

Test Plan

(CogniOpen - Nurturing Memory Wellness for Cognitive Impairment)

Submitted to:

University of Maryland Global Campus in partial fulfillment of the course Software Engineering Capstone (SWEN 670)

Submitted By: Team B

Fall Semester 2023

Version 1.0

September 23, 2023

Department of Software Engineering

University of Maryland Global Campus (UMGC)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Issue Date | Changes | Change Made By |
| 1.0 | 09/23/2023 | Initial Version | Team B |
| 1.1 | 10/28/2023 | Add testing phases, standardize test cases, and add testing anlaysis | Zachary Cappella  Malachi Jamison |
| 1.2 | 11/5/2023 | Update test cases | Malachi Jamison |
|  |  |  |  |
|  |  |  |  |

[Test Plan 3](#_Toc37846072)

[1. Introduction 4](#_Toc1311437330)

[1.1 Purpose 4](#_Toc1656212670)

[1.2 Scope 4](#_Toc1636230710)

[1.2.1 In Scope Work 4](#_Toc1340463693)

[1.2.2 Out of Scope Work 5](#_Toc977383969)

[1.3 Testing Strategy 5](#_Toc1359733417)

[1.4 Project Documents 6](#_Toc144824547)

[Definitions, Acronyms, and Abbreviations 6](#_Toc1203824345)

[1.6 References 7](#_Toc1063293025)

[2. System Overview 7](#_Toc1367665957)

[2.1 Test Environment 7](#_Toc660742301)

[2.2 Expected Usage 8](#_Toc1838100546)

[3. Manual Test Case Overview 8](#_Toc1037055538)

[3.1 Functional Tests 9](#_Toc177999343)

[3.1.1 User Login Test Cases 9](#_Toc726309143)

[3.1.1.1 User Can Login 9](#_Toc332572638)

[3.1.1.2 User Cannot Login 9](#_Toc856324125)

[3.1.1.3 User Can Request to Create an Account 10](#_Toc1380381135)

[3.1.2 Home Interface Test Cases 11](#_Toc1597309421)

[3.1.2.1 Home Screen Buttons 11](#_Toc2030803386)

[3.1.3 Virtual Assistant Interface Test Cases 13](#_Toc1434820327)

[3.1.3.1 Verify Virtual Assistant Interaction 13](#_Toc1517999098)

[3.1.3.2 Verify Virtual Assistant Receives Transcripts 15](#_Toc675519036)

[3.1.4 Record Audio Test Cases 16](#_Toc1239122817)

[3.1.4.1 Activate/Deactivate Record Audio 17](#_Toc1686865473)

[3.1.4.2 Play/Stop Preview Recorded Audio 18](#_Toc24448695)

[3.1.4.3 Start New Recording from Audio Preview Screen 19](#_Toc2034561525)

[3.1.4.4 Delete Recorded Audio from Audio Preview Screen 21](#_Toc1446944734)

[3.1.5 Record Video Test Cases – Team A 22](#_Toc1879647384)

[3.1.6 Photo Gallery Test Cases – Team A 22](#_Toc1433860113)

[3.1.7 Video Gallery Test Cases – Team A 22](#_Toc891086861)

[3.1.8 Audio Gallery Test Cases – Team A 22](#_Toc1625228429)

[3.1.9 Onboarding Voice Assistance 22](#_Toc652834917)

[3.1.9.1 Onboarding Voice Assistance 22](#_Toc303559368)

[3.1.10 Logout Application 23](#_Toc1438547597)

[3.1.10.1 Logout Application 23](#_Toc514443360)

[3.1.11 Profile Interface Test Cases 25](#_Toc1126088901)

[3.1.11.1 User can View Profile Information 25](#_Toc293138651)

[3.1.11.2 User can Edit Profile Information 26](#_Toc1586033068)

[3.1.12 Guided Tour Test Cases 27](#_Toc1758562666)

[3.1.12.1 View Guided Tour 27](#_Toc33651214)

[3.1.12.2 Cancel Guided Tour 28](#_Toc425723165)

[3.2 Nonfunctional Tests 29](#_Toc239001780)

[3.2.1.1 NF-Sec-1 - Data Encryption 31](#_Toc1114082393)

[3.2.1.2 NF-Sec-2 – Data Transfer 31](#_Toc137065168)

[3.2.1.3 NF-Sec-3 – Authentication 32](#_Toc566622917)

[3.2.1.4 NF-Sec-4 – Authorization Levels 32](#_Toc1203194354)

[3.2.1.6 NF-Sec-5 – Privacy Compliance 33](#_Toc393705666)

[3.2.1.7 NF-Sec-6 – Secure Development 33](#_Toc390385973)

[3.2.2 Scalability 34](#_Toc964324411)

[3.2.2.1 NF-Sca-1 – User Growth 34](#_Toc1565803801)

[3.2.2.2 NF-Sca-2 – Server Scalability 34](#_Toc893421593)

[4 Testing Phases 35](#_Toc1503343211)

[4.1 Unit Testing 35](#_Toc986030509)

[4.2 Functional Testing 36](#_Toc709171345)

[4.3 Integration Testing 36](#_Toc946784240)

[4.4 Regression Testing 36](#_Toc1820725281)

[4.5 User Acceptance Testing 36](#_Toc423094285)

# Test Plan

# 1. Introduction

The CogniOpen Application is an innovative mobile application designed to assist individuals who suffer from cognitive impairment and related conditions. The application aims to provide support and help to such individuals using modern technology, artificial intelligence (AI) services, and user-friendly designs. To ensure the effectiveness and reliability of the application, it is crucial to have a comprehensive and detailed software test plan that outlines the testing strategy, approach, and scope.

