Software Requirements Specification (SRS)

University of Maryland Global Campus SWEN 670

Team B

Version 1.2

**Document Control**

# Document History

|  |  |  |
| --- | --- | --- |
| **Version** | **Issue Date** | **Changes** |
| 1.0 | 8/27/2023 | Initial Draft. |
| 1.1 | 9/23/2023 | Added and modified Use Cases, added UML diagrams |
| 1.2 | 9/30/2023 | Address feedback from Milestone 2 delivery – add table and figure captions |
|  |  |  |

Table of Contents

[Document History 2](#_Toc146963422)

[1. Introduction 6](#_Toc146963423)

**[1.1 Purpose](#_Toc146963424)** [6](#_Toc146963424)

**[1.2 Scope](#_Toc146963425)** [6](#_Toc146963425)

**[1.3 Definitions, Acronyms and Abbreviations](#_Toc146963426)** [7](#_Toc146963426)

**[1.4 Project Documents](#_Toc146963427)** [7](#_Toc146963427)

**[1.5 References](#_Toc146963428)** [8](#_Toc146963428)

**[1.6 Overview](#_Toc146963429)** [9](#_Toc146963429)

[2. Overall Description 9](#_Toc146963430)

**[2.1. Use-Case Model Survey](#_Toc146963431)** [10](#_Toc146963431)

**[2.2. Use-Case Diagram](#_Toc146963432)** [10](#_Toc146963432)

**[2.3 Assumptions and Dependencies](#_Toc146963433)** [11](#_Toc146963433)

[3. Specific Requirements 12](#_Toc146963434)

**[3.1. Use-Case Reports](#_Toc146963435)** [12](#_Toc146963435)

**[3.1.1 Accessing the application.](#_Toc146963436)** [12](#_Toc146963436)

**[3.1.2 Interacting with the Virtual Assistant](#_Toc146963437)** [14](#_Toc146963437)

**[3.1.3 Create an audio recording](#_Toc146963438)** [17](#_Toc146963438)

**[3.1.4 Utilize the search function](#_Toc146963439)** [21](#_Toc146963439)

**[3.1.5 Accessing the recent questions/request](#_Toc146963440)** [23](#_Toc146963440)

**[3.1.6 View the menu](#_Toc146963441)** [25](#_Toc146963441)

**[3.1.7 Access the user login credential](#_Toc146963442)** [26](#_Toc146963442)

**[3.1.8 User registration application](#_Toc146963443)** [28](#_Toc146963443)

**[3.1.9 View or modify the user's registration](#_Toc146963444)** [31](#_Toc146963444)

**[3.1.10 Utilize the guided tour](#_Toc146963445)** [36](#_Toc146963445)

**[3.2. Supplementary Requirements](#_Toc146963446)** [40](#_Toc146963446)

**[3.2.1 Accessibility](#_Toc146963447)** [40](#_Toc146963447)

**[3.2.2 Compatibility](#_Toc146963448)** [40](#_Toc146963448)

**[3.2.3 Security](#_Toc146963449)** [40](#_Toc146963449)

**[3.2.4 Legal](#_Toc146963450)** [40](#_Toc146963450)

[4. Non-Functional Requirements 40](#_Toc146963451)

**[4.1 Performance:](#_Toc146963452)** [41](#_Toc146963452)

**[4.2 Security:](#_Toc146963453)** [42](#_Toc146963453)

**[4.3 Reliability:](#_Toc146963454)** [42](#_Toc146963454)

**[4.4 Scalability:](#_Toc146963455)** [42](#_Toc146963455)

Table of Tables

[Table 1: Definitions and Acronyms 6](#_Toc146962670)

[Table 2: Project Documents 7](#_Toc146962671)

[Table 3: Use-Case Model Survey 9](#_Toc146962672)

[Table 4: Accessing the Application Use Case 12](#_Toc146962673)

[Table 5: Interacting with the Virtual Assistance Use Case 15](#_Toc146962674)

[Table 6: Virtual Assistance Alternate Action Use Case 16](#_Toc146962675)

