Software Design Document

for

Want2Remember

Version 1.0 approved

	Prepared by
Adrian Soria	**Project Co-Lead**
Davy Chi	**Project Co-Lead**
Zain Syed	**Documentation Co-Lead**
Joanne Wu	**Documentation Co-Lead**
Nicholas Barger	**Engineer**
Kevin Fong	**Engineer**
Hein Htet	**Engineer**
Jennifer Lizarraga	**Engineer**
Carlos Ramirez	**Engineer**
Ivan Wang	**Engineer**
Huiping Guo	**Faculty Advisor**
Jesus Rodriguez	**We2Link Product Manager**
Ricardo Marroquin	**We2Link Liaison**
Michael Malone	**We2Link CEO**

08 December 2023

Revision History

Name	Date	Reason For Changes	Version

<Add rows as necessary when the document is revised. This document should be consistently updated and maintained throughout your project. If ANY requirements are changed, added, removed, etc., immediately revise your document.>

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.

2. Design Considerations

2.1 Assumptions and Dependencies

Software used to support the Want2Remember application include:

● JavaScript

● ReactJS

● Github

● JIRA & Agile Development Technology

● Firebase

2.2 General Constraints

The major hurdles associated with this project are:

● Learning the JavaScript web application framework, ReactJS

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. Using firebase to emulate the web app.

2.3 Goals and Guidelines

Want2Remember is a mobile and web application to help those with memory issues - whether they are from brain injuries, Alzheimer's, or other cognitive impairments. The Want2Remember app shall provide the user templates to log memories, passwords, to-do lists, medications, appointments, contacts, 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 2024.

2.4 Development Methods

The approach used to this project is that the web team used React and JavaScript for the development of the web application and the mobile team used React Native with JavaScript for the mobile application. The usage of React and React Native allowed us to efficiently develop and maintain the codebase, as these frameworks enable component-based development and easy code reuse between the web and mobile applications. The Agile framework is also utilized for project management, receiving incoming tasks, task monitoring and assignment . We also used a backend platform, Firestore, to store the user's data which can integrate with the frontend

efficiently, enabling seamless communication between the user interface and database. The integration with Firestore not only facilitates real-time data synchronization but also ensures a responsive and dynamic user experience. As for the development environment, Visual Studio Code was used for its robust features, extensions, compatibility with React and React Native, and the live share function where team members can work collaboratively.

3. Architectural Strategies

For the choice of technologies and development environment, we used React and React Native as the primary development frameworks for the web and mobile applications, respectively. JavaScript is used as the primary programming language for its versatility and widespread support. As for development tools and environments, we used VS Code as the primary code editor and Android Studio, Xcode for platform-specific development and testing. For data storage and management, we used Firestore emulator to facilitate offline development and testing. It ensures reliable data interactions even without a live Firestore connection. The UI framework and styling include Tailwind CSS, which adopts a utility-first approach, offering a highly configurable, maintainable styling solution. It promotes consistent styling across components, ensuring a cohesive and visually appealing user interface. To improve the overall mobile application, implementing the location picker and time detection with platform-specific considerations for Android and older versions. Recognizing and addressing issues related to older Android versions demonstrates a commitment to providing a positive user experience across a diverse range of devices. These design decisions collectively contribute to the overall organization of the system, focusing on cross-platform development, efficient tooling, robust data management, and a consistent and visually appealing user interface