## 1.1 Purpose

The purpose of this test plan is to outline the scope and approach of all testing activities for the CogniOpen Application. The plan identifies the items to be tested and details the tests to be performed on each. The test plan serves as a reference for various stakeholders involved in the testing process, including developers, project managers, testers, and end-users. The plan provides guidance on the testing activities required to ensure that the CogniOpen Application meets its functional, non-functional, and user-interface requirements. The test plan gives clear direction on the testing approach, including the testing environment, testing tools, test data, and test deliverables.

## 1.2 Scope

This project is completed in conjunction with another development team. For the purposes of this project the following sections will define portions of the application that are in scope and out of scope for Team B.

### 1.2.1 In Scope Work

This software test plan presents a detailed outline of the testing strategy, approach, and scope for Team B's portion of the CogniOpen Application. The testing activities will be focused on the functional, non-functional, and UI testing of the features listed below, and the testing approach for manual test cases will ensure comprehensive coverage. The manual test case deliverables will include the test case name, description, requirements, prerequisites, test data, test steps, expected results, test environment, pass/fail criteria, and assumptions. In addition, the test plan will also outline test phases, which will include unit testing, functional testing, integration testing, regression testing, and user-acceptance testing. The testing will be conducted on various mobile devices with different screen sizes, resolutions, and operating systems, and it will be performed in a controlled environment to ensure the accuracy and reliability of the results.

The following features are part of the scope of work for Team B and will be covered by the test cases in this document:

* Registering and logging in to the application
* Accessing features through the Home interface
* Interacting with the Virtual Assistant
* Searching for and viewing previously asked questions
* Recording audio
* Using the hamburger menu
* Managing the user profile
* Viewing the guided tour

### 1.2.2 Out of Scope Work

Any features not listed in the previous section are not part of the scope of work for Team B and are not addressed by the test cases in this document. Any requirements relating to the photo and video features are out of scope for Team B.

## 1.3 Testing Strategy

Testing for the CogniOpen application will primarily assess the function of the system and is therefore composed primarily of functional test cases. Development of test cases has been divided by User Interface (UI) screens. Each screen was analyzed to exhaustively determine possible user interactions and each interaction was then given a test case to define the expected system behavior in response.

Any non-functional requirements not otherwise covered by functional tests are covered in the non-functional tests section. These are not self-contained tests that can be performed on the system. Instead, they should be referenced whenever the system performs relevant activity to ensure the system meets the related non-functional requirements.

Additional test cases may be defined as the system is constructed. No unit tests have been created at this time, but they may be included in the future if needed.

**1.3.1 Test Prioritization**

This section outlines how the testing activities will be prioritized based on the importance of the features and their impact on the overall functionality of the CogniOpen Application. By prioritizing the most important features, the testing team can ensure that they are thoroughly tested. The prioritization of the features will be based on the following criteria:

* Criticality of the feature: This criterion will prioritize the features based on their importance to the application's overall functionality. Features that are critical to the application's primary purpose will be given top priority.
* Frequency of use: This criterion will prioritize the features based on how frequently the end-users use them. Features that are used more frequently will be given a higher priority.
* Impact of failure: This criterion will prioritize the features based on the impact of their failure on the application. Features that significantly impact the application's usability or functionality when they fail will be given a higher priority.

Based on these criteria, the testing team will prioritize the testing activities and allocate resources accordingly. The prioritizing testing section will ensure that the most important and high-stakes test conditions are tested first during testing, thereby ensuring that the application meets its functional, non-functional, and user-interface requirements.

**1.3.2 Test Monitoring and Control**

The Test Monitoring and Control section is a critical component of the test plan that outlines the criteria for test entry, exit, suspension, and resumption. This section helps to ensure that the testing process is well-monitored and that the testing team can effectively control the testing activities.

To effectively monitor and control the testing process, the following is a detailed description of the criteria for test entry, exit, suspension, and resumption:

* Test entry criteria: The conditions that must be met before testing can begin. This could include the completion of previous testing phases, the availability of necessary test environments, and the availability of test data.
* Test exit criteria: The conditions that must be met before testing can be completed. This could include meeting the pass/fail criteria for each test case, achieving the required level of test coverage, and resolving all open defects.
* Test suspension criteria: The conditions that must be met before testing can be suspended temporarily. This could include the need to address critical defects, the unavailability of necessary resources, or changes to project timelines.
* Test resumption criteria: The conditions that must be met before testing can be resumed after being suspended. This could include the resolution of critical defects, the availability of necessary resources, and changes to project timelines.

By defining these criteria, the testing team can ensure that the testing process is well-controlled, and that proper monitoring of the testing activities takes place.