[Table 7: Create Audio Recording Use Case 20](#_Toc146962676)

[Table 8: Create Recording Alternate Action Use Case 20](#_Toc146962677)

[Table 9: Search Function Use Case 22](#_Toc146962678)

[Table 10: Recent Questions Use Case 24](#_Toc146962679)

[Table 11: View Menu Use Case 25](#_Toc146962680)

[Table 12: User Login Use Case 27](#_Toc146962681)

[Table 13: User Registration Use Case 30](#_Toc146962682)

[Table 14: Modify User Registration Use Case 34](#_Toc146962683)

[Table 15: Modify User Registration Alternate Action Use Case 34](#_Toc146962684)

[Table 16: Guided Tour Use Case 38](#_Toc146962685)

Table of Figures

[Figure 1: CogniOpen UML Diagram 11](#_Toc146963081)

[Figure 2: Accessing the Application UML Diagram 12](#_Toc146963082)

[Figure 3: Interacting with Virtual Assistant UML Diagram 14](#_Toc146963083)

[Figure 4: Create Audio Recording UML Diagram 18](#_Toc146963084)

[Figure 5: Search Function UML Diagram 22](#_Toc146963085)

[Figure 6: Recent Questions UML Diagram 24](#_Toc146963086)

[Figure 7: Accessing the Application UML Diagram 27](#_Toc146963087)

[Figure 8: User Registration UML Diagram 29](#_Toc146963088)

[Figure 9: Update Registration UML Diagram 32](#_Toc146963089)

[Figure 10: Guided Tour UML Diagram 36](#_Toc146963090)

# Introduction

The CogniOpen App is an innovative app designed to help those who suffer from short-term memory loss and other conditions related to it. Even those who are healthy tend to forget where they put things or specifics of a recent conversation in today's fast-paced society when multitasking is the norm. This app intends to close this gap by providing help and support to such people using modern technology, AI services, and user-friendly designs.

## **1.1 Purpose**

The primary objective of this document is to provide a comprehensive and in-depth specification of the software requirements for the CogniOpen App. The goal of the document is to list and describe the functional, technical, and user requirements for the app. This document can be used as a reference for stakeholders, such as developers, project managers, testers, and end users, at any time during the development lifecycle.

## **1.2 Scope**

The Flutter framework will be used to create the cross-platform mobile application known as the CogniOpen App. The application will mainly:

* Use the camera on the device to record live events.
* Recognize and record surrounds, surroundings, and things.
* Manage and store large video streams while allowing users to browse through them.
* Summarize chats so consumers can evaluate the main points of their conversations.
* Provide alerts and reminders for frequently overlooked details or objects.
* Use AI services to improve the features, assuring prompt and accurate responses.

The application wants to become a digital companion for users by helping them recall and remember facts that they might have forgotten, such as the location of a basic object or an important discussion.

## **1.3 [Definitions, Acronyms and Abbreviations](file:///C:/Users/gebre/Downloads/ViroTour_SRS_Team_A.docx" \l "_TOC_250024)**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| AI | Artificial Intelligence |
| API | Application Programming Interface: a method for different applications to communicate with each other. |
| App | Application: a program or piece of software designed to fulfill a particular purpose. |
| ChatGPT | An AI language model, designed to understand and generate human-like text. |
| Dart | Dart is a client-optimized programming language for developing fast apps on any platform. |
| Flutter | A programming framework designed to build, test, and deploy mobile, web, desktop, and embedded apps from a single codebase. |
| Mobile Device | A smart phone, tablet, or other portable device, typically with an Android or IOS operating system. |
| OS | Operating System |
| STT | Speech to text |
| UI | User Interface: the components of a system that users can utilize to control the system. |
| UX | User Experience: the processes and behaviors through which a user controls and receives feedback from a system. |

Table 1: Definitions and Acronyms

## **1.4 Project Documents**

This SRS is part of a series of documents related to the Afterthought application. The documents are designed to provide individuals with the necessary information to maintain and update the project throughout its life cycle.

