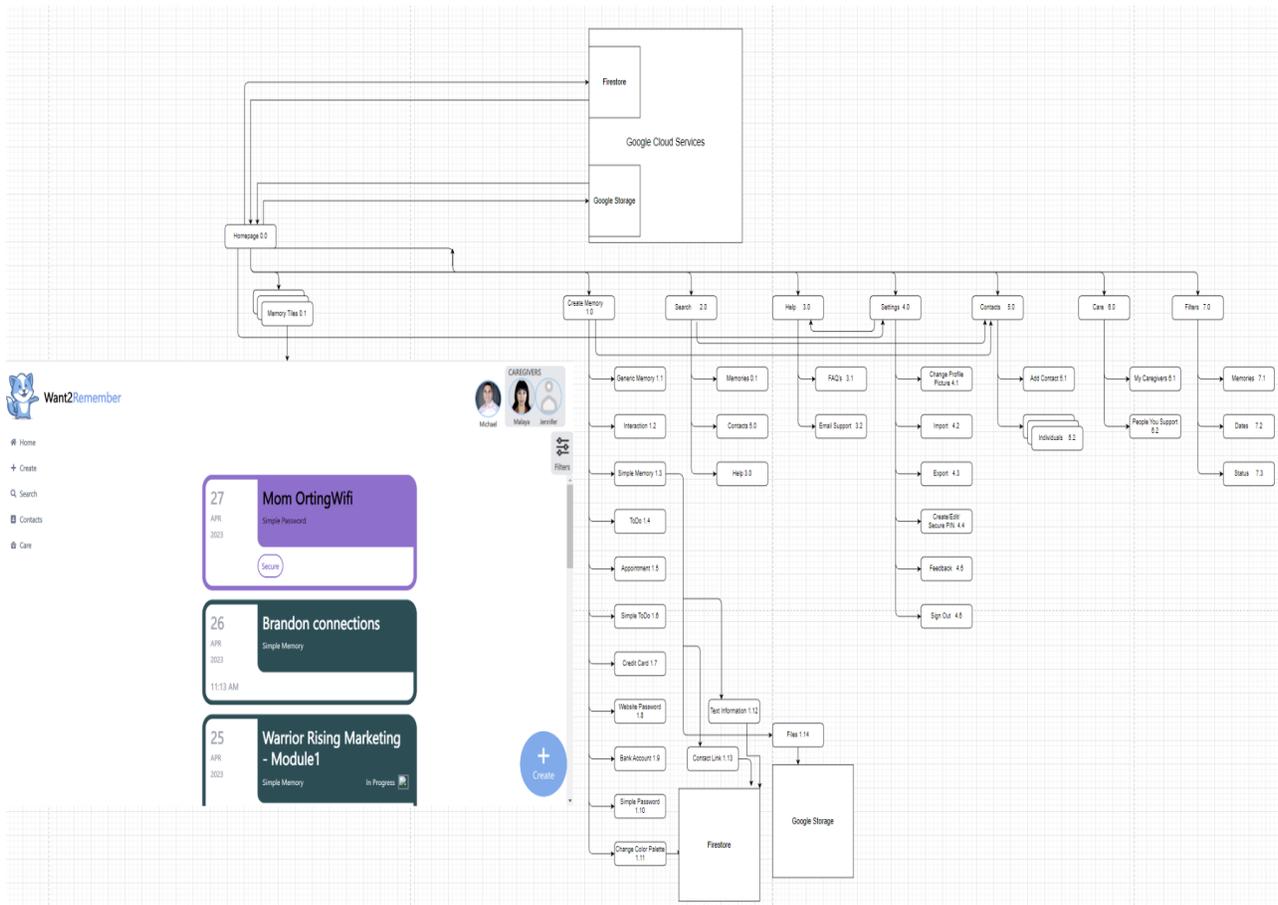
# 4. System Architecture

## 4.1 Level 0 DFD



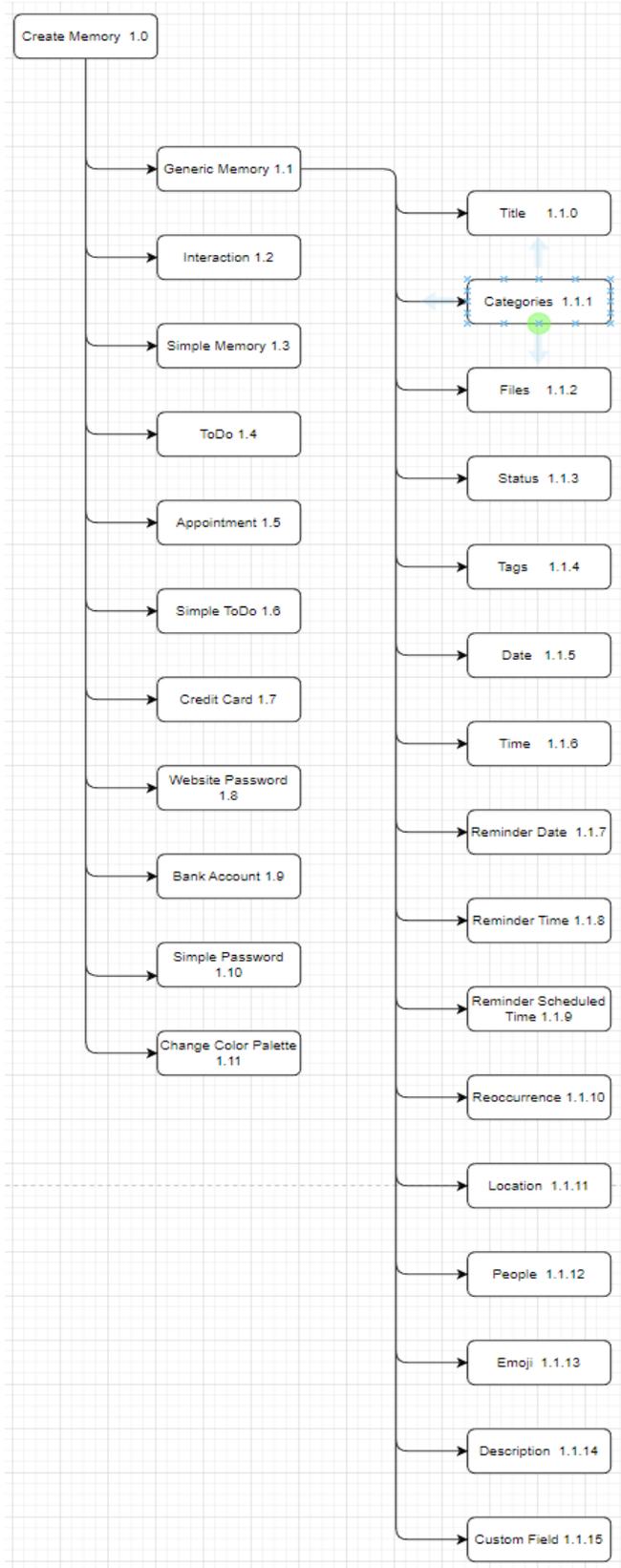
The Want2Remember app is meant to allow users to store and create memories, while allowing for caregivers to support the user. In our implementation, we stored memories and user data in firebase. When the user interacts with the Want2Remember app, they create and edit memories. When a change is detected, a change is made in Firebase. Finally, caregivers have the ability to interact with the app to assist the user.