## 1.4 Project Documents

This document is one part of the deliverables for the CogniOpen project. The other documents are listed in the table below with the most recent version of each at the time of editing this Test Plan.

|  |  |  |
| --- | --- | --- |
| Document | Version | Date |
| Project Plan (PPL) | 1.2 | 10/28/2023 |
| Software Requirements Specification (SRS) | 1.2 | 10/28/2023 |
| Technical Design Document (TDD) | 1.1 | 10/28/2023 |
| Test Plan (TP) | 1.1 | 10/28/2023 |
| Programmer Guide (PG) | 1.0 | 10/28/2023 |
| Deployment and Operations Guide (DOG) | 1.0 | 10/28/2023 |
| Software Test Report (STR) | - | - |
| User Guide (UG) | - | - |
| Traceability Matrix (TM) | - | - |

Table 1 – Project Documents

|  |  |
| --- | --- |
| Definitions, Acronyms, and AbbreviationsTerm | Definition |
| ADO | Azure DevOps |
| App | A program that is included on the User’s mobile device |
| HIPPA | Health Insurance Portability and Accountability Act |
| HTTPS | Hypertext Transfer Protocol Secure |
| iOS | iPhone Operating System |
| Mobile Device | A smart phone, tablet, or some other portable computer with either the iOS or Android operating system |
| TDD | Technical Design Document |
| UAT | User acceptance testing |
| UI | User Interface |

Table 2 – Definitions Acronyms and Abbreviations

## 1.6 References

1. OpenAI. (2023, September 21). Mobile App TDD Overview [Online Conversation]. OpenAI. <https://chat.openai.com/share/f894c79d-a3d5-4f70-a414-ee4a872a2b0d>

*2. University of Maryland Global Campus (UMGC) SWEN 670 Capstone Project Management System - CaPPMS*. (2023). Azurewebsites.net; CaPPMS. <https://umgc-cappms.azurewebsites.net/previousprojects>

3. Spillner, A., Linz, T. (2018). Software Testing Foundations, 5th Edition. Rocky Nook.

‌

# 2. System Overview

The system overview section provides a high-level understanding of the CogniOpen application, including details about the test environment and the expected usage.

## 2.1 Test Environment

The system is expected to run on Android and IOS operating systems (OS). Further details of the intended testing environment will be added to this section as they are determined.

The testing environment plays a crucial role in assessing the application's compatibility and performance. The CogniOpen application is designed to be versatile and run seamlessly on both Android and iOS OS. The following details will be added to this section as they are determined:

* Operating Systems: Android and iOS are the primary target platforms for the CogniOpen application. Testing will encompass a range of versions and devices to ensure broad compatibility.
* Hardware Specifications: The specific hardware requirements and configurations will be documented here, detailing any variations in device capabilities and form factors as development continues.
* Network Conditions: Different network conditions, including varying levels of connectivity, will be considered during testing to assess the application's performance under diverse scenarios.
* Testing Tools: Any specialized testing tools, simulators, emulators, or physical devices used in the testing process will be listed in this section as development continues.
* Additional Environmental Factors: Any other environmental factors relevant to testing, such as screen resolutions, input methods, or accessibility settings, will be documented as development continues.

## 2.2 Expected Usage

The expected usage section provides a comprehensive overview of how the CogniOpen application will be utilized, including its core functions and the components that drive its functionality.

The CogniOpen application is designed to serve as a comprehensive memory wellness solution for individuals dealing with cognitive impairment. Its primary functions and key components include:

* User Login: Users can securely log in to the application using their registered credentials, ensuring data privacy and security.
* User Application Registration: New users can create accounts to access the application's features.
* Home Interface: The central hub where users can access various features and navigate through the application's features.
* Virtual Assistant Interface: A conversational interface designed to assist users with their memory wellness activities, answer questions, and provide support.
* Record Audio: Users can record audio content, such as voice notes or reminders, to aid in memory retention.
* Record Video: Video recording functionality for capturing important moments or instructions.
* Photo Gallery: A gallery feature that allows users to organize and browse photos, enhancing memory recall through visual cues.
* Video Gallery: A gallery feature for storing and playing back videos, promoting engagement, and reminiscence.
* Audio Gallery: A gallery for organizing and replaying audio recordings, assisting users in remembering spoken information.
* Search Interface: A search function enabling users to quickly locate and retrieve specific content.
* Previously Asked Questions: A repository of commonly asked questions and answers to support users in memory-related inquiries.
* Menu Interface: Navigation and settings options to customize the user experience.
* Profile Interface: User profile management for updating personal information and preferences.
* Guided Tour: A feature that guides users through the application's functionalities and provides an interactive tutorial.

Each of these components contributes to the application's overall goal of nurturing memory wellness for individuals experiencing cognitive impairment. Throughout this test plan, we will comprehensively assess each of these functions to ensure they meet the intended objectives and operate reliably under various scenarios and environmental conditions.