The listed documents will be included in the final version of this package:

|  |  |  |
| --- | --- | --- |
| Document Name | Document Version | Last Revision |
| Project Plan | 1.2 | 10/28/2023 |
| Software Requirements Specification | 1.2 | 10/28/2023 |
| Technical Design Document | 1.1 | 10/28/2023 |
| Software Test Plan | 1.1 | 10/28/2023 |
| Programmer Guide | 1.0 | 10/28/2023 |
| Deployment and Operations Guide | 1.0 | 10/28/2023 |
| Software Test Report | In Development |  |
| User Guide | In Development |  |
| Traceability Matrix | In Development |  |

Table 2: Project Documents

## **1.5 References**

Baecker, R., Moffatt, K., & Massimi, M. (2012). Technologies for aging gracefully. Interactions, 19(3), 32-36. <https://doi.org/10.1145/2168931.2168940>

Chen, Y., & Pu, P. (2014). Designing a memory assistance app for older adults. In Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare (pp. 305-308). ICST. <https://doi.org/10.4108/icst.pervasivehealth.2014.255046>

Gao, Y., & Li, H. (2011). A meta-analysis of the impact of mobile phone use on driver distraction and crash risk. In Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems (ITSC) (pp. 1656-1661). IEEE. <https://doi.org/10.1109/ITSC.2011.6083077>

Klimova, B., & Poulova, P. (2018). Challenges in designing mobile applications for people with mild cognitive impairment. In Proceedings of the 11th International Conference on Digital Information Management (ICDIM) (pp. 1-6). IEEE. <https://doi.org/10.1109/ICDIM.2018.8846997>

Asana. (n.d.). How to write a software requirement document (with template). <https://asana.com/resources/software-requirements-document>

Perforce Software. (2020). How to Write a Software Requirements Specification (SRS). <https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs>

Sharma, R. (2018). The Only Software Requirements Document Template You Need. <https://arkenea.com/blog/software-requirements-document-template/>

Almalki, F., Al-Thunayan, A., Al-Salman, A., Alarifi, A., & Al-Khalifa, H. (2019). Memory apps for dementia: A systematic review and meta-analysis. Journal of Medical Internet Research, 21(10), e15100. <https://doi.org/10.2196/15100>

Flutter. (n.d.). Flutter: Beautiful native apps in record time. <https://flutter.dev/>

Sharma, S. (2020). AI Services: A Comprehensive Guide. <https://www.analyticsvidhya.com/blog/2020/07/ai-services-a-comprehensive-guide/>

Accessibility in Software Development (n.d.). <https://www.interaction-design.org/literature/topics/accessibility-in-software-development>

Cross-Platform Mobile App Development (2020). <https://www.altexsoft.com/blog/mobile/cross-platform-mobile-app-development/>

Mobile App Security Best Practices (2020). <https://www.appknox.com/blog/mobile-app-security-best-practices>

GDPR Compliance Checklist for Mobile Apps (2019). <https://clearbridgemobile.com/gdpr-compliance-checklist-for-mobile-apps/>

How to Write Good Non-Functional Requirements (2019). <https://reqtest.com/requirements-blog/how-to-write-good-non-functional-requirements/>

Non-Functional Requirements: A Guide with Examples & Types (2020). <https://www.perforce.com/blog/alm/non-functional-requirements-guide-examples-types>

What are Non-Functional Requirements? Definition & Examples (n.d.). <https://www.visual-paradigm.com/guide/requirements-engineering/what-is-non-functional-requirement/>

## **1.6 Overview**

The following sections of the SRS will contain the Overall Description of the software application being developed and the Specific Requirements needed to realize the software and meet the Customer’s expectations. The Overall Description section is designed to cover the high-level overview of the software application including, but not limited to, a use-case model survey, use-case diagrams, and assumptions and dependencies. The Specific Requirements section of the SRS will outline the specific conditions needed to be met to achieve the application’s goals. This will include, but is not limited to, use-case reports, UML diagram table, supplementary requirements, and non-functional requirements.