4. System Architecture

4.1 Level 0 DFD
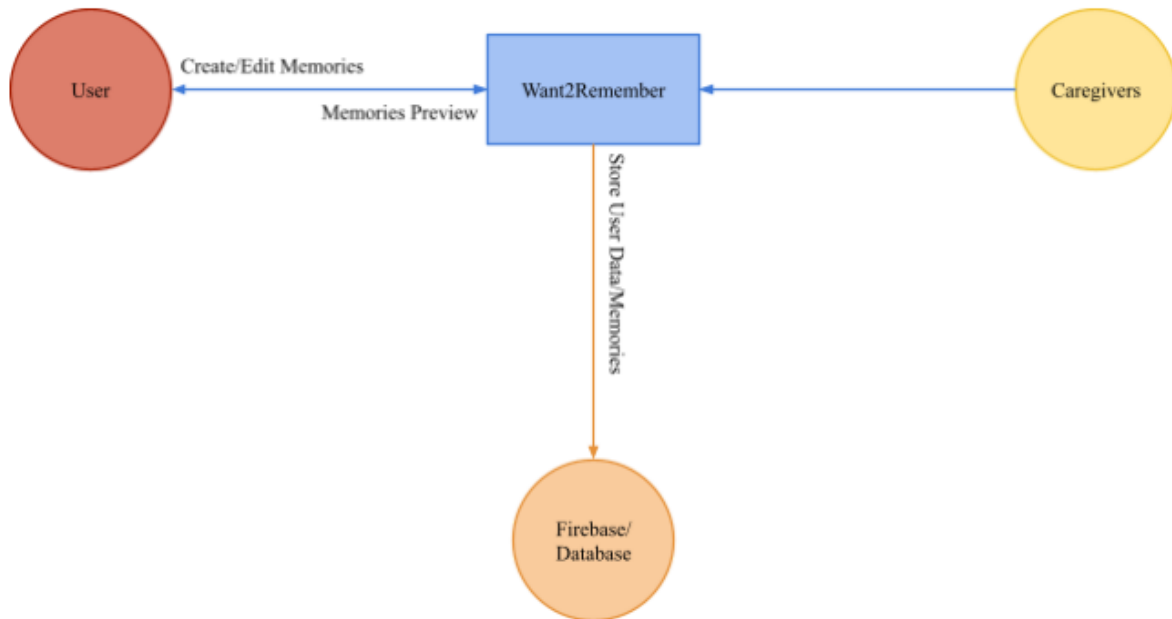
The Want2Remember app is meant to allow users to store and create memories while allowing caregivers to support the user. In our implementation, we stored memories and user data in Firebase. Users who interact with the Want2Remember app can create and edit memories. When a change is detected, a change is made in Firebase. Finally, caregivers have the ability to interact with the app to assist the user.

4.2 Level 1 DFD
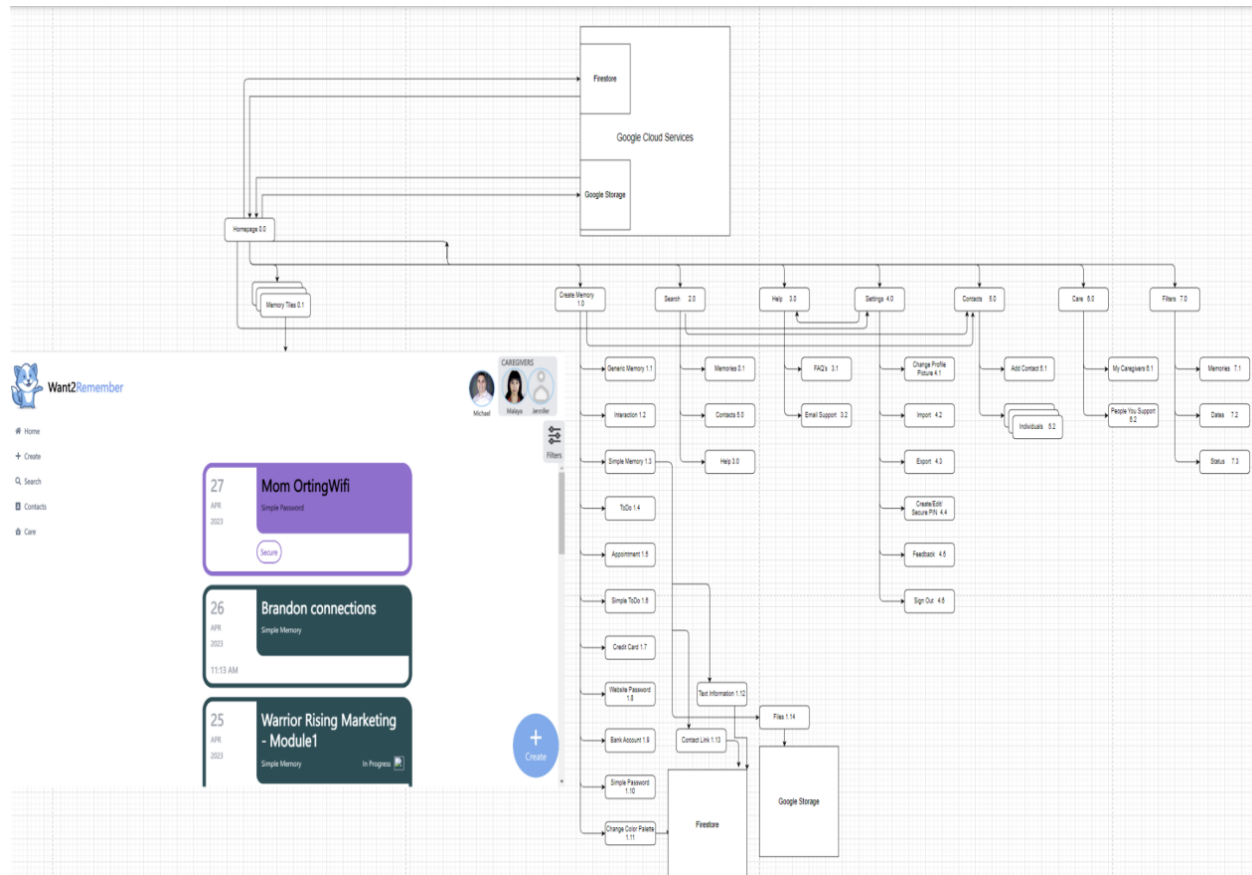
When the user is directed to the login page and enters their login credentials, the data is stored in Firebase. There is also an option for the user to create a new account and log in, which will also be saved. Firebase is able to store the data, so it is maintained every time the user logs in. The components for the screen are created using React JS, 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 on Firebase 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