# 3. Manual Test Case Overview

## 3.1 Functional Tests

### 3.1.1 User Login Test Cases

#### 3.1.1.1 User Can Login

|  |  |
| --- | --- |
| **Description** | A user can login to the CogniOpen application with valid user credentials. |
| **Requirements** | The application shall authenticate and redirect a user to the home screen when they enter valid login information and tap the login button. |
| **Prerequisites** | * The user has previously registered with the application * The application has been downloaded on the user’s phone |
| **Test Data** | * User’s biometric authentication information |
| **Test Steps** | 1. Launch the CogniOpen application from the device 2. User taps the “Login in Here” button. 3. Users enter their biometric authentication information. |
| **Test Environment** | * Android Emulator: Pixel 7 Pro API |
| **Pass/Fail Criteria** | Pass: The user is logged in and is redirected to the Home Interface  Fail: The user is not logged in |
| **Assumptions** | * The user has previously registered with the application * The user is connected to the internet * The application is active and able to receive requests * The application is able to communicate with the backend database services |

Table 2: Valid Login Test Case

#### 3.1.1.2 User Cannot Login

**Test Case Name:** Deny Login with Invalid Credentials

|  |  |
| --- | --- |
| **Description** | A user cannot login to the CogniOpen application with invalid user credentials. |
| **Requirements** | The application shall deny users access to the application if they enter invalid user credentials and display a generic login error message. |
| **Prerequisites** | * The user has previously registered with the application * The application has been downloaded on the user’s phone |
| **Test Data** | * N/A |
| **Test Steps** | 1. Launch the CogniOpen application from the device 2. User taps the “Login in Here” button 3. Users enter their biometric authentication information. |
| **Test Environment** | * Android Emulator: Pixel 7 Pro API |
| **Pass/Fail Criteria** | Pass: The application does not redirect the user to Home Interface and displays an error message to the user.  Fail: The user can bypass the invalid login and is able to access CogniOpen resources. |
| **Assumptions** | * The user did not register with the application * The user is connected to the internet * The application is active and able to receive requests * The application can communicate with the backend database services |

Table 3: Deny Login Test Case

#### 3.1.1.3 User Can Request to Create an Account

**Test Case Name:** Request to Create an Account

|  |  |
| --- | --- |
| **Description** | A user can request to create an account. |
| **Requirements** | The application shall allow users to create a new account. |
| **Prerequisites** | * The application has been downloaded on the user’s phone |
| **Test Data** | * User’s biometric authentication information |
| **Test Steps** | 1. Launch the CogniOpen application from the device 2. User taps the “Create Account” button 3. Users enters first name, last name and email 4. Users taps the “Create Account” button. |
| **Test Environment** | * Android Emulator: Pixel 7 Pro API |
| **Pass/Fail Criteria** | Pass: The user is redirected to the screen where they can create a new account  Fail: The user is not redirected to the screen to create a new account |
| **Assumptions** | * The user is connected to the internet * The application is active and able to receive requests * The application can communicate with the backend database services |

Table 4: Create Account Test Case

### 3.1.2 Home Interface Test Cases

#### 3.1.2.1 Home Screen Buttons

**Test Case Name:** Home Screen Application Buttons Features Functionality

|  |  |
| --- | --- |
| **Description** | This test case is designed to verify that the home screen application with six button features functions correctly and meets all specified requirements. The application contains six buttons that lead to different features and functionalities. The main objective of this test case is to ensure that all the buttons and features work as expected and that the application is user-friendly and easy to navigate. The six features are as follows Team B responsibility:   1. Virtual Assistant Button 2. Record Audio Button 3. Guided Tour Button 4. Home Button 5. Menu Button 6. Back Button |
| **Requirements** | * The home screen will allow users to access all the application’s main features. The home screen application should display buttons with their respective icons and labels to users upon logging in. * Each button should lead to the expected feature or functionality * The buttons should be responsive when clicked * All UI elements, such as buttons and text boxes, should be responsive and functional – * All required data should be displayed accurately and consistently throughout the application * User input should be accepted and processed correctly by the application * The application should be usable in all supported device orientations and screen sizes |
| **Prerequisites** | * The CogniOpen application must be successfully installed and running on the test device. * The test device must have a compatible operating system version for the application. * The user must have a valid account and be logged into the CogniOpen application. |
| **Test Data** | * User’s biometric authentication information |
| **Test Steps** | 1. Verify that the application is installed and configured correctly on the device. 2. Launch the CogniOpen application from the device 3. User taps the “Login in Here” button. 4. User enters their biometric authentication information. 5. Verify that the home screen displays all six buttons with their respective icons and labels. 6. Test each button by clicking on it and verify that it leads to the expected feature or functionality. 7. Verify that the buttons are responsive and that they highlight when clicked. 8. Test that all UI elements, such as buttons and text boxes, are responsive and functional. 9. Verify that all required data is displayed accurately and consistently throughout the application. 10. Test that user input is accepted and processed correctly by the application. 11. Verify that the application can be used in all supported device orientations and screen sizes. |
| **Test Environment** | * Android Emulator: Pixel 7 Pro API |
| **Pass/Fail Criteria** | Pass:   * All buttons and features are displayed as expected. * All features should open and load without any errors or crashes. * The feature's UI elements should be responsive and functional. * User input should be accepted and processed correctly by the feature.   Fail:   * Any buttons or features that are not displayed. * Any buttons or features fail to open or load correctly. * The feature's UI elements are not responsive or functional. * Required data is missing or displayed incorrectly throughout the feature. * User input is not accepted or processed correctly by the feature. |
| **Assumptions** | * The user is connected to the internet * The application is active and able to receive request * The application can communicate with the backend database services |