# Overall Description

## **2.1. Use-Case Model Survey**

|  |  |
| --- | --- |
| **Use Case** | **Description** |
| 1. **Accessing the application** | Opening the “Home” page of the application, showcasing the six main features; an artificial intelligent Virtual Assistant, gallery, video recording, audio recording, search and recent questions/ request. |
| 1. **Interacting with the Virtual Assistant** | Gain access to the Artificial Intelligence (AI) feature that allows users to ask questions and receive information either by speaking or typing text. |
| 1. **Create an audio recording** | The process of creating and storing new audio recordings to the application. |
| 1. **Listen to an audio recording** | Display all saved audio recordings and listen to recorded audio. |
| 1. **Delete an audio recording** | The process of deleting an audio recording. |
| 1. **Utilize the search function** | The ability to search photos, videos and audio for trigger words. |
| 1. **Accessing the recent questions/request** | Display a list of previously asked questions/ requests. |
| 1. **View menu list** | Display a drop-down menu containing the application features. |
| 1. **Access the user login credential** | Display the first user interface screen, allowing users to login or to create an account. |
| 1. **User registration application** | The process of creating a new account. |
| 1. **View or modified the user’s profile** | Allowing users to view or easily edit their personal information. |
| 1. **Utilize the guided tour** | Display step-by-step instructions and visual aids. |

Table 3: Use-Case Model Survey

## **2.2. Use-Case Diagram**

A use case diagram is a type of behavioral diagram in Unified Modeling Language (UML) that represents the interactions between actors and a system. It is used to visually represent the various ways that users interact with a system. The diagram consists of actors, use cases, and the relationships between them. The relationships between actors and use cases show how the actors interact with the system to achieve their goals.

The UML diagram shows the relationship between the actor and the CogniOpen application.

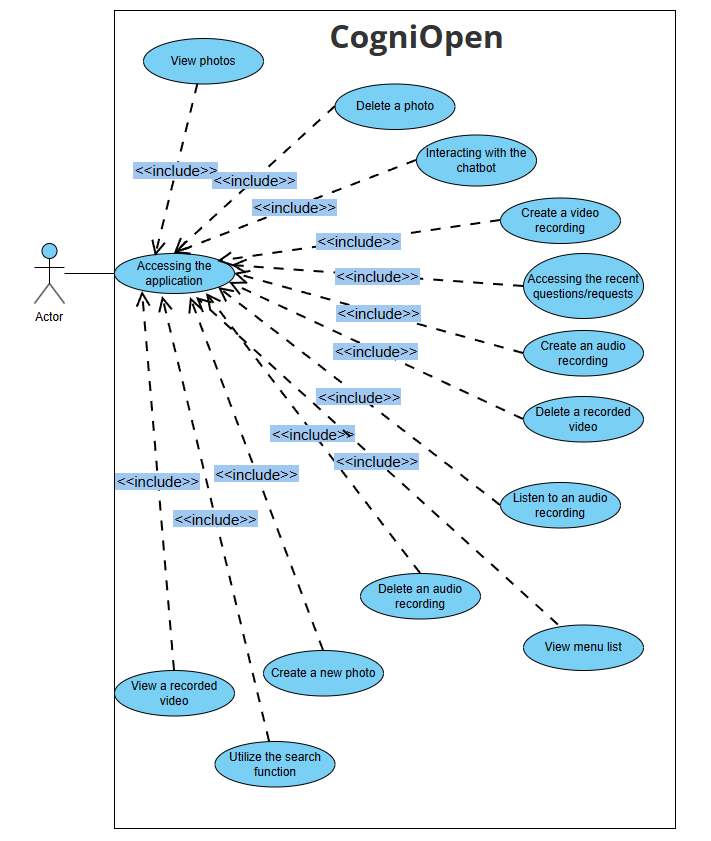


Figure 1: CogniOpen UML Diagram

## **2.3 Assumptions and Dependencies**

* The application must be user-friendly since each may feel like the user’s first time seeing the application
* The application will utilize ChatGPT
* The application will be developed in Flutter and Dart
* The application will utilize the mobile devices camera to capture images and videos
* The application will utilize the mobile devices microphone to record audio
* The application requires a network connection

