

Software Design Document

for Want2Remember

Version 2.0 approved

Prepared by

Antonio Campos

Project Co-Lead

Alec Kaczmarek

Project Co-Lead

Amy Guttman

Documentation Co-Lead

Alexandra Strong

Documentation Co-Lead

Vincent Li

Engineer

Saiyang Liu

Engineer

Ricardo Marroquin

Engineer

Miguel Nonoal-Garcia

Engineer

Jonathan Sum

Engineer

Edwin Zapata Minero

Engineer

Zilong Ye

Faculty Advisor

Kevin Benavente

We2Link Liaison

Michael Malone

We2Link CEO

10 December 2021

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
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
Architectural Strategies.....	8
System Architecture.....	11
4.1.	11
4.2.	11
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.....	<pg #>
5.5. Guidelines and conventions.....	<pg #>
5.6. Protocols.....	<pg #>
5.7. Maintaining the software.....	<pg #>
5.8. Interfaces.....	<pg #>
5.9. System's deliverables.....	<pg #>
5.10. Abstraction.....	<pg #>
Detailed System Design.....	14
6.x Name of Module.....	<pg #>
6.x.1 Responsibilities.....	<pg #>
6.x.2 Constraints.....	<pg #>
6.x.3 Composition.....	<pg #>
6.x.4 Uses/Interactions.....	<pg #>
6.x.5 Resources.....	<pg #>
6.x.6 Interface/Exports.....	<pg #>
Detailed Lower level Component Design.....	16
7.x Name of Class or File.....	<pg #>
7.x.1 Classification.....	<pg #>
7.x.2 Processing Narrative(PSPEC).....	<pg #>
7.x.3 Interface Description.....	<pg #>
7.x.4 Processing Detail.....	<pg #>
7.x.4.1 Design Class Hierarchy.....	<pg #>
7.x.4.2 Restrictions/Limitations.....	<pg #>
7.x.4.3 Performance Issues.....	<pg #>
7.x.4.4 Design Constraints.....	<pg #>

7.x.4.5 Processing Detail For Each Operation.....	<pg #>
Database Design	17
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
Amy Guttman, Alexandra Strong	8/24/2021	Initial Access	1.0
Amy Guttman, Alexandra Strong	10/26/2021	Cross-Reference with Past SRS	1.1
Amy Guttman, Alexandra Strong	12/10/2021	First Draft	2.0

1. Introduction

1.1 Purpose

Want2Remember is a mobile 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 detail 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:

- Redux
- NodeJS
- JavaScript
- React Native
- 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 (iOS 7 or above; Android Oreo (8) or above).

2.2 General Constraints

The major hurdles associated with this project are:

- Learning the JavaScript mobile application framework, React Native
- Working as a group remotely during the COVID-19 pandemic

We were given access to the Udemy course “React Native - The Practical Guide” 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 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.

2.3 Goals and Guidelines

Want2Remember is a mobile phone 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, 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 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 2022.

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 React Native: Developing with the React Native framework gives us the advantage of developing simultaneously for Android and iOS projects. The code is written in JavaScript but is then rendered with native code. React Native also puts on 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 React Native such as a multi-platform codebase, native app development, and fast refresh as soon as new code is saved.
- 3.1.2 Redux: We use Redux for data operations.
- 3.1.3 Github: We use Github for version control.

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 refactoring their codebase and extending functionalities.

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 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 (may or may not be touch 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

TBD

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 Test Fairy for Android and Test flight for iOS.