Table 5: Home Screen Test Case

### 3.1.3 Virtual Assistant Interface Test Cases

#### 3.1.3.1 Verify Virtual Assistant Interaction

**Test Case Name:** Verify Virtual Assistant Interaction

|  |  |
| --- | --- |
| **Description** | Verify that the Virtual Assistant screen in the application functions as expected, allowing users to interact with the AI assistant through text input, voice input, and receiving responses. |
| **Requirements** | * The Virtual Assistant screen in the CogniOpen application must allow users to interact with the AI assistant through text input, voice input, and receiving responses. * The Virtual Assistant must accurately recognize and respond to user inputs, including text-based and voice-based queries. * The chat history of the conversation between the user and the Virtual Assistant must be displayed and maintained until the user closes the application or navigates to a different screen. |
| **Prerequisites** | * The CogniOpen application must be properly installed and operational on the test device. * The Virtual Assistant screen must be accessible within the application. * Internet connectivity must be available for Virtual Assistant functionality. |
| **Test Data** | * User’s biometric authentication information * Test questions or requests for interaction with the Virtual Assistant. * Test voice input (if applicable). * Samples:  1. "What's the weather like today?" 2. "Tell me a joke." 3. "Give me the latest news updates." 4. "Navigate to the settings menu." 5. "Search for nearby restaurants." 6. Test voice input (if applicable):  * Clear and distinct spoken queries, corresponding to the text input test questions. |
| **Test Steps** | 1. Launch the CogniOpen application from the device. 2. User taps the “Login in Here” button. 3. Users enter their biometric authentication information. 4. Navigate to the Virtual Assistant screen 5. Observe the Virtual Assistant screen and verify the presence of the following elements: 6. A text box for user input (keyboard input). 7. An option to enable microphone input (voice input) as seen in Figure 5 of the TDD. 8. Chat history displaying previous interactions. 9. Virtual Assistant responses or options presented to the user.   **Interaction with the Virtual Assistant:**   1. Enter a text-based question or request in the text box and tap the "Send" button. 2. Enable microphone input (if available) and speak a question or request. 3. Tap on any buttons or links presented by the Virtual Assistant within the conversation. 4. Continue the conversation by asking multiple questions or making requests. |
| **Test Environment** | * + Android Emulator: Pixel 7 Pro API |
| **Pass/Fail Criteria** | Pass:   * + The Virtual Assistant responds with a relevant text-based reply or presents options if applicable.   + The Virtual Assistant accurately recognizes and responds to voice input, providing text-based replies or options.   + The Virtual Assistant maintains context and responds appropriately to each user input.   Fail:   * If any step fails to meet the expected outcome, such as inaccurate responses, unresponsive elements, or failure to recognize voice input. * If the Virtual Assistant fails to recognize voice input accurately or provides incorrect responses. * If the Virtual Assistant fails to provide a coherent or relevant response to complex queries. |
| **Assumptions** | * The user is connected to the internet * The application is active and able to receive requests * The application can communicate with the backend database services * Internet connectivity is available for Virtual Assistant functionality. * The Virtual Assistant's natural language processing (NLP) capabilities are properly configured and connected to the internet for processing user queries. |

Table 6: Virtual Assistant Test Case

#### 3.1.3.2 Verify Virtual Assistant Receives Transcripts

**Test Case Name:** Verify Virtual Assistant Receives Transcripts

|  |  |
| --- | --- |
| **Description** | Verify that the Virtual Assistant screen in the application receives transcripts from the Gallery when the user navigates from there, allowing users to ask questions about the transcript. |
| **Requirements** | * The Virtual Assistant screen in the CogniOpen application must allow users to interact with the AI assistant through text input, voice input, and receiving responses. * The Virtual Assistant must accurately recognize and respond to user inputs, including text-based and voice-based queries. * The Virtual Assistant must be able to receive a conversation transcript and be able to interpret it and respond to questions about it. * The chat history of the conversation between the user and the Virtual Assistant must be displayed and maintained until the user closes the application or navigates to a different screen. |
| **Prerequisites** | * The CogniOpen application must be properly installed and operational on the test device. * The Virtual Assistant screen must be accessible within the application. * Internet connectivity must be available for Virtual Assistant functionality. |
| **Test Data** | * User’s biometric authentication information * Test questions or requests for interaction with the Virtual Assistant. * Test voice input (if applicable). * Samples:  1. "What's the weather like today?" 2. "Tell me a joke." 3. "Give me the latest news updates." 4. "Navigate to the settings menu." 5. "Search for nearby restaurants." 6. Test voice input (if applicable):  * Clear and distinct spoken queries, corresponding to the text input test questions. |
| **Test Steps** | 1. Launch the CogniOpen application from the device. 2. User taps the “Login in Here” button. 3. Users enter their biometric authentication information. 4. Navigate to the Gallery screen 5. Tap on a conversation. 6. Tap on the “Ask Cora” button. 7. Verify the app redirects to the Virtual Assistant screen.   **Interaction with the Virtual Assistant:**   1. Enter a text-based question or request, related to the transcript selected in step 5, in the text box. Then tap the "Send" button. 2. Continue the conversation by asking multiple questions or making requests. |
| **Test Environment** | * + Android Emulator: Pixel 7 Pro API |
| **Pass/Fail Criteria** | Pass:   * + The Virtual Assistant responds with a relevant text-based reply that demonstrates understanding of the transcript.   + `The Virtual Assistant maintains context of both the transcript and previous messages and responds appropriately to each user input.   Fail:   * + The Virtual Assistant does not respond with a relevant text-based reply that demonstrates understanding of the transcript.   + `The Virtual Assistant can’t maintain context of both the transcript and previous messages and responds appropriately to each user input. |
| **Assumptions** | * The user is connected to the internet * The application is active and able to receive requests * The application can communicate with the backend database services * There is at least one conversation stored in the Gallery. * Internet connectivity is available for Virtual Assistant functionality. * The Virtual Assistant's natural language processing (NLP) capabilities are properly configured and connected to the internet for processing user queries. |