# Specific Requirements

## **3.1. Use-Case Reports**

### **3.1.1 Accessing the application.**

**Summary:** The actor opens the application.

**UML Diagram:**

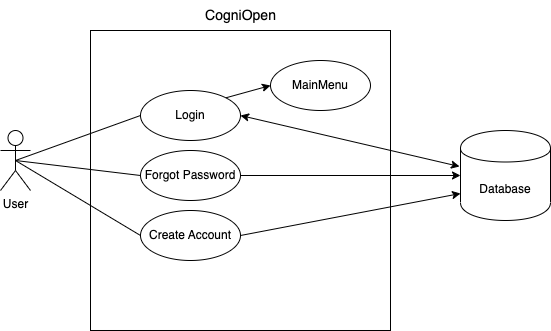


Figure 2: Accessing the Application UML Diagram

**Preconditions:** The actor installed the application on their device.

**Triggers:** The actor opens the application from their device.

**Basic course of events (main scenario):**

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Screen** |
| 1. The actor opens the application on their mobile device. |  |  |
|  | 2. The System displays the login screen for the application. |  |
| 3. The user enters their email and password. |  |  |
|  | 1. The system verifies the user credentials and logs the user into their account. The system will display the home screen. |  |

Table 4: Accessing the Application Use Case

### **3.1.2 Interacting with the Virtual Assistant**

**Summary:** The system's Virtual Assistant feature is designed to mimic human-like responses to a variety of prompts and inquiries and can be accessed either by speaking or typing your request.

**UML Diagram:**

A diagram of a system

Description automatically generated

Figure 3: Interacting with Virtual Assistant UML Diagram

**Preconditions:** The actor must register to use the application and sign in.

**Triggers:** The actor selects the voice command button.

**Basic course of events (main scenario):**

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Screen** |
| 1. The actor selects Virtual Assistant button from the home screen. |  |  |
|  | 2. The system opens the Virtual Assistant screen. |  |
| 3. The actor selects the speaker icon on the screen |  |  |
|  | 4. The system opens the phone’s microphone. |  |
| 5. The actor asks a question using their voice. |  |  |
|  | 6. The system displays the question on the screen and answers their question. |  |

Table 5: Interacting with the Virtual Assistance Use Case

**Post-conditions:** The Virtual Assistant answered the user's question and saved the questions to the recently asked questions list.

**Alternative course of events:**

**Asking a question by typing:**

|  |  |  |
| --- | --- | --- |
| 3. The actor selects the speaker icon on the screen |  |  |
|  | 4. The system opens the phone’s keyboard. |  |
| 5. The actor texts a question using the keyboard. |  |  |
|  | 6. The system displays the question on the screen and answers their question. |  |

Table 6: Virtual Assistance Alternate Action Use Case

**Post-conditions:** The Virtual Assistant answered the user's question and saved the questions to the recently asked questions list.



### **3.1.3 Create an audio recording**

**Summary:** The audio recording feature allows users to generate and store new audio recordings within the application. Once activated, the application opens the phone's microphone and displays an audio recording screen.

**UML Diagram:**

A diagram of a chatbot

Description automatically generated

Figure 4: Create Audio Recording UML Diagram

**Preconditions:** The application must have the user’s device permission to enable the phone’s microphone.

**Triggers:** The user presses the “Record Audio” button.

**Basic course of events (main scenario):**

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Screen** |
| 1. The actor selects the “Record Audio” button from the home screen. |  |  |
|  | **2.** The system opens the Audio screen. |  |
| **3.** The actor presses the red circle button to start recording. |  |  |
|  | **4.** The system begins recording audio and initiates a timer for the audio recording. |  |
| **5.** The actor presses the red square button to stop the recording. |  |  |
|  | **6.** The system stops the audio recording, opens a new window, and gives the actor the option to save or cancel the audio recording. |  |
| **7.** The actor presses the save button. |  |  |
|  | **8.** The system saves the recorded audio to the application gallery section. The system returns to the home screen. |  |

Table 7: Create Audio Recording Use Case

**Postconditions:** The audio is saved in the gallery section.

**Alternative course of events:**

**The actor presses the cancel button**

|  |  |  |
| --- | --- | --- |
| 7. The actor presses the cancel button. |  |  |
|  | 8. The system returns to the home screen. |  |

Table 8: Create Recording Alternate Action Use Case

**Postconditions:** The system does not save the recorded audio.











### **3.1.4 Utilize the search function**

**Summary:** The search function is a feature that allows users to find specific information within the application. This function enables users to enter keywords and the system will then display results that match the search criteria.