## 4.2 Level 1 DFD



When the user is directed to the login page and enters their login credentials, the data is stored into Firestore. There is also an option for the user to create a new account and login, which will also be saved. Firestore is able to store the data so it is maintained every time the user logs in. The components for the screen are created using React JS which is a Javascript library. The screens all display individual components that are imported. The screens that are created using React include the Caregivers, Memories, Account, and the Create Button screen. The account screen has a modal that displays the account information, along with several options that the user can choose from for their account. These include: Importing memories, exporting memories, creating and editing pins, sending feedback, and help/FAQ. The create memory button directs the user to a screen to view existing memories with tags. They also had the option to create a new memory through the create memory page. The create memory page has types of memories available to select and there are components that each memory will have. These components include: Category, Date, Time, Status, Reminders, Location, Credit Card, and Custom Info. The info for each component is written and stored onto Firestore each time they are edited and saved. The main reason why this layout for the DFD level 1 was chosen was because it is easy to follow the order flow of the system. The arrows clearly define the interaction and relationship between the various components and subsystems.

Below are more architectural designs behind our project.



Search 2.0



Memories 2.1

Contacts 2.2

Help 2.3

Help 4.0



FAQ's 3.1

Email Support 3.2

Settings 3.0



Change Profile Picture 4.1

Import 4.2

Export 4.3

Create/Edit/Secure PIN 4.4

Feedback 4.5

Sign Out 4.6

Contacts 5.0



Add 5.1

Individuals 5.2



Name 5.2.1

Phone Number 5.2.2

Relationship 5.2.3

Memories 5.2.4

Tags 5.2.5

Care 6.0



My Caregivers 6.1

People You Support  
6.2

Filters 7.0



Memories 7.1

Status 7.2

Date 7.3

## **5. Policies and Tactics**

### **5.1 Choice of which specific products used**

- Integrated Development Environment (IDE): Visual Studio Code
- Project Management Software: Atlassian JIRA board
- Web Application Framework: ReactJS

### **5.2 Plans for ensuring requirements traceability**

Github allows us to have version control. It enables our developers to branch off and collaborate, which is especially useful in a virtual environment.

### **5.3 Plans for testing the software**

We are currently alpha testing this application through a firebase emulation.

## **6. Detailed System Design**

### **6.1 MemoriesScreen.js**

#### **6.1.1 Definition**

Memory landing page. Users will see an overview of all the memories they have created on this application.

#### **6.1.2 Responsibilities**

The home screen is responsible for the most navigation and the most detailed user interface. This is the user's first impression of the application and must be simple, have good flow, and be intuitive to use.

#### **6.1.3 Constraints**

Memories must be created before they can be displayed, filtered, searched, or edited.

#### **6.1.4 Composition**

#### **6.1.5 Uses/Interactions**

#### **6.1.6 Resources**

#### **6.1.7 Interface/Exports**

### **6.2 CreateScreen.js**

#### **6.2.1 Definition**

#### **6.2.2 Responsibilities**

#### **6.2.3 Constraints**

#### **6.2.4 Composition**

#### **6.2.5 Uses/Interactions**

#### **6.2.6 Resources**

#### **6.2.7 Interface/Exports**

### **6.3 .0 MoreDetailScreen.js**

#### **6.3.1 Definition**

#### **6.3.2 Responsibilities**

#### **6.3.3 Constraints**

#### **6.3.4 Composition**

#### **6.3.5 Uses/Interactions**

#### **6.3.6 Resources**

#### **6.3.7 Interface/Exports**

### **6.4 ContactsScreen.js**

#### **6.4.1**

#### **6.4.2 Responsibilities**

#### **6.4.3 Constraints**

#### **6.4.4 Composition**

#### **6.4.5 Uses/Interactions**

#### **6.4.6 Resources**

#### **6.4.7 Interface/Exports**

## **6.5 ContactDetailsScreen.js**

### **6.5.1**

### **6.5.2 Responsibilities**

### **6.5.3 Constraints**

### **6.5.4 Composition**

### **6.5.5 Uses/Interactions**

### **6.5.6 Resources**

### **6.5.7 Interface/Exports**

# **7. Detailed Lower level Component Design**

## **7.1 QuickLookComponent**

- 7.1.1 Classification**
- 7.1.2 Processing Narrative (PSPEC)**
- 7.1.3 Interface Description**
- 7.1.4 Processing Detail**
  - 7.1.4.1 Design Class Hierarchy**
  - 7.1.4.2 Restrictions/Limitations**
  - 7.1.4.3 Performance Issues**
  - 7.1.4.4 Design Constraints**
  - 7.1.4.5 Processing Detail For Each Operation**