Table 7: Virtual Assistant Transcriptions Test Case

### 3.1.4 Record Audio Test Cases

#### 3.1.4.1 Activate/Deactivate Record Audio

**Test Case Name:** Activate/Deactivate Record Audio

|  |  |
| --- | --- |
| **Description** | Verify that the “Start Audio Recording” circular button on the Record Audio screen initiates the recording process when tapped. Verify that the circular button is replaced by a square-shaped button during recording, indicating that recording is in progress. Verify that tapping the "Stop" button ends the recording and transitions to the preview screen. |
| **Requirements** | * The CogniOpen application must have a “Start Audio Recording” circular button on the Record Audio screen that initiates the recording process when tapped. * During recording on the Record Audio screen, the circular button must be replaced by a square-shaped button to indicate that recording is in progress. The timer must also start counting. * The Record Audio screen must feature a " Stop Audio Recording" " button that ends the recording when tapped, transitioning to the preview screen. |
| **Prerequisites** | * The CogniOpen application must be successfully installed and running on the test device. * The test device must have a compatible operating system version for the application. * The user must have a valid account and be logged into the CogniOpen application. * The user must navigate to the Record Audio screen within the application. |
| **Test Data** | * User’s biometric authentication information |
| **Test Steps** | 1. Launch the CogniOpen application. 2. User taps the “Login in Here” button. 3. Users enter their biometric authentication information. 4. Navigate to the Record Audio screen. 5. Tap the “Start Audio Recording” circular button in the center of the screen. 6. Check if the timer has started and accurately count during recording. 7. While recording is in progress (with the square-shaped button displayed), tap the "Stop" button. |
| **Test Environment** | * Android Emulator: Pixel 7 Pro API |
| **Pass/Fail Criteria** | Pass:   * + Tapping the “Start Audio Recording” circular button starts the recording.   + The circular button is replaced by a square-shaped button, indicating that recording is in progress. The timer starts counting.   + Tapping the "Stop Audio Recording" button ends the recording and transitions to the audio preview screen   Fail:   * + Tapping the “Start Audio Recording” circular button does not start the recording.   + The circular button is not replaced by a square-shaped button, indicating that recording is in progress. The timer starts counting.   + Tapping the "Stop Audio Recording" button does not ends the recording and transitions to the audio preview screen |
| **Assumptions** | * The CogniOpen application is properly installed and functioning on the test device. * The device's operating system is compatible with the application. * The user has successfully logged into the application and navigated to the Record Audio screen. |

Table 8: Activate/Deactivate Audio Recording Test Case

#### 3.1.4.2 Play/Stop Preview Recorded Audio

**Test Case Name:** Play/Stop Preview Recorded Audio

|  |  |
| --- | --- |
| **Description** | Verify that the on the “Play Preview” button starts playing the recorded audio when tapped. Verify that the on the “Stop Preview” button stop playing the recorded audio when tapped. |
| **Requirements** | * The CogniOpen application must have a “Play Preview” button on the Record Audio screen that initiates the recorded audio when tapped. * The Record Audio screen must feature a " Stop Preview” button that stops the recorded audio when tapped. |
| **Prerequisites** | * The CogniOpen application must be successfully installed and running on the test device. * The test device must have a compatible operating system version for the application. * The user must have a valid account and be logged into the CogniOpen application. * The user must navigate to the Record Audio screen within the application. * The stop and play button must be available and functional during the recording process. |
| **Test Data** | * User’s biometric authentication information |
| **Test Steps** | 1. Launch the CogniOpen application. 2. User taps the “Login in Here” button. 3. Users enter their biometric authentication information. 4. Navigate to the Record Audio screen. 5. Tap the “Start Audio Recording” button. 6. Say a few words to be recorded. 7. Tap the “Stop Audio Recording" button. 8. Tap the “Play Preview” button” 9. Ensure the accuracy of the audio by listening to it and verifying that it correctly captures what you were saying. 10. Tap the “Stop Preview” button. |
| **Test Environment** | Android Emulator: Pixel 7 Pro API |
| **Pass/Fail Criteria** | Pass:   * + Tapping the “Start Preview” button plays the recording.   + Tapping the "Stop Preview" button stops the recording.   Fail:   * + Tapping the “Start Preview” button does not play the recording.   + Tapping the "Stop Preview" button does not stop the recording. |
| **Assumptions** | * The CogniOpen application is properly installed and functioning on the test device. * The device's operating system is compatible with the application. * The user has successfully logged into the application and navigated to the Record Audio screen. |