**UML Diagram:**

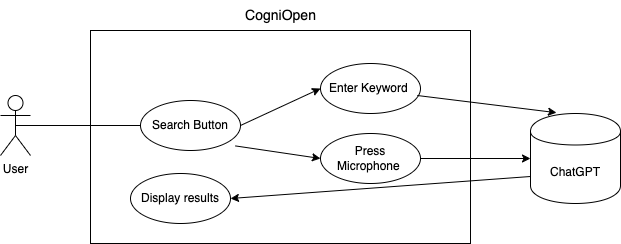


Figure 5: Search Function UML Diagram

**Preconditions:** The user has saved photos, videos, and audio files to the Gallery.

**Triggers:** The user presses the “Search” button.

**Basic course of events (main scenario):**

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Screen** |
| 1. The actor selects the “Search” button from the home screen. |  |  |
|  | **2.** The system opens the “Search” screen. |  |
| **3.** The actor enters text in the text field. |  |  |
|  | **4.** A list of search results that corresponds to the entered text is presented on the screen. |  |

Table 9: Search Function Use Case

**Postconditions:** None

### **3.1.5 Accessing the recent questions/request**

**Summary:** The recent questions or requests feature displays a list of the user's most recent inquiries or requests, providing easy and efficient access to this information.

**UML Diagram:**

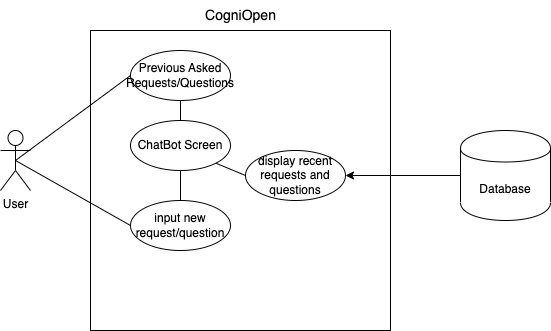


Figure 6: Recent Questions UML Diagram

**Preconditions:** The user has used the Virtual Assistant.

**Triggers:** The user presses the “Previous Asked Requests/Questions” button.

**Basic course of events (main scenario):**

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Screen** |
| **1.** The actor selects the “Previous Asked Requests/Questions” button from the home screen. |  |  |
|  | **2.** The system opens the “Virtual Assistant” screen showing recent requests and questions. |  |

Table 10: Recent Questions Use Case

**Postconditions:** None

### **3.1.6 View the menu**

**Summary:** The menu provides users with access to all available features. Upon opening the menu, users can browse through and select the options or features they desire.

**UML Diagram:** N/A

**Preconditions:** N/A

**Triggers:** N/A

**Basic course of events (main scenario):**

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Screen** |
| 1. The actor selects the “Menu” button from the home screen. |  |  |
|  | **2.** The system opens a drop-down window containing the application menu. |  |

Table 11: View Menu Use Case

**Postconditions:** None

### **3.1.7 Access the user login credential**

**Summary:** The actor opens the application to log into their account

**UML Diagram:**

A diagram of a login and password

Description automatically generated

Figure 7: Accessing the Application UML Diagram

**Preconditions:** The actor installed the application on their device.

**Triggers:** The actor opens the application from their device.

**Basic course of events (main scenario):**

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Screen** |
| 1. The actor opens the application on their mobile device. |  |  |
|  | 2. The System displays the login screen. |  |
| 3. The actor enters their email and password and clicks “Login”. |  |  |
|  | 1. The system verifies the user’s credentials and logs the user into their account. |  |

Table 12: User Login Use Case

### **3.1.8 User registration application**

**Summary:** Allows users to create an account and register for the application. The registration will include input fields for name, email, and password.