→ 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
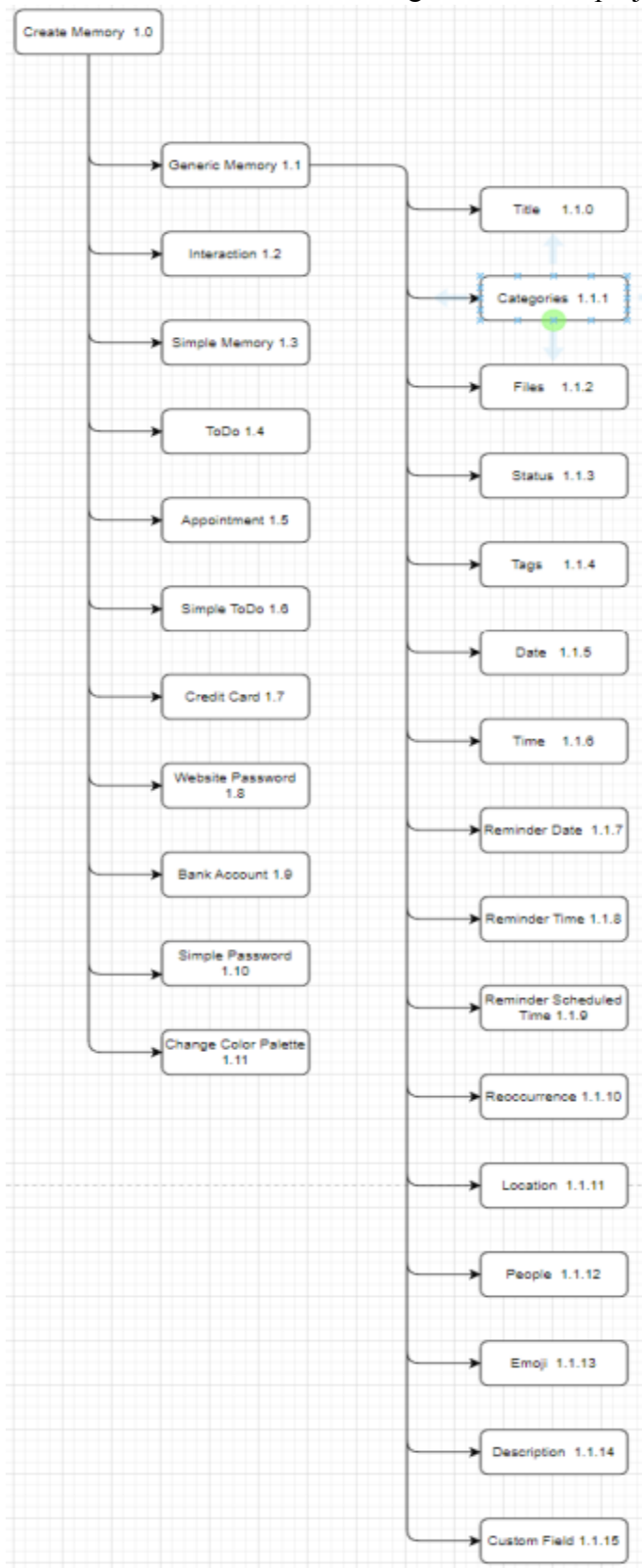