Table 9: Play/Pause Audio Recording Test Case

#### 3.1.4.3 Start New Recording from Audio Preview Screen

**Test Case Name:** Start New Recording from Audio Preview Screen

|  |  |
| --- | --- |
| **Description** | To start a new recording, simply press the "New Recording" button, this will begin capturing new audio. |
| **Requirements** | * The CogniOpen application must have a “New Recording” button on the Record Audio screen that initiates a new recording when tapped. |
| **Prerequisites** | * The CogniOpen application must be successfully installed and running on the test device. * The test device must have a compatible operating system version for the application. * The user must have a valid account and be logged into the CogniOpen application. * The user must navigate to the Record Audio screen within the application. * The “Start New Recoding” must be available and functional after the recording process. |
| **Test Data** | * User’s biometric authentication information |
| **Test Steps** | 1. Launch the CogniOpen application. 2. User taps the “Login in Here” button. 3. Users enter their biometric authentication information. 4. Navigate to the Record Audio screen. 5. Tap the “Start Audio Recording” button. 6. Say a few words to be recorded. 7. Tap the “Stop Audio Recording" button. 8. Tap the “New Recording” button. |
| **Test Environment** | * + Android Emulator: Pixel 7 Pro API |
| **Pass/Fail Criteria** | Pass:   * + Tapping the “New Recording” button starts a recording.   Fail:   * + Tapping the “New Recording” button does not start a new a recording. |
| **Assumptions** | * The CogniOpen application is properly installed and functioning on the test device. * The device's operating system is compatible with the application. * The user has successfully logged into the application and navigated to the Record Audio Preview screen. |

Table 10: Start New Audio Recording Test Case

#### 3.1.4.4 Delete Recorded Audio from Audio Preview Screen

**Test Case Name:** Delete Recorded Audio from Audio Preview Screen

|  |  |
| --- | --- |
| **Description** | To delete a recorded audio, press the “Trash Can" icon to remove the recording from the CogniOpen application. |
| **Requirements** | * The Record Audio screen must feature a " Trash Can" icon button that deletes the recorded audio when tapped. |
| **Prerequisites** | * The CogniOpen application must be successfully installed and running on the test device. * The test device must have a compatible operating system version for the application. * The user must have a valid account and be logged into the CogniOpen application. * The user must navigate to the Record Audio screen within the application. * A recorded audio file must be available for discarding within the application. * The application must have the capability to transition from the preview screen to the Record Audio screen upon tapping the "Trash Can" icon button. |
| **Test Data** | * User’s biometric authentication information |
| **Test Steps** | 1. Launch the CogniOpen application. 2. User taps the “Login in Here” button. 3. Users enter their biometric authentication information. 4. Navigate to the Record Audio screen. 5. Tap the “Start Audio Recording” button. 6. Say a few words to be recorded. 7. Tap the “Stop Audio Recording" button. 8. Tap the “Trash Can” icon button. |
| **Test Environment** | * Android Emulator: Pixel 7 Pro API |
| **Expected Result** | Pass: Tapping the "Trash Can" icon discards the recording and returns to the Record Audio screen.  Fail: If the recording is not discarded, or if the button does not function as expected. |
| **Assumptions** | * The CogniOpen application is properly installed and functioning on the test device. * The device's operating system is compatible with the application. * The user has successfully logged into the application and navigated to the Record Audio Preview screen. |

Table 11: Delete Recorded Audio Test Case

### 3.1.5 Record Video Test Cases – Team A

These test cases will be covered by our accompanying team that is assisting in the development of the CogniOpen application.

### 3.1.6 Photo Gallery Test Cases – Team A

These test cases will be covered by our accompanying team that is assisting in the development of the CogniOpen application.

### 3.1.7 Video Gallery Test Cases – Team A

These test cases will be covered by our accompanying team that is assisting in the development of the CogniOpen application.

### 3.1.8 Audio Gallery Test Cases – Team A

These test cases will be covered by our accompanying team that is assisting in the development of the CogniOpen application.

### 3.1.9 Onboarding Voice Assistance 3.1.*9*.1 Onboarding Voice Assistance

**Test Case Name:** Onboarding Voice Assistance

|  |  |
| --- | --- |
| **Description** | During the onboarding process, the virtual assistant welcomes the user and provides an overview of the system. It then proceeds to guide the user through a series of steps to complete the necessary tasks. The virtual assistant will have a conversation with the user to help them understand the system functions. |
| **Requirements** | * The onboarding should be able to guide users through the process of interacting with the virtual assistance and create a conversation summary. * The virtual assistance onboarding should be able to adapt to the user's individual needs and preferences and provide them with a personalized onboarding experience. * The AI onboarding should be easy to use and navigate, and should not require any prior knowledge of the application |
| **Prerequisites** | * The CogniOpen application must be properly installed and operational on the test device. * The onboarding screen must be accessible when the user first registers and opens the application. |
| **Test Data** | * User’s biometric authentication information * Test questions or requests for interaction with the onboarding Virtual Assistant. * Test voice input (if applicable). * Samples:  1. "What do you like to do?" 2. "What is your favorite memory."  * Clear and distinct spoken queries, corresponding to the text input test questions. |
| **Test Steps** | 1. Launch the CogniOpen application from the device 2. User taps the “Create Account” button 3. Users enters first name, last name, and email 4. Users tap the “Create Account” button. 5. The screen is redirected to the onboarding welcome screen. 6. The user performs each action virtual assistance is asking of you, then click the “Next” button. 7. After the onboarding, the user is redirected to the home screen |
| **Test Environment** | * + Android Emulator: Pixel 7 Pro API |
| **Pass/Fail Criteria** | Pass: The user should be able to complete the onboarding process successfully and the system should redirect the user to the home screen.  Fail: The user is not able to complete the onboarding process and the system does not redirect the user to the home screen at the end. |
| **Assumptions** | * + The user must be a first-time registered user to login to the application. * The user is connected to the internet * The application is active and able to receive requests * The application can communicate with the backend database services |