**UML Diagram:**

A diagram of a computer

Description automatically generated

Figure 8: User Registration UML Diagram

**Preconditions:** The actor installed the application on their device.

**Triggers:** The actor opens the application from their device.

**Basic course of events (main scenario):**

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Screen** |
| 1. The actor opens the application on their mobile device. |  |  |
|  | 2. The System displays the login screen. |  |
| 3. The actor clicks “Create Account”. |  |  |
|  | 4. The system opens the user registration screen. |  |
| 5. The user enters their personal information and clicks “Create Account”. |  |  |
|  | 1. The system creates a new account for the users and displays the home screen. |  |

Table 13: User Registration Use Case

### **3.1.9 View or modify the user's registration**

**Summary:** Allows users to manage and modify user registration information. It provides a centralized platform for managing personal information, contact details, and login credentials.

**UML Diagram:**

A diagram of a chatbot

Description automatically generated

Figure 9: Update Registration UML Diagram

**Preconditions:** The actor installed the application on their device.

**Triggers:** The actor opens the application from their device

**Basic course of events (main scenario):**

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Screen** |
| **1.** The actor selects the “Menu” button from the home screen. |  |  |
|  | **2.** The system opens a drop-down window containing the application menu. |  |
| **3.** The actor selects “My Profile” |  |  |
|  | **4.** The system opens the “My Profile” screen. |  |
| **5.** The actor selects a text box to update |  |  |
|  | **6.** The system opens a keyboard. |  |
| 1. The actors enter new information and selects “Save” |  |  |
|  | 1. The system saves information and refreshes the page. |  |

Table 14: Modify User Registration Use Case

**Alternative course of events:**

**The actor presses the cancel button**

|  |  |  |
| --- | --- | --- |
| 1. 7. The actors enter new information and selects “Cancel” |  |  |
|  | 8. The system cancels the new information and refreshes the page. |  |

Table 15: Modify User Registration Alternate Action Use Case

### **3.1.10 Utilize the guided tour**

**Summary:** Provide step-by-step instructions and visual aids to help users become familiar with the application's features and functionality. The guided tour is an important part of the user experience, as it can help users feel more comfortable and confident using the application.

**UML Diagram:**

A screenshot of a computer screen

Description automatically generated

Figure 10: Guided Tour UML Diagram

**Preconditions:** The actor installed the application on their device.

**Triggers:** The actor opens the application from their device

**Basic course of events (main scenario):**

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Screen** |
| **1.** The actor selects the “Menu” button from the home screen. |  |  |
|  | **2.** The system opens a drop-down window containing the application menu. |  |
| **3.** The actor selects “Guided Tour” |  |  |
|  | **4.** The system opens the first screen of the guided tour. |  |
| **5.** The actor navigatesthrough the guided tour. |  |  |
|  | **6.** Visual aids are displayed by the system, and the screen is updated each time the actor clicks on the "next" or "previous" button. |  |

Table 16: Guided Tour Use Case

## **3.2. Supplementary Requirements**

### **3.2.1 Accessibility**

* The application should put a focus on simplicity of use for users with memory issues, ensuring easy navigation and detailed explanations at each step.
* To assist users in their interactions, features including reminders, voice requests, and visual signals should be implemented.

### **3.2.2 Compatibility**

* The app needs to run on both iOS and Android devices.
* The app will support mobile devices of all sizes by utilizing responsive design.
* Major versions of both operating systems should be supported.

### **3.2.3 Security**

* User data should be encrypted both in transit and at rest, with a focus on videos, audio files, and photos.
* Every third-party integration, including ChatGPT, should make use of secure APIs that are properly authenticated.

### **3.2.4 Legal**

* The application must adhere to GDPR regulations, protecting user privacy and the right to be forgotten.
* All software licenses utilized during development must be documented.