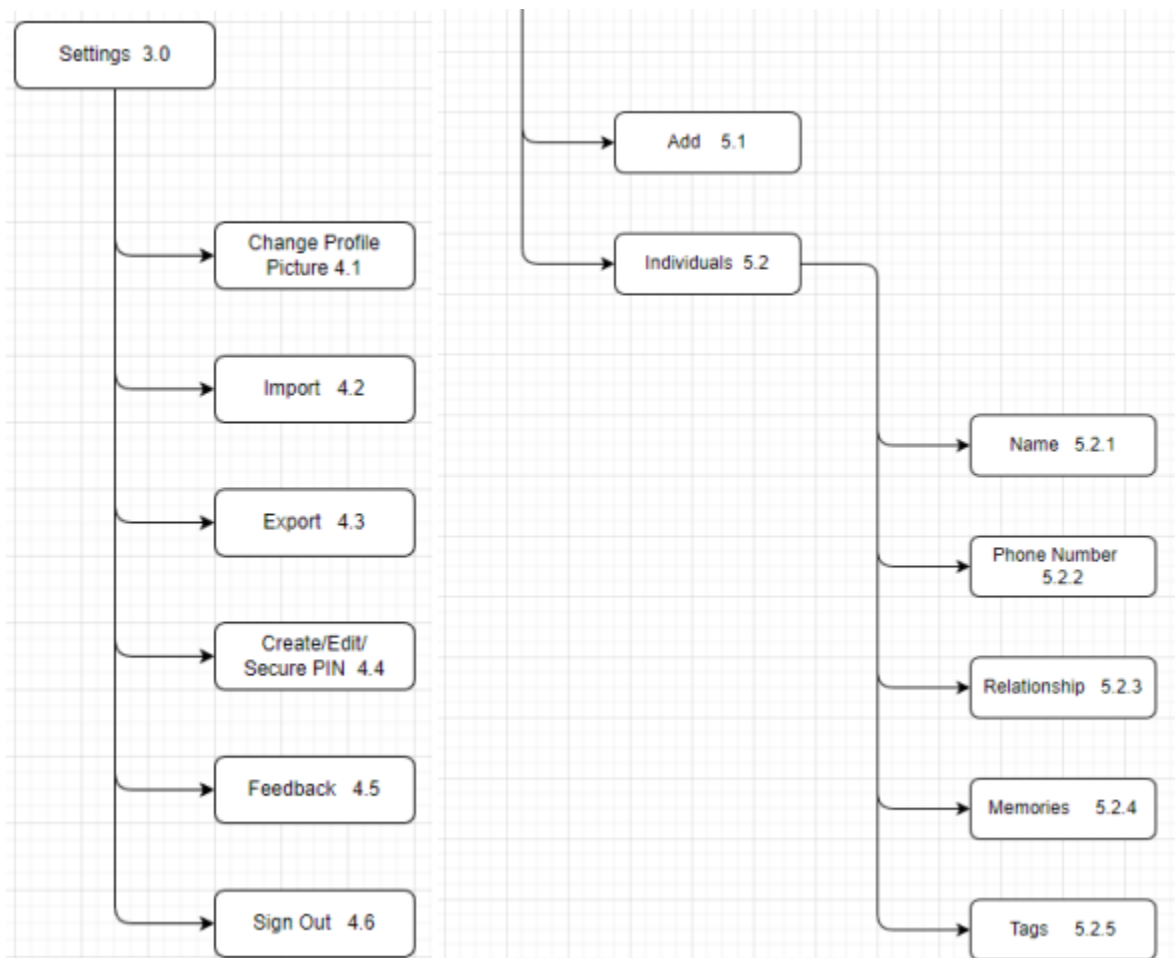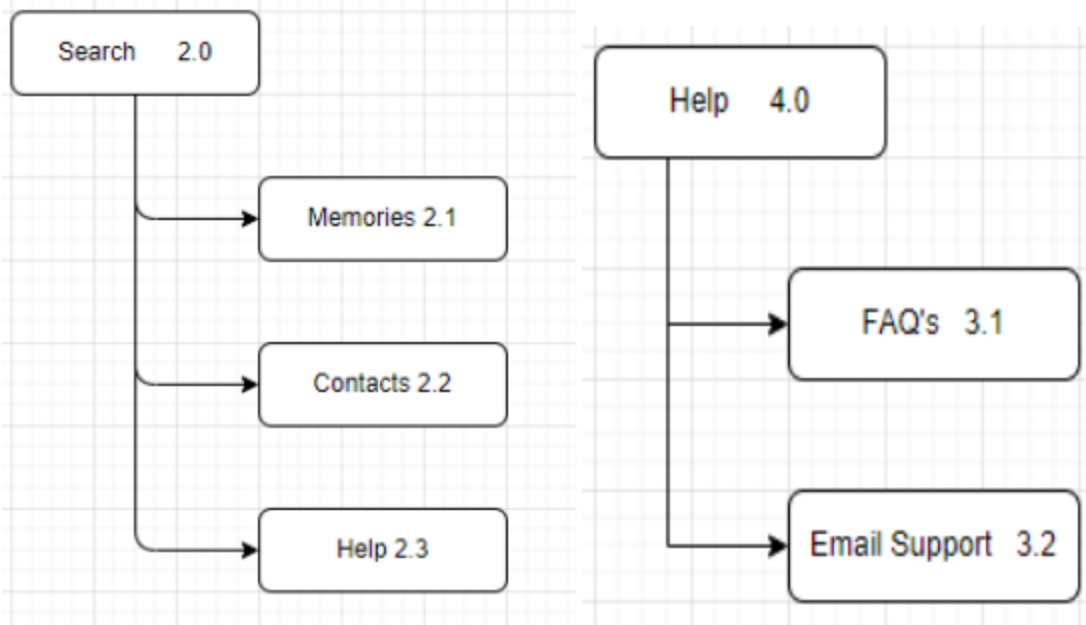