## **7.2 Detail.js**

- 7.2.1 Classification**
- 7.2.2 Processing Narrative (PSPEC)**
- 7.2.3 Interface Description**
- 7.2.4 Processing Detail**
  - 7.2.4.1 Design Class Hierarchy**
  - 7.2.4.2 Restrictions/Limitations**
  - 7.2.4.3 Performance Issues**
  - 7.2.4.4 Design Constraints**
  - 7.2.4.5 Processing Detail For Each Operation**

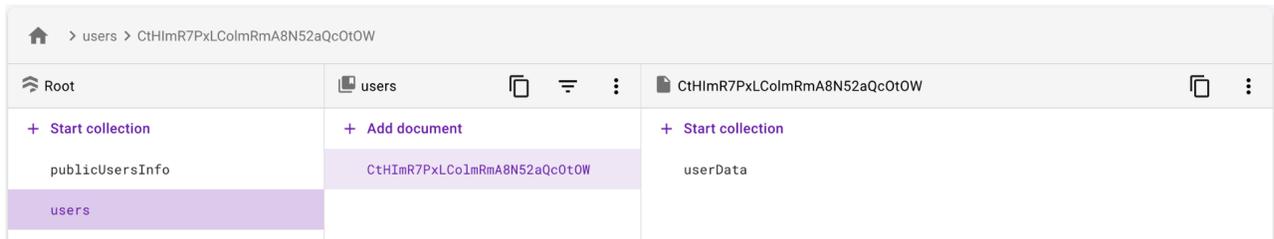
## **7.3 ContactQuicklook.js**

- 7.3.1 Classification**
- 7.3.2 Processing Narrative (PSPEC)**
- 7.3.3 Interface Description**
- 7.3.4 Processing Detail**
  - 7.3.4.1 Design Class Hierarchy**
  - 7.3.4.2 Restrictions/Limitations**
  - 7.3.4.3 Performance Issues**
  - 7.3.4.4 Design Constraints**
  - 7.3.4.5 Processing Detail For Each Operation**

## 8. Database Design

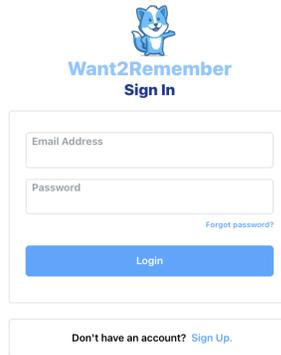
We use Cloud Firestore as our database for this project. The mobile app for Want 2 Remember also uses Firestore, allowing for compatibility. Cloud Firestore is a document-based NoSQL database hosted by Google. The documents are stored in collections, which are then used to organize the data and make queries.

We organize our data into two collections: `publicUsersInfo` and `users`. The `publicUsersInfo` collection stores data such as the UUID to match the id to the user. The `users` collection stores more information about the user including private user data.



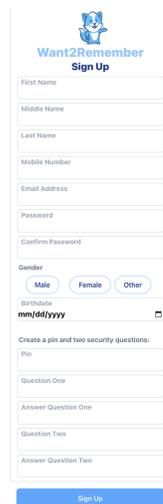
## 9. User Interface

- Users are introduced to the main page where they are asked to sign in or create an account. This uses the *SignIn.jsx* file inside the *Screens* folder to take care of this step.



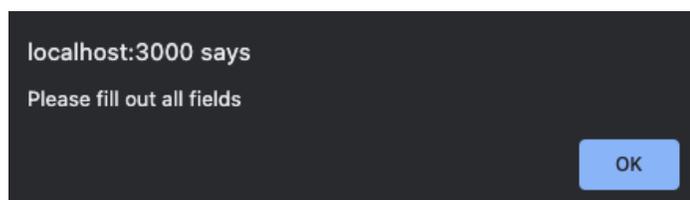
The screenshot shows the 'Sign In' page for 'Want2Remember'. At the top is a blue cartoon fox logo and the text 'Want2Remember Sign In'. Below this are two input fields: 'Email Address' and 'Password'. A blue 'Login' button is positioned below the password field. To the right of the password field is a link that says 'Forgot password?'. At the bottom of the form is a link that says 'Don't have an account? Sign Up.'

- If the user does not have an account the button “sign up” is displayed right under the login area. The user is taken to a page where they are able to enter all their credentials to create an account. This step uses the *SignUp.jsk* file in the *Screens* folder.



The screenshot shows the 'Sign Up' page for 'Want2Remember'. At the top is the blue cartoon fox logo and the text 'Want2Remember Sign Up'. The form contains several input fields: 'First Name', 'Middle Name', 'Last Name', 'Mobile Number', 'Email Address', 'Password', and 'Confirm Password'. Below these is a 'Gender' section with three radio buttons labeled 'Male', 'Female', and 'Other'. There is a 'Birthday' field with a calendar icon and the placeholder 'mm/dd/yyyy'. Below that is a section titled 'Create a pin and two security questions:' with fields for 'Pin', 'Question One', 'Answer Question One', 'Question Two', and 'Answer Question Two'. A blue 'Sign Up' button is at the bottom.

- User needs to enter all necessary information. If failed to do so, users will be presented with an error message asking them to fill out all fields.



- Once logged in, the user is then displayed with the home page where they can create a memory. The following options to click on are: Generic Memory, Interaction, Simple Memory, ToDo, Appointment, Simple ToDo, Credit Card, Website Password, Bank Account, Simple Password, Or Purchase. All these different categories have slightly different options to choose from within each one. File that takes care of this part is the *MemoryTemplateCreation.jsx* in the *Screens* folder.