4. System Architecture

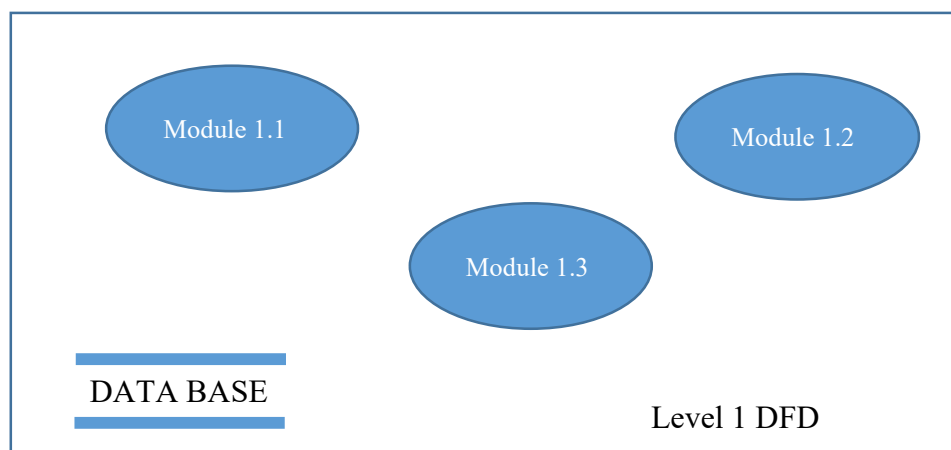
This section should provide a high-level overview of how the functionality and responsibilities of the system were partitioned and then assigned to subsystems or components. Do not go into too much detail about the individual components themselves (there is a subsequent section for detailed component descriptions). The main purpose here is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together to provide the desired functionality.



This is where the level 0 DFD will probably work best.

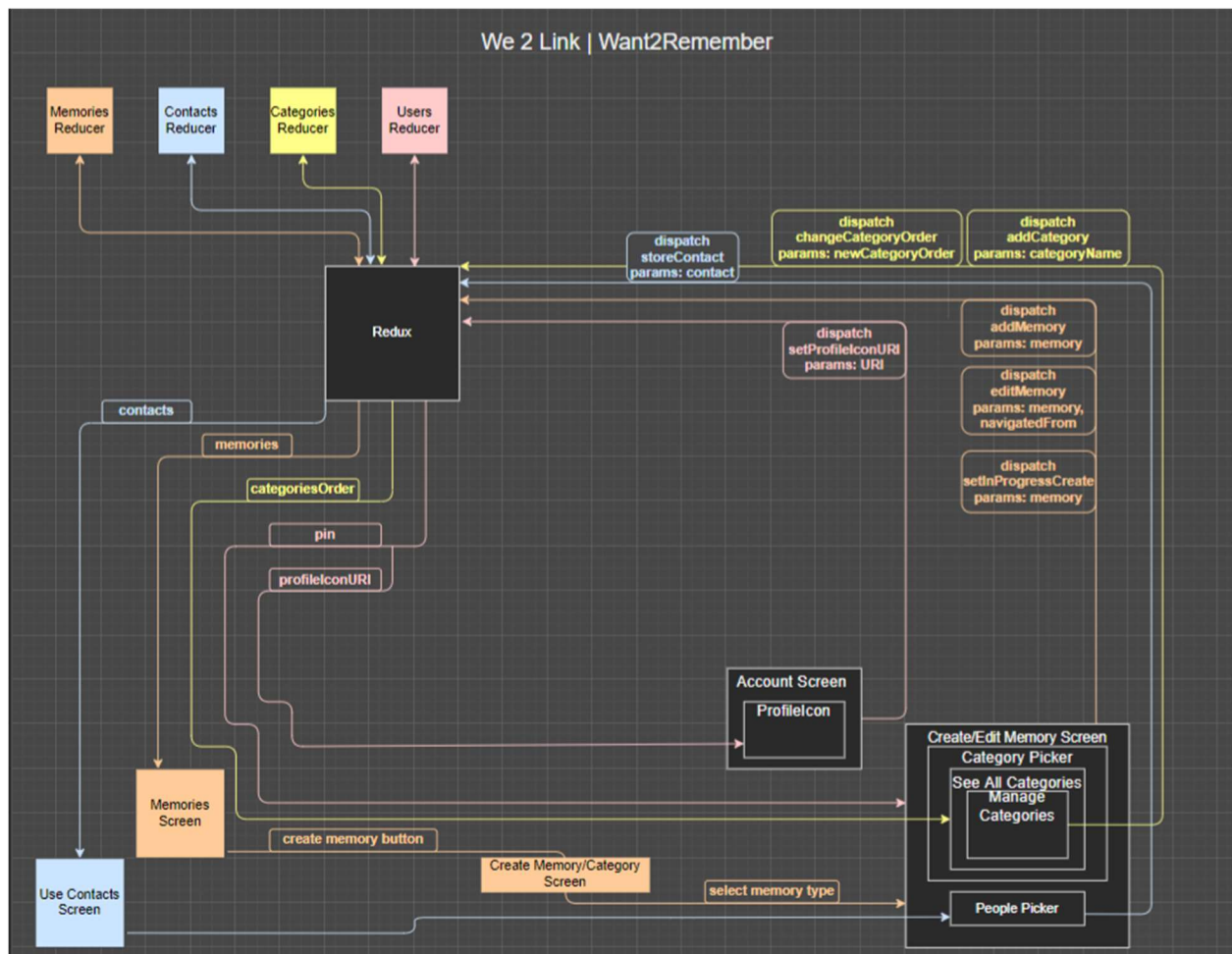
At the top-most level, describe the major responsibilities that the software must undertake and the various roles that the system (or portions of the system) must play. Describe how the system was broken down into its modules/components/subsystems (identifying each top-level modules/component/subsystem and the roles/responsibilities assigned to it).