# 4. Non-Functional Requirements

The notion of a complete baseline of capabilities includes non-functional needs, which cover system elements that go beyond the application’s core operational capability. These requirements go into depth about the system's performance, security, stability, and scalability, each of which is crucial in determining the overall user experience and assuring the system's usefulness in various situations.

**Performance:** Non-functional performance criteria describe how well the system reacts to user interactions and job requests. This covers how quickly an application responds, how long it takes to load, and how many concurrent users it can support. Performance criteria ensure that users can interact with the system quickly and smoothly, and even under heavy demand, the application's performance stays constant.

**Security:** Non-functional requirements that are security-focused strongly emphasize protecting sensitive user data and preventing unwanted access. These specifications provide rigorous authentication techniques to confirm user identities, data encryption for transport and storage, and protocols for controlling access to various application components. The system builds trust and confidence for users, caregivers, and administrators by solving security issues.

**Reliability:** Non-functional criteria for dependability depend on how regularly and dependably the system performs over time. This comprises safeguards for high uptime and availability, data integrity safeguards against information loss or corruption, and techniques for system failure recovery. Users should be able to rely on the system to be accessible when needed and to keep their data correct and complete, thanks to reliability standards.

**Scalability:** Non-functional requirements motivated by scalability consider the system's ability to handle increases in user numbers and data load. Considerations for scaling include horizontal scaling (distributing demand across numerous servers) and vertical scaling (raising resources on a single server). The system must be scaled to accommodate rising demand without sacrificing usability or performance.

Non-functional requirements, in essence, go beyond the core functions of the system to build a comprehensive framework that prioritizes performance optimization, user data protection, reliable operation, and the capacity to change to meet changing user needs. The system achieves its primary goals and guarantees a superior and dependable user experience in various circumstances by addressing these characteristics.

## **4.1 Performance:**

* **NF1 - Response Time:** The application should respond to user interactions within 2 seconds for tasks such as navigation, accessing memory games, and setting reminders.
* **NF2 - Loading Time:** The app's initial loading time should not exceed 5 seconds to ensure a seamless user experience.
* **NF3 - Scalability:** The app should handle a minimum of 10,000 concurrent users without significant degradation in performance.
* **NF4 - Memory Consumption:** The application should be optimized to minimize memory consumption, ensuring it runs smoothly on devices with varying specifications.
* **NF5 - Network Efficiency:** The app should minimize data usage to accommodate users with limited internet connectivity while ensuring a functional experience.

## **4.2 Security:**

* **NF6 - Data Encryption:** User data, including personal and medical information, should be encrypted both during transmission and storage to prevent unauthorized access.
* **NF7 - Authentication:** Robust user authentication mechanisms, including biometric options, should be implemented to ensure only authorized users can access the app.
* **NF8 - Authorization Levels:** Different levels of authorization should be established to manage access to sensitive features and data, such as caregiver accounts.
* **NF9 - Privacy Compliance:** The app should adhere to relevant data protection regulations, such as GDPR or HIPAA, depending on the jurisdiction and data processed.
* **NF10 - Secure Development:** Adhere to secure coding practices to minimize vulnerabilities and potential entry points for cyberattacks.

## **4.3 Reliability:**

* **NF10 - Uptime and Availability:** The application should have a minimum uptime of 99% to ensure consistent access for users, caregivers, and healthcare professionals.
* **NF11 - Data Integrity:** Implement mechanisms to ensure the integrity of user data, minimizing the risk of data corruption or loss.
* **NF12 - Backup and Recovery:** Regular automated backups should be performed to facilitate data recovery in case of system failures or data breaches.
* **NF13 - Error Handling:** The app should provide clear and user-friendly error messages to guide users in cases of unexpected behavior or errors.

## **4.4 Scalability:**

* **NF14 - User Growth:** The application should be designed to accommodate increasing numbers of users as the user base grows, without significant performance degradation.
* **NF15 - Server Scalability:** The backend infrastructure should be designed to scale horizontally, adding additional resources as demand increases.
* **NF16 - Database Scalability:** The database should be capable of handling increased data load and user interactions without compromising performance.