Home

Search

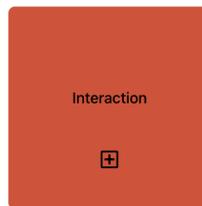
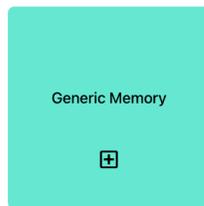
Care

Contacts

### Create Memory

Edit

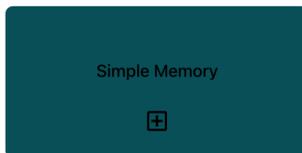
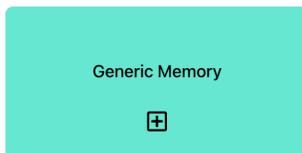
List View



### Create Memory

Edit

Grid View



- Users are displayed with different options to apply to their New Memory. This step comes after choosing one of the few Memory options. File that takes care of this step is *index.jsx* in the *Main* folder inside the *Screens* folder.

Insert Components Below

Title

**Enter Title**

Select a category [See All](#)

[Fact](#) [Event](#) [People](#)

[Secure](#)

Tags [Add Tags](#)

[Add Tag +](#)

Memory Date

mm/dd/yyyy 

Memory Time

3:32 PM [Time](#) 

Status

[Pending](#) [In Progress](#)

[Delayed](#) [Stopped](#)

[Done](#)

Reminder Date

None [Calendar](#) 

Reminder Time

None [Time](#) 

Reminder Recurrence

None [Recurrent](#) 

- Provides a section that allows the users to type text for their Memory Title. The *TitlePicker.jsx* file in the *Containers* folder takes care of this step.

Title

Enter Title

- Allows users to click the category that best suits their Memory. File that is in charge of this option is *CategoryPicker.jsx* which is located in the *Containers* Folder.

Select a category

See All

Fact Event People Secure

- Tags are available for the user to provide if needed. Users are shown “Add Tag+” where they are then given the ability to provide any tag they would like to use. File that takes care of this step is *TagsPicker.jsx* in the *Containers* folder.

Tags

Add Tags

Add Tag +

Tags

Add Tags

Add a Tag X close

Enter text here then tap Add

Add

- Memory Date allows users to upload a date to their Memory. File used in this step is *DatePicker.jsx* in the *Components* Folder.

Memory Date

mm/dd/yyyy

Memory Date

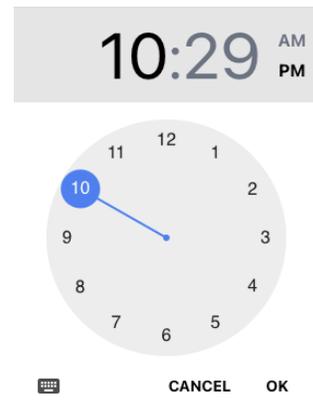
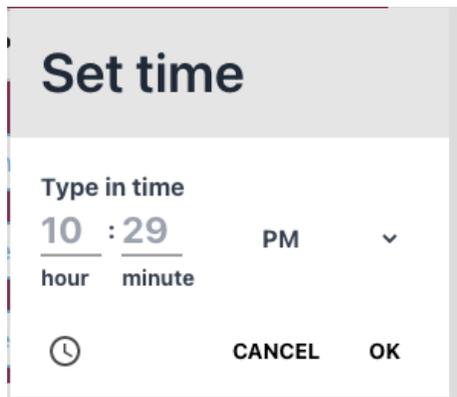
mm/dd/yyyy

December 2022

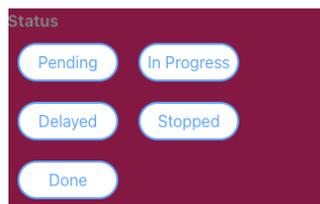
S	M	T	W	T	F	S
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Clear Today

- Allows users to upload a time for their Memory. It has two different ways a user can interact with this feature. One of the ways is by being able to use their mouse to click and move the arm of the clock to the desired time. The second way of uploading a time is by entering the time manually. Users can change to either of the two clock options by clicking on the lower left corner where either a clock icon or keyboard icon will appear. Files used for this step is *TimePicker.jsx* in the *Components* folder.

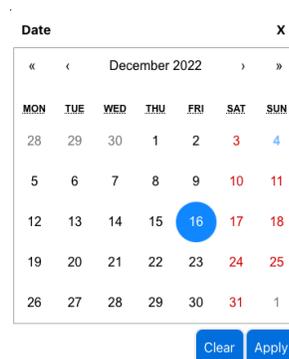


- A status area with buttons a user can choose from. File for this feature is *StatusPicker.jsx* in the *Components* folder.



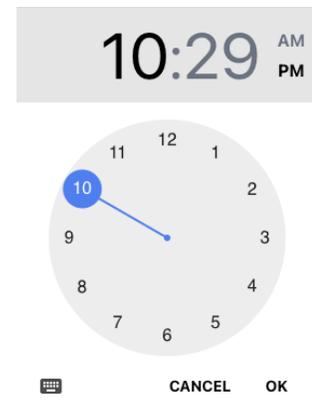
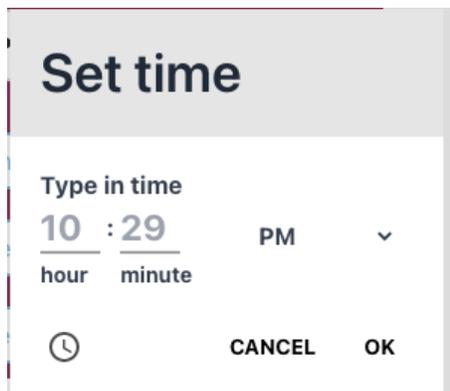
- Gives the option of setting a Reminder Date to their Memory. File for this feature is *ReminderDateTime.jsx* in the *Components* folder.