Each subsection (i.e. “4.1.3 The ABC Module”) of this section will refer to or contain a detailed description of a system software component.



Level 1 Data Flow Diagrams (DFD) and Control Flow Diagrams (CFD) should probably go here.

Describe how the higher-level components collaborate with each other in order to achieve the required results. Don't forget to provide some sort of rationale for choosing this particular decomposition of the system (perhaps discussing other proposed decompositions and why they were rejected). Feel free to make use of design patterns, either in describing parts of the architecture (in pattern format), or for referring to elements of the architecture that employ them. Diagrams that describe a particular component or subsystem in detail should be included within the particular subsection that describes that component or subsystem.



Redux Flow Diagram

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
- Mobile Application Framework: React Native

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 beta testing this application. We use a combination of Test Fairy for Android and Test flight for iOS to gather feedback from the beta users. The issues generated from user feedback is pushed onto our JIRA board for further examination and response.

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 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

TBD

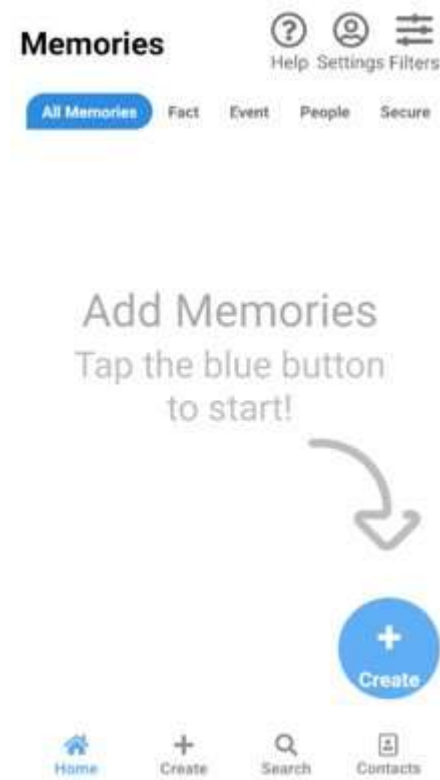
9. User Interface

The user interface is the application, from the point of view of the users. Do your classes and their interactions (the logical and process views) impose restrictions on the user interface? Would removing some of these restrictions improve the user interface? Use some form of user interface flow model to provide an overview of the UI steps and flows. Don't go into too much refinement. You should include screen shots or wireframe layouts of significant pages or dialog elements. Make sure to indicate which of the system level modules or components that each of these user interface elements is interacting with.

9.1 Overview of User Interface

Describe the functionality of the system from the user's perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback Information that will be displayed for the user. This is an overview of the UI and its use. The user manual will contain extensive detail about the actual use of the software.