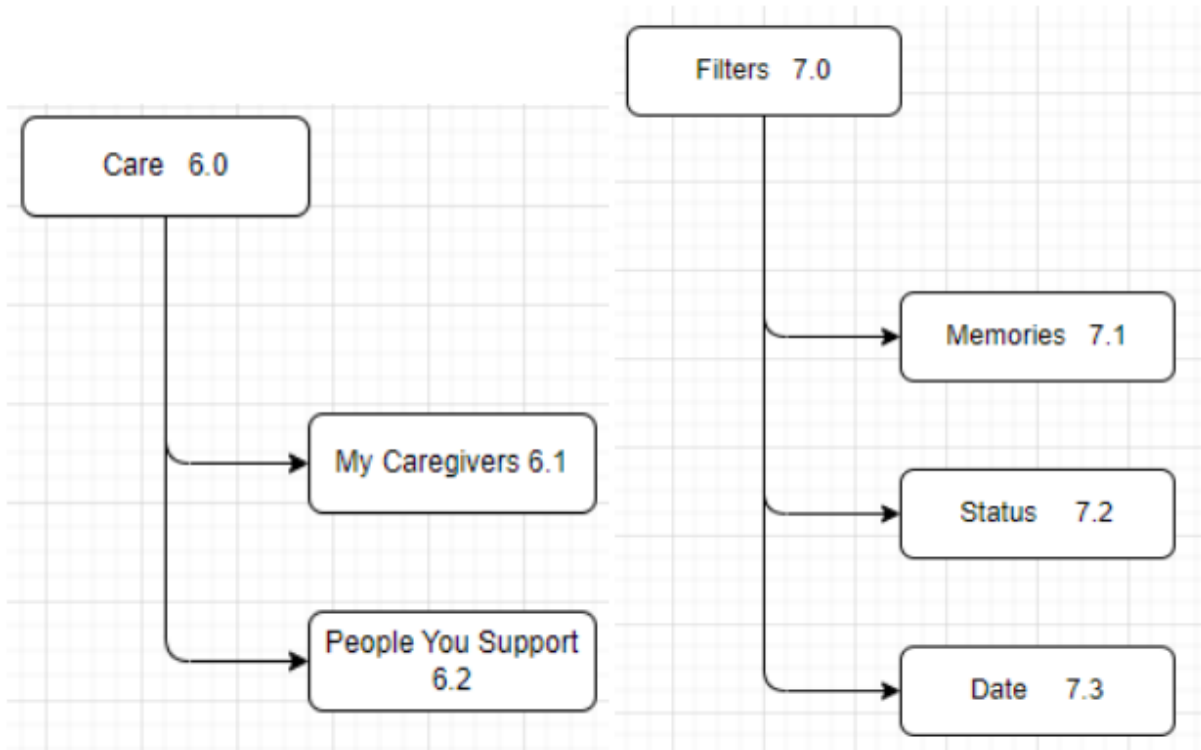
5. Policies and Tactics