### Reminder Date



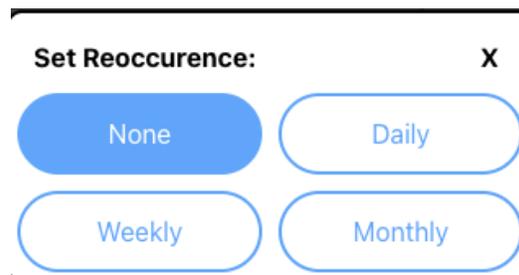
- Reminder Time allows the user to enter a specific time they want to be notified of their Memory. It has two different ways a user can interact with this feature. One of the ways is by being able to use their mouse to click and move the arm of the clock to the desired time. The second way of uploading a time is by entering the time manually. Users can change to either of the two clock options by clicking on the lower left corner where either a clock icon or keyboard icon will appear. File used for this feature is *ReminderDateTime.jsx* in the *Components* folder.

### Reminder Time



- Reminder Recurrence provides the user to be able to have their Memory be notified to them more than once. File used for this feature is *ReminderDateTime.jsx* in the *Components* folder.

### Reminder Reocurrence



- Snooze Timer provides the user the option to set a snooze time to their Memory. File used for this feature is *ReminderDateTime.jsx* in the *Components* folder.

#### Snooze Timer

None **Snooze Timer** ⌚

**Snooze Timer** X

+	+	Never Occurs
0	0	
-	-	
Hours	Minutes	

**Clear** **Apply**

- One Line Sample provides the option to add any text the user might want to add onto their Memory. File used for this feature is *OneLineTextPicker.jsx* in the *Components* folder.

**One Line Sample**

Enter Text Here

- Description Title provides the option of typing a description of the title. File used for this feature is *TextPicker.jsx* in the *Components* folder.

#### Description Title

Write Description Title

- Description Content provides the option to type in text any kind of description content of the user's Memory. File used for this feature is *TextPicker.jsx* in the *Components* folder.

#### Description Content

Write Description

- Location provides the option of attaching a location to the user's Memory. The location feature has three different options to select from: Search for Location, Manual Address, and Current Location. Search location helps the user to auto complete anything they are in search for by providing options based on what the user has typed in the search bar. Manual Address provides text boxes to type in an address. Current Location displays your current location. Once a user clicks on the button "save" for any of the location features, the location will be displayed on a map

Location

Select Location 

Close 

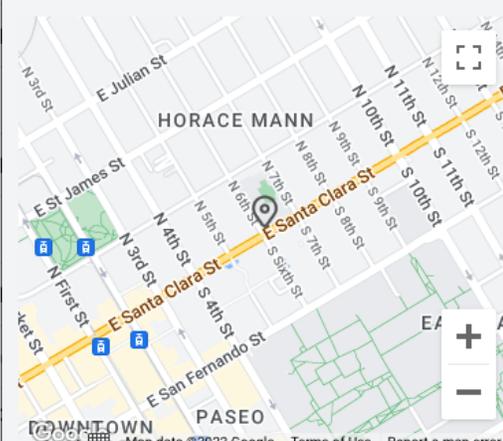
-  **Search for Location**  
Search for location using Google
-  **Manual Address**  
Enter your very own manual address
-  **Current Location**  
Use your current location address

Cancel  Save 

San J

- San Jose, CA, USA
- San Juan Capistrano, CA, USA
- San Jacinto, CA, USA
- San Julian Street, Los Angeles, CA, USA
- San José International Airport, Airport Parkway, San Jose, CA, USA