9.2 Screen Frameworks or Images



Memories

Help Settings Filters

All Memories Fact Event People Secure

5
DEC

Dinner

Simple Memory In Progress

in 4 days
8:51:00 PM

Food Drinks Work

18
NOV

Group Meeting

Interaction Delayed

13 days ago
5:53:00 PM

People Event

1
DEC

GM

Generic Memory

+

Create

Home Create Search Contacts

←

Bank Account

Enter Title

Select a category

See All

Fact

Event

People

Secure

Files

Bank Name

Bank Name

Routing Number

Routing Number

Account Number

Account Number

Pin

Pin

Date Created

December 2, 2021

Calendar

Bank Website

Bank Website

←

Generic Memory

Enter Title

Select a category

See All

Fact

Event

People

Secure

Files

Status

Pending

In Progress

Delayed

Stopped

Done

Tags

ADD TAGS

Add Tag +

Date

December 2, 2021

Calendar

Time

12:09 AM

Time

Reminder Date

None

Calendar

Reminder Time

None

Time

←

ToDo

Enter Title

Select a category

See All

Fact

Event

People

Secure

Files

Status

Pending

In Progress

Delayed

Stopped

Done

Tags

ADD TAGS

Add Tag +

Date

December 2, 2021

Calendar

Starting Time

12:11 AM

Time

Ending Time

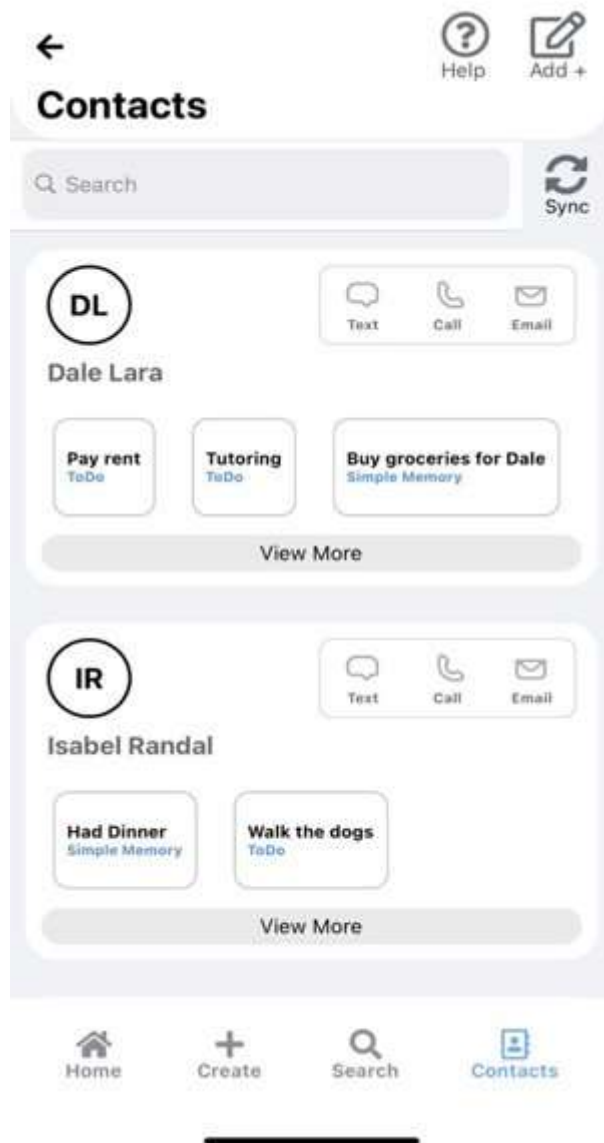
12:11 AM

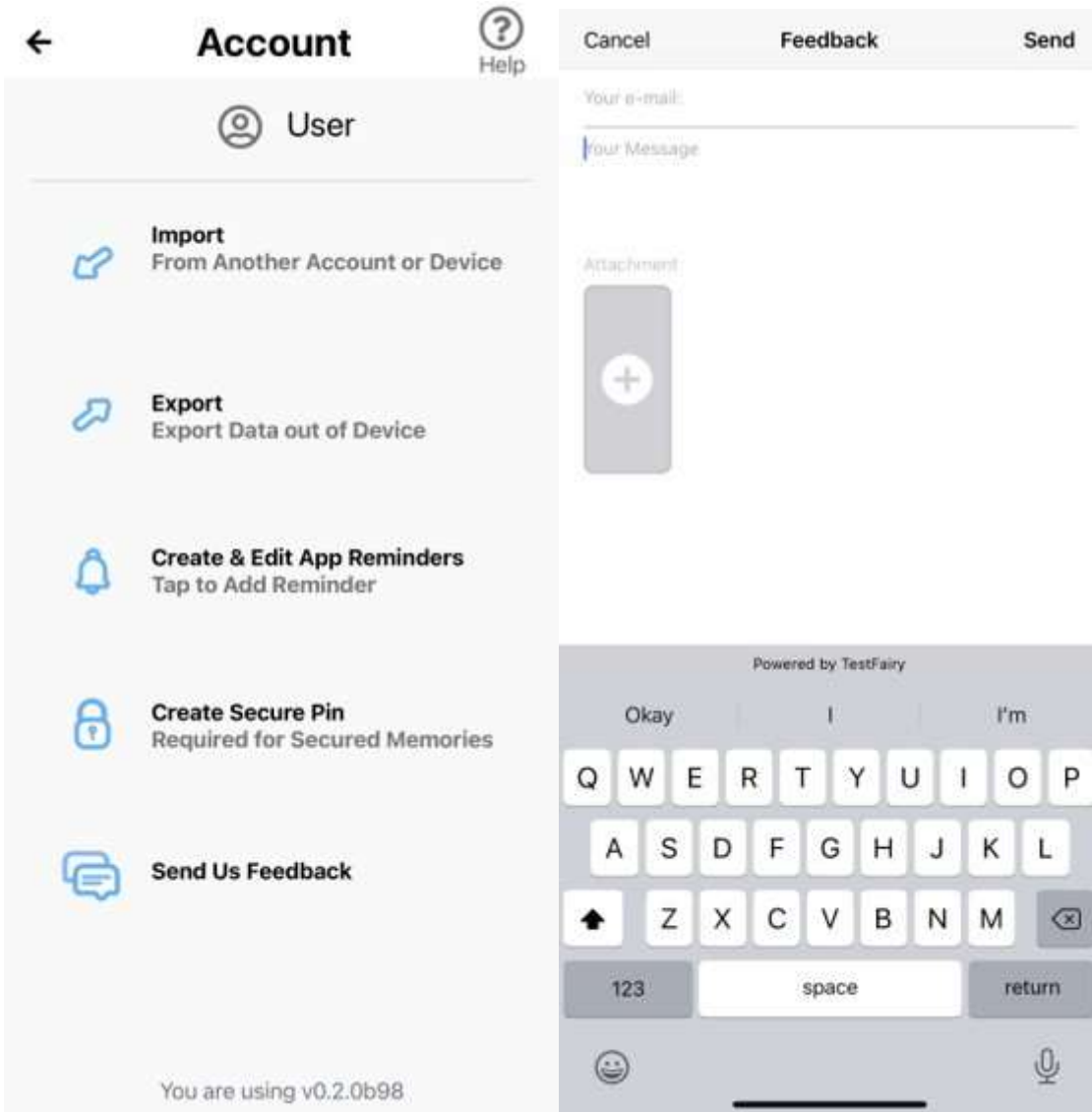
Time

Reminder Date

None

Calendar







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.

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) August 24, 2021

Title: React Native - The Practical Guide

Authors: Academind by Maximilian Schwarzmüller, Maximilian Schwarzmüller

Date: (First Accessed) August 31, 2021

<https://www.udemy.com/course/react-native-the-practical-guide/>

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: May 13, 2021

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 10, 2021