5.1 Choice of which specific products used

Programming/Development:

- Firebase emulator for data storage.
- HTML for frontend development.
- Tailwind CSS for the design and to speed up the development process by writing less code.
- Visual Studio Code is the IDE for open-source code editing.
- Javascript/React is used for implementing functions, and React is a framework for web teams to maintain states in components.
- React Native as a framework for mobile team front-end development.
- Android Studio is the IDE for Android phone app development
- Xcode for Apple platforms

5.2 Plans for ensuring requirements traceability

Source Control:

GitHub

Jira (Agile Tracking)

Jira provides a structured and centralized platform for tracking requirements, tasks, and issues.

Its flexibility allows adaptation to the agile development approach Want2Remember may follow.

These decisions in the choice of specific products and plans for ensuring requirements traceability contribute to the efficiency, maintainability, and collaboration aspects of the Want2Remember web application without imposing significant architectural changes.

5.3 Plans for testing the software

In the context of the software, various engineering trade-offs were considered to balance conflicting objectives and constraints. It involved the choice between optimizing for performance and ensuring a smooth user experience. This decision impacted the selection of data storage solutions, such as Firestore, for efficient real-time data synchronization in both mobile and web applications. Firestore offers real-time updates but might come with associated costs in terms of database operations. To maintain code quality and consistency across the development team, comprehensive coding guidelines and conventions were established. This included adherence to ESLint for static code analysis and Prettier for code formatting. Each subsystem, module, or subroutine within "want2remember" followed a well-defined protocol to facilitate seamless communication and interoperability. This included standardized input and output formats, error-handling procedures, and clear documentation for function usage. A proactive approach to software maintenance was adopted, emphasizing regular updates, bug fixes, and feature enhancements. A clear plan for version control using tools like Git was established, ensuring a systematic approach to tracking changes and rolling out updates. The design of "want2remember" involved defining interfaces for various stakeholders. End-user interfaces were crafted to provide an intuitive experience on both mobile and web platforms.

6. Detailed System Design

6.1.1 Responsibilities

The Memory Creation Module within the "want2remember" system serves as a pivotal component dedicated to facilitating the creation of memories for patients dealing with cognitive impairment. This module is designed to empower clients to capture and store significant moments, events, or information they want to remember. It plays a crucial role in enhancing the quality of life for individuals facing memory challenges. The software includes a memory input interface, timestamp and location integration, reminder, and personalization features. The memories screen interface allows users to add memories in different categories, aiming for an intuitive and user-friendly interface for the memories screen.

6.1.2 Constraints

Currently, some of the data is getting read directly from the firestore which causes inefficiency.

6.1.3 Composition

We used GitHub as the centralized repository for storing the source code and project documentation. It ensures versioned and collaborative development, allowing team members to access, contribute, and review code changes. It utilizes branches for parallel development, bug fixes, and feature implementations. The pull request feature in Git is leveraged for code review and collaboration among team members. We also used the Agile framework for task management, sprint planning, and iterative development.

6.1.4 Uses/Interactions

The User will be able to see all the memories created and edit them, as well as filter and search them

6.1.5 Resources

6.1.6 Interface/Exports

6.2 MemoryScreen.js

6.2.1 Responsibilities

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

6.2.2 Constraints

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.