Edit  Close 

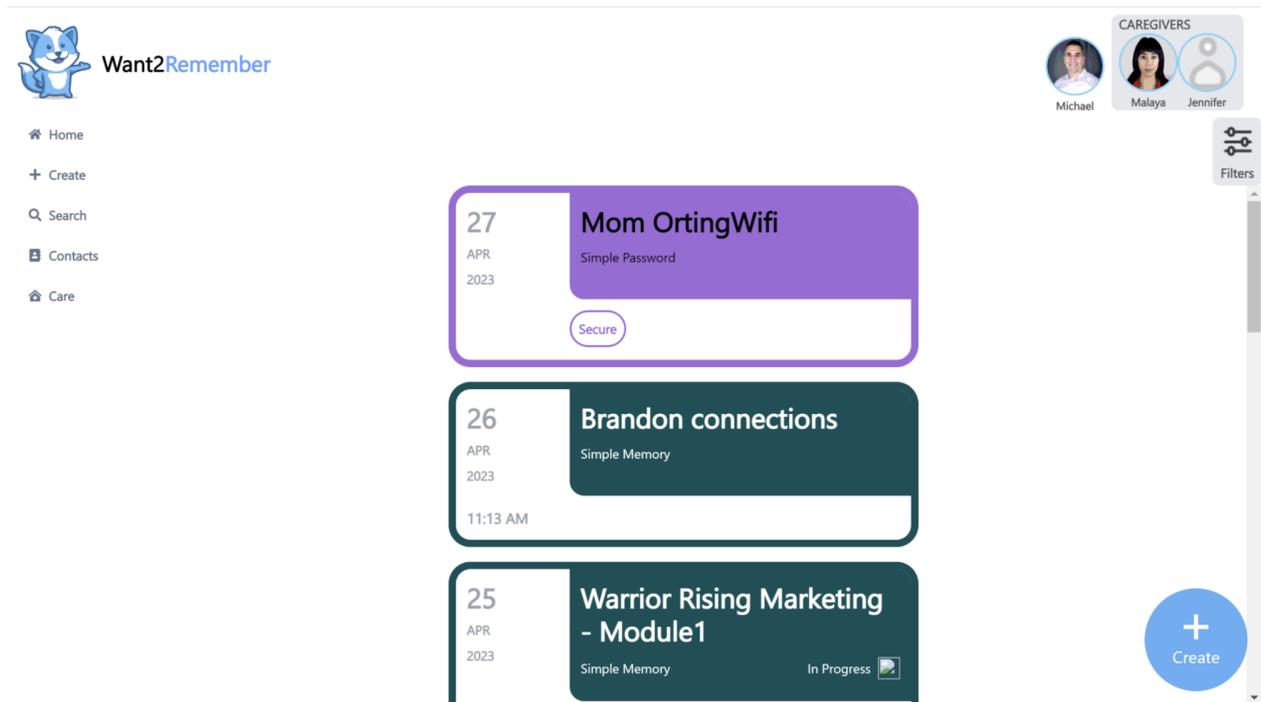


**44 East Santa Clara Street**  
San Jose  
California  
United States  
95113

## 9.1 Overview of User Interface

When a user uses Want2Remember, they will find themselves with the tools they need to help them be able to remember events, appointments, or anything they want to be notified in the future. The user will have to create an account in order to get started and have access to these helpful features that Want2Remember offers. Once a user creates an account they will have access to their profile, either on mobile or online. Whenever a user decides to create a new Memory a user will be presented with a grid of options to choose what fits best for their memory. Each one of these memory options have slightly different features within them to fit that specific type of memory. Once a user chooses a Memory they will be presented with features that they can choose to fill out or select. These features vary from different ways of picking time, dates, location, and categories that accommodate their Memory. All the different features can be seen in detail on the User Interfaces section of this document. What a user will experience from these options and features from Want2Remember is notification, planning, memory, and efficacy. This will help a user live a productive, on schedule, and minimize stress from their everyday life without forgetting deadlines, important events, or life situations. It is built for the users to navigate with ease.

## 9.2 Screen Frameworks or Images



## Create Memory

Edit

List View

Generic Memory

+

Interaction

+

Simple Memory

ToDo

## Create Memory

Edit

Grid View

Generic Memory

+

Interaction

+

Simple Memory

+

## Edit

Allows you to change the order of the memory template. You change the order rearranging them in the first column. You will see the result in the next column

Accept  
Changes

List View



Change  
Color Palette

Drag here to change order

Preview

1.  
Generic Memory  
+

2.  
Interaction  
+

3.  
Simple Memory  
+

1.  
Generic Memory  
+

2.  
Interaction  
+

3.  
Simple Memory  
+

4.  
ToDo  
+

### Change Color Palette



Default Color



Appointment



Credit Card



Simple Memory



ToDo



Purchase



Simple ToDo



Interaction



Bank Account



Generic Memory



Simple Password



Website Password

Pantone Palette



Apply

Memory ID Test

Test Memory Id

## Generic Memory

Title

Enter Title

Select a category

Edit Categories

Fact

Event

People

Secure

Food

Drinks

Vacation

Work

Fun

+ Add File

Status

Pending

In Progress

Delayed

Stopped

Done

Tags

Add Tags

Add Tag +

Memory Date

mm/dd/yyyy



Memory Time

None

Time ⌚

Reminder Date

mm/dd/yyyy



Reminder Time

None

Time ⌚

REMINDER\_SCHEDULED\_TIME Not Yet Implemented

## Reoccurrence

None

Reoccurrence 

## Snooze Timer

None

Snooze Timer 

## Add to Calendar

Add Memory To Calendar



## Location

Select Location



PEOPLE Not Yet Implemented

## Emojis

None

Tap Here to add emoji(s)!

## Description 1

Remove

### Description Title

Write Description Title

## Description Content

Write Description

## Description 2

Remove

### Description Title

Write Description Title

## Description Content

Write Description

## Description 3

Remove

### Description Title

Write Description Title

## Description Content

Write Description

+ Add Description

+ Add Custom Field

Title

# Enter Title

Select a category

[See All](#)

- Fact
- Event
- People
- Secure

Tags

[Add Tags](#)

**Add a Tag** X close

Enter text here then tap Add

---

Add

## Memory Date

12/03/2022



December 2022



Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Clear

Today

## Memory Date

12/03/2022



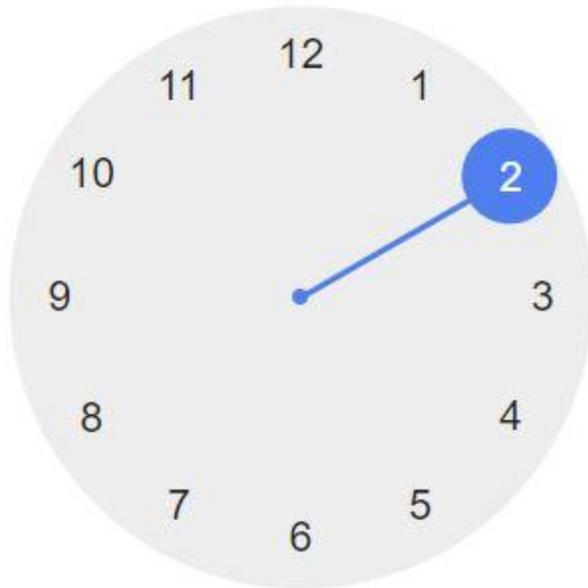
## Memory Time

2:55 PM

Time



2:52 AM  
PM



CANCEL

OK

## Set time

Type in time

2 : 52  
hour minute

PM



CANCEL

OK

## Starting Time

2:55 PM

Time 

## Ending Time

2:55 PM

Time 

## Status

Pending

In Progress

Delayed

Stopped

Done

## ReminderDate

None

Calendar 

**Date** **X**

« ‹ December 2022 › »

MON	TUE	WED	THU	FRI	SAT	SUN
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

ClearApply

**Reminder Time**

**None** Time

**Reminder Reoccurrence**

**None** Reoccurrence

**Set Reoccurrence:**

**X**

<b>None</b>	Daily
Weekly	Monthly

**Snooze Timer**

**None**

**Snooze Timer** ⌚

**Notification Time**

**None**

**Set Notification Time**

**X**

<b>None</b>	On Time
5 Minutes Before	10 Minutes Before
15 Minutes Before	30 Minutes Before
1 Hour Before	2 Hour Before

