Want2Remember SDD

CS3338 Group 3: Joshua Hanscom, Nicholas Montales, Jessie Guijosa, Perla Reyes

20 November 2024

Contents

Version Description

Version	Description of Changes	Date
1.0	Initial version of the Software Design Document.	20 November 2024

1 Introduction

1.1 Purpose of the Document

Want2Remember is a web application that assists those with cognitive impairments in remembering day-to-day tasks and memories. The system keeps track of user-generated data, such as memories, to-do lists, appointments, contacts, and payment information. This document explains the functionality, design, architecture, and requirements of the Want2Remember application.

1.2 Intended Audience

This document is intended for the California State University of Los Angeles (CSULA) Computer Science department. It is part of the Senior Design Project sponsored by We2Link. The audience includes professors, students of CSULA, and employees of We2Link.

1.3 Overview of the System

Want2Remember uses a component-based architecture to enhance reusability and reduce the size and complexity of the codebase.

2 System Architecture

2.1 Workflow of the System

The Want2Remember app allows users to store and create memories while enabling caregivers to provide support. User data is stored in Firebase. Users can create and edit memories, which are synchronized in real-time with Firebase. Caregivers can also interact with the app to assist users.

2.2 System Components

Client-Side: The client side uses React for the web interface and React Native for the mobile app. Tailwind CSS ensures responsive and consistent styling. Key screens include: - **Caregivers screen**: Displays caregiver information and access. - **Memories screen**: Shows a list of memories and details. - **Account screen**: Allows users to manage settings. - **Create Memory screen**: Facilitates adding new memory details (e.g., category, date, location).

Server-Side: Firebase serves as the backend, providing real-time data storage and synchronization. Firebase Authentication ensures secure user access. React JS components handle interactions with Firebase, ensuring seamless updates.

3 User Interface

3.1 How to Use the System

Users interact through various screens, including login, dashboard, and memory creation. Real-time data updates are stored in Firestore. Features include filtering and searching memories by category, type, and timeframe.

3.2 Database Design

The app utilizes Firestore, organizing data into collections for efficient management: - **publicUsersInfo**: Contains less sensitive data (e.g., UUIDs). - **users**: Stores private user data. Firebase Authentication ensures secure data access. Real-time updates maintain consistency.

4 Glossary

Acronym	**Definition**
SDD	Software Design Document
UI	User Interface
UUID	Universally Unique Identifier
API	Application Programming Interface
JS	JavaScript
CSS	Cascading Style Sheets