6.2.3 Composition

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

6.2.4 Uses/Interactions

6.2.5 Resources

6.2.6 Interface/Exports

6.3 CreateScreen.js

6.3.1 Responsibilities

Will allow the user to create a new memory.

6.3.2 Constraints

The Create Memory button will be easily accessible on the bottom toolbar and in the memories screen.

6.3.3 Composition

6.3.4 Uses/Interactions

After clicking the Create Memory button, the user will go to the CreateScreen, where they will fill out a form consisting of different memories names, categories, any files to attach, statuses, tags, etc.

6.3.5 Resources

6.3.6 Interface/Exports

6.4 MoreDetailScreen.js

6.4.1 Responsibilities

MoreDetailScreen will show the user all the details of the memory as well as the option to edit the information

6.4.2 Constraints

User will be able to access the MoreDetailScreen by simply tapping the memory

6.4.3 Composition

6.4.4 Uses/Interactions

User will be able to see details of memory such as the date it was created and updated, and the user will be able to edit the memory information by clicking the edit button in the memory moreDetailScreen

6.4.5 Resources

6.4.6 Interface/Exports

6.5 ContactsScreen.js

6.5.1 Responsibilities

ContactsScreen will show the user all contacts they have saved

6.5.2 Constraints

User will have to save the contacts before they appear on the contacts screen

6.5.3 Composition

6.5.4 Uses/Interactions

User will be able to click on the contacts button in the toolbar and be redirected to the ContactsScreen screen and see all their contacts they have saved

6.5.5 Resources

6.5.6 Interface/Exports

6.6 ContactsDetailsScreen.js

6.6.1 Responsibilities

ContactDetailsScreen will show the information of contacts the user has saved

6.6.2 Constraints

ContactDetailsScreen will only be available for contacts the user has saved

6.6.3 Composition

6.6.4 Uses/Interactions

ContactsDetailsScreens will be available through the ContactsScreen, where user will be able to see all the contact information

6.6.5 Resources

6.6.6 Interface/Exports

7. Detailed Lower level Component Design

7.1 QuickLookComponent

7.1.1 Classification
The kind of component, such as a subsystem, class, package, function, file, etc.

7.1.2 Processing Narrative (PSPEC)
A process specification (PSPEC) can be used to specify the processing details

7.1.3 Interface Description

7.1.4 Processing Detail

7.1.4.1 Design Class Hierarchy
Class inheritance: parent or child classes.

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
7.1 Detail.js

7.1.1 Classification
The kind of component, such as a subsystem, class, package, function, file, etc.

7.1.2 Processing Narrative (PSPEC)
A process specification (PSPEC) can be used to specify the processing details

7.1.3 Interface Description

7.1.4 Processing Detail

7.1.4.1 Design Class Hierarchy
Class inheritance: parent or child classes.

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

7.1 ContactQuickLook.js

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

8. Database Design

We use Cloud Firebase as our database for this project. The mobile app for Want 2 Remember also uses Firebase, allowing for compatibility. Cloud Firebase is a document-based 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 user's collection stores more information about the user, including private user data

9. User Interface

9.1 Overview of User Interface
The user functionality of want2remember is designed to be intuitive, accessible, and customizable across both platforms (web & mobile). Users can seamlessly create memories, manage contacts, and personalize their experience. Clear and concise feedback ensures that users are informed about the success of their actions and alerted to any issues that may arise during interaction with the system. The combination of local and cloud storage ensures continuous access to memories, providing a reliable and user-friendly experience.

9.2 Screen Frameworks or Images

Change Color Palette ✕

Custom ☑

Generic Memory   Interaction

Simple Memory   ToDo

Appointment   Simple ToDo

Credit Card   Website Password

Bank Account   Simple Password

Purchase

Default ☐

Apply

Select Filters

✕
Close

Memories	Status	Date

Early	Continuous

		Sun	Mon	Tue	Wed	Thu	Fri	Sat

All Days

◄ November ⌄ 2023 ⌄ ►

Today

Yesterday

This Week

		Sun	Mon	Tue	Wed	Thu	Fri	Sat

| 29 | 30 | 31 | 1 | 2 | 3 | 4 |

Last Week

| 5 | 6 | 7 | 8 | 9 | 10 | 11 |

This Month

| 12 | 13 | 14 | 15 | 16 | 17 | 18 |

Last Month

| 19 | 20 | 21 | 22 | 23 | 24 | 25 |

| 26 | 27 | 28 | 29 | 30 | 1 | 2 |

Apply

Want2Remember

👤

Template Builder

⌂ Home

＋ Create Memories

✎ Create Memory Templates

▣ Contacts

🛡 Care Support

Select a template to edit from the dropdown menu:

No template selected yet... ▾

Add, Remove Or Reorder the Components

🎨

Available Components   New Template Format

Categories	＋
Category Component	

teamhappy
Pre H H, Suf

Text   Call   Email

User Info

Job Title	Job
Department	Depart
Company	Company
Relationship	Me

Phone Numbers

P label	(909) 999-9999

Emails

HH	hh@gmail.com

Addresses

White House	1600 Pennsylvania Avenue NW, Washington, DC 20500, USA

URLs

www.w2link.com

Note

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the lean into electronic typesetting

---

11:14

Create Memory

Change Color Palette

Grid ◯ List

Generic Memory ⊞

Interaction ⊞

Simple Memory ⊞

ToDo ⊞

Appointment ⊞

Simple ToDo ⊞

Home   Create   Search   Contacts

11:13

Create Memory

Change Color Palette

Grid ◯ List

Generic Memory ⊞	Interaction ⊞
Simple Memory ⊞	ToDo ⊞
Appointment ⊞	Simple ToDo ⊞
Credit Card	Website Password

Home   Create   Search   Contacts

11:24

Cancel   **People**   Save

Add New Contact +

Search 🔍

B Bobby

Associated Memories

Memory 4	Memory 3	Nice Memc
Generic Memory	Generic Memory	Generic Mem

ADD

View Contact Information

Jane Doe

Associated Memories

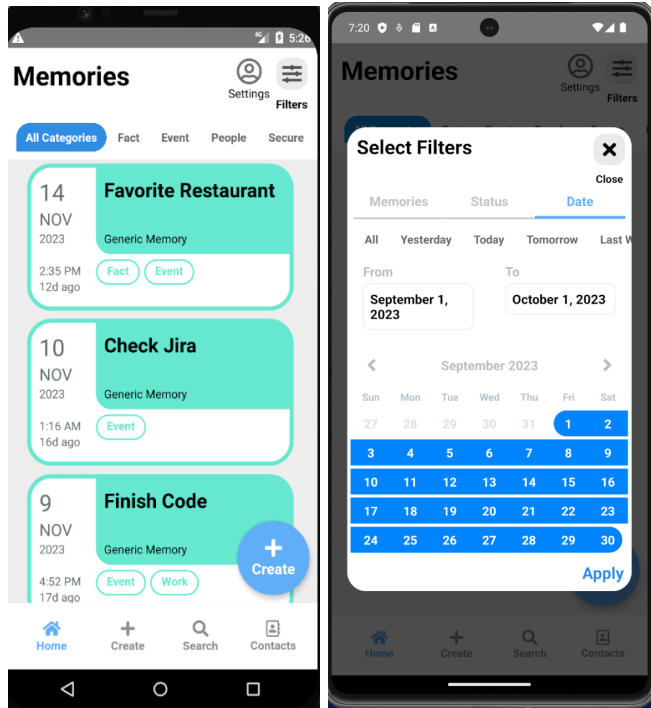| Memory 3	Memory 1
Generic Memory	Generic Memory

ADD

View Contact Information

John Smith

9.3 User Interface Flow Model

A discussion of screen objects and actions associated with those objects. This should include a flow diagram of the navigation between different pages.

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

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

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

- CEO = Chief Executive Officer
- Liason = a person who establishes and maintains communication for mutual understanding and cooperation
- CSULA = California State University of Los Angeles
- HTML = Hypertext Markup Language
- UI = User Interface
- VS = Visual Studio
- CSS = Cascading Style Sheet

12. References

Brad Appleton <brad@bradapp.net> http://www.bradapp.net

https://www.cs.purdue.edu/homes/cs307/ExampleDocs/DesignTemplate_Fall08.doc

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