Location

Select Location



Close 



**Search for Location**

Search for location using Google



**Manual Address**

Enter your very own manual address



**Current Location**

Use your current location address

Cancel 

Save 

Sa

**Santa Monica, CA, USA**

**San Gabriel, CA, USA**

**San Diego, CA, USA**

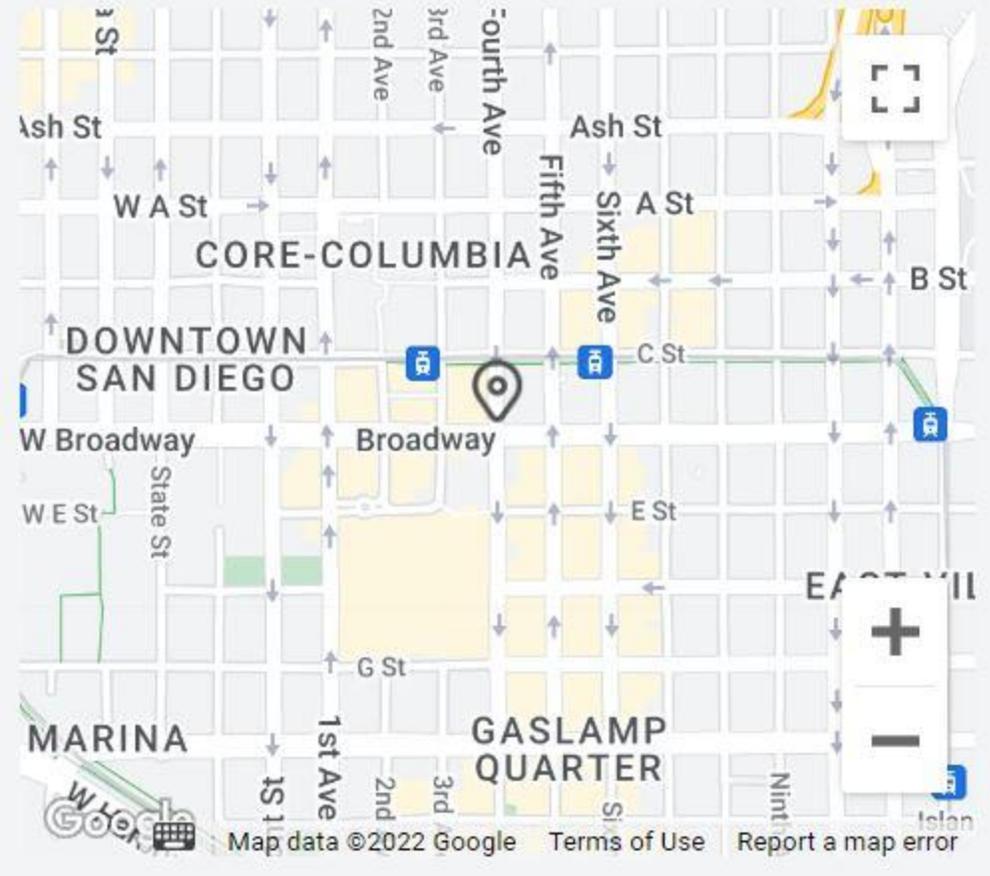
**Santa Ana, CA, USA**

**San Marino, CA, USA**

Cancel 

Save 

**San Diego, CA, USA**



**One Line Sample**

Enter Text Here

**Provider**

Provider

### Card Number

### Questions To Ask

### Pin

### CVC

### Things to Bring

### Custom Field Title

### Content

## Description Title

Write Description Title

## Description Content

Write Description

Close 

## My Account

t@gmail.com



### Change Profile Picture

Upload new image



### Import Memories

From another account



### Export Memories

# Memories



### Create or Edit Secure Pin

Recommended for Private Files



### Send us Feedback

Get in Contact with the Team



### Help

Frequently Asked Questions

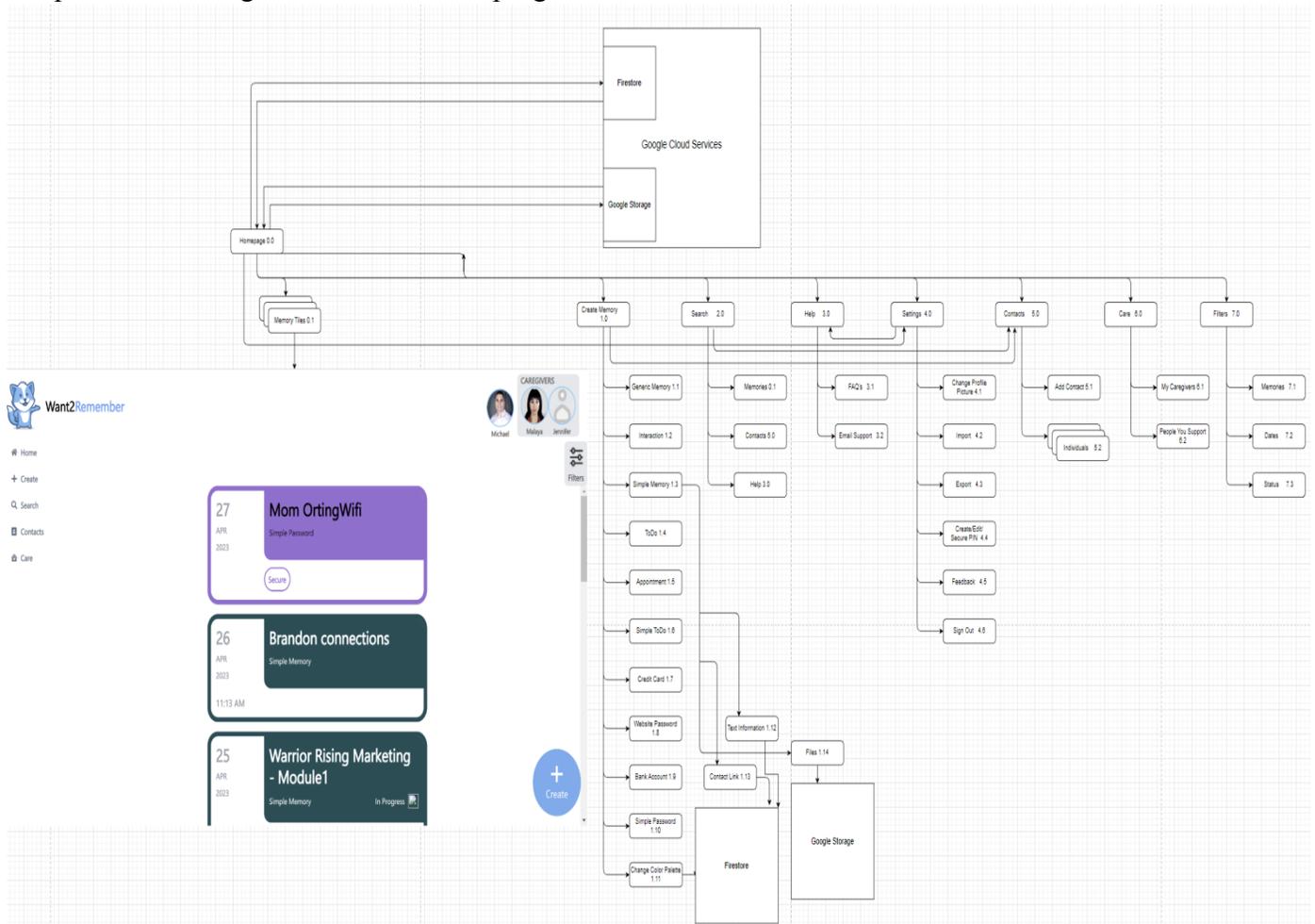
Sign Out

### 9.3 User Interface Flow Model

Currently, the user is able to navigate through the:

- Login
- Sign Up
- Main Index Page
- QuickLook UI (Soon to be user interactive)
- Profile Modal
- SideBar Menu
- Create Memory
- Memory Templates
- Help
- Settings
- Contacts
- Care
- Filters
- Smaller components within a memory creation

All these components are currently in development for the Cloud side application with most components working while others are in progress.



# 10. Requirements Validation and Verification

## 10.1 Home Screen

10.1.1	The system shall display created memories on the home screen	
10.1.2	The system shall display memories within a category selected by the user (Filter by category)	
10.1.3	The system shall display memories within a type selected by the user (Filter by memory type)	
10.1.4	The system shall display memories within a time frame selected by the user (Filter by time/date)	
10.1.5	The system shall allow the user to create a new memory	
10.1.6	The system shall allow the user to select a memory to view	

## 10.2 Create Screen

10.2.1	The system shall allow the user to select from premade memory templates	
10.2.2	The system shall allow the user to give memories a title	
10.2.3	The system shall allow the user to select a category for a memory	
10.2.4	The system shall allow the user to create a PIN for secured memories if not already created	
10.2.5	The system shall allow the user to select a date for a memory	
10.2.6	The system shall allow the user to select a time for a memory	
10.2.7	The system shall allow the user to select a reminder time for a memory	
10.2.8	The system shall allow the user to select a status for a memory	
10.2.9	The system shall allow the user to enter tags for a memory	
10.2.10	The system shall allow the user to enter a location for a memory	
10.2.11	The system shall allow the user to select people for a memory	
10.2.12	The system shall allow the user to add a custom field to a memory	
10.2.13	The system shall allow the user to enter a title to a custom field	
10.2.14	The system shall allow the user to enter content to a custom field	

## 10.3 Edit Screen

10.3.1	The system shall allow the user to edit a memory	
10.3.2	The system shall allow the user to delete a memory	

## 10.4 More Details Screen

10.4.1	The system shall allow the user to view the details of a screen	
--------	---	--

### 10.5 Contacts Screen

10.5.1	The system shall display contact quick looks	
10.5.2	The system shall display associated memories	
10.5.3	The system shall allow the user to call a contact	
10.5.4	The system shall allow the user to text a contact	
10.5.5	The system shall allow the user to email a contact	
10.5.6	The system shall allow the user to search for a contact	
10.5.7	The system shall allow the user to update contacts	

### 10.6 Search Screen

10.6.1	The system shall allow the user to search through the database with the search bar.	
10.6.2	The system shall display scrollable results below the search bar.	

### 10.7 Settings Screen

10.7.1	The system shall allow the user the ability to import/export JSON format memories.	
10.7.2	The system shall allow the user to create and reset their secured PIN.	
10.7.3	The system shall allow the user to send feedback.	

### 10.8 Backend (TBD)

10.8.1	The system shall allow synchronization between caregiver and care receiver data.	
10.8.2	The system shall encrypt user data.	

## 11. Glossary

**API:** Application Programming Interface

**CRUD:** Create, Read, Update, and Delete

**GUI:** Graphical User Interface

**HIPAA:** Health Insurance Portability and Accountability Act

**iOS:** Apple's mobile operating system

**JSON:** JavaScript Object Notation

**JS:** JavaScript

**MacOS:** Apple's Macintosh operating system

**OS:** Operating System

**PIN:** Personal Identification Number

**RSA Encryption:** public key/private key encryption method

## 12. References

Title: Software Requirements Specification Template

Author: Professor Jiang Guo

Date: (Accessed) Dec 02, 2022

Title: React - The Complete Guide (incl Hooks, React Router, Redux)

Authors: Academind by Maximilian Schwarzmüller, Maximilian Schwarzmüller

Date: (First Accessed) August 31, 2022

<https://www.udemy.com/course/react-the-complete-guide-incl-redux/>

Title: Software Design Document for Want2Remeber (Ver 2)

Authors: Kevin Benavente, Thomas Weatherell, Alejandro Salazar, Jesus Roman, Leon/Liangbin Huang, Jesus B. Osuna, Edward Ramirez, Tanya Kitchaiskulrit

Date: Dec 02, 2022

Title: Software Requirements Specification for Want2Remeber (Ver 2)

Authors: Antonio Campos, Alec Kaczmarek, Amy Guttman, Alexandra Strong, Vincent Li, Saiyang Liu, Ricardo Marroquin, Miguel Nonoal-Garcia, Jonathan Sum, Edwin Zapata Minero

Date: December 09, 2022