Table 12: Onboarding Assistant Test Case



### 3.1.10 Logout Application

#### 3.1.10.1 Logout Application

**Test Case Name:** Logout Application

|  |  |
| --- | --- |
| **Description** | Verify that the logout functionality of the application works as expected and meets all specified requirements. The test case involves testing the logout button and verifying that the user's session is securely ended. |
| **Requirements** | * Launch the CogniOpen application from the device. * User taps the “Login in Here” button. * Users enter their biometric authentication information. * The logout button should be accessible to the user from the menu screen in the application. * Clicking on the logout button or option should immediately end the user's session and securely clear any unsaved data or input. * The user should be returned to the login screen, or the application should be exited altogether. |
| **Prerequisites** | * The CogniOpen application must be successfully installed and running on the test device. * The test device must have a compatible operating system version for the application. * The user must have a valid account and be logged into the CogniOpen application. |
| **Test Data** | * User’s biometric authentication information |
| **Test Steps** | 1. Verify that the logout button or option is available and accessible to the user. 2. Click on the logout button or option and verify that the user's session is immediately ended. 3. Verify that any unsaved data or input is securely cleared from the device or server. 4. Verify that the user is returned to the login screen or that the application is exited altogether. |
| **Test Environment** | * + Android Emulator: Pixel 7 Pro API |
| **Pass/Fail Criteria** | Pass:   * + The logout feature performs as intended, closing the user's session in a secure manner, erasing any unsaved information, and providing suitable feedback to the user.   + The logout functionality is protected by appropriate security protocols to prevent unauthorized access.   Fail:   * The logout functionality does not work as expected, the user's session is not securely ended, unsaved data is not cleared, inappropriate feedback is provided to the user, or appropriate security measures are not in place. |
| **Assumptions** | * The user has previously registered with the application * The user is connected to the internet * The application is active and able to receive requests * The application can communicate with the backend database services |

Table 13: Application Logout Test Case

### 3.1.11 Profile Interface Test Cases

#### 3.1.11.1 User can View Profile Information

**Test Case Name:** View Profile Information

|  |  |
| --- | --- |
| **Description** | A user can successfully view their profile information in the application. |
| **Requirements** | * The profile screen displays the user's current information, including first name, last name, email address, and password * The user shall be able to edit any of the displayed fields * The user shall be able to save the edited information * The user shall be able to cancel any changes made and revert to the previous information |
| **Prerequisites** | * The user is logged in to the CogniOpen application. * The profile screen is accessible from the home screen. |
| **Test Data** | * User’s biometric authentication information * First Name: John * Last Name: Doe * Email Address: testuser@sample.com |
| **Test Steps** | 1. Launch the CogniOpen application from the device. 2. User taps the “Login in Here” button. 3. Users enter their biometric authentication information. 4. Navigate to the profile screen from the home screen. 5. On the profile screen, locate and edit any of the provided fields. 6. Tap the "Cancel" button to exit the profile. |
| **Test Environment** | Android Emulator: Pixel 7 Pro API |
| **Pass/Fail Criteria** | Pass: The user's profile information is displayed correctly on the profile screen.  Fail: The user's profile information is not displayed correctly on the profile screen. |
| **Assumptions** | * The user is already registered with the application * The user has previously entered profile information during registration * The application is connected to the internet to retrieve user profile data * The application has access to a backend database to store and retrieve user profile data |

Table 14: View Profile Test Case

#### 3.1.11.2 User can Edit Profile Information

**Test Case Name:** Edit Profile Information

|  |  |
| --- | --- |
| **Description** | A user can successfully edit their profile information in the application. |
| **Requirements** | * The profile screen displays the user's current information, including first name, last name, email address, and password * The user shall be able to edit any of the displayed fields * The user shall be able to save the edited information * The user shall be able to cancel any changes made and revert to the previous information |
| **Prerequisites** | * The user is logged in to the CogniOpen application. * The profile screen is accessible from the home screen. |
| **Test Data** | * User’s biometric authentication information * First Name: John * Last Name: Doe * Email Address: testuser@sample.com |
| **Test Steps** | 1. Launch the CogniOpen application from the device. 2. User taps the “Login in Here” button. 3. Users enter their biometric authentication information. 4. Navigate to the profile screen from the home screen. 5. On the profile screen, locate and edit any of the provided fields. 6. Tap the "Save" button to save the changes made. 7. Verify that the edited information is saved by checking the displayed profile information. |
| **Test Environment** | * + Android Emulator: Pixel 7 Pro API |
| **Pass/Fail Criteria** | Pass: The edited information is successfully saved, and the displayed profile information reflects the changes.  Fail: The edited information is not saved, and the displayed profile information does not reflect the changes. |
| **Assumptions** | * The user is already registered with the application * The user has previously entered profile information during registration * The application is connected to the internet to retrieve user profile data * The application has access to a backend database to store and retrieve user profile data |

Table 15: Edit Profile Test Case

### 3.1.12 Guided Tour Test Cases

#### 3.1.12.1 View Guided Tour

**Test Case Name:** View Guided Tour

|  |  |
| --- | --- |
| **Description** | The guided tour feature is designed to assist users in navigating the system and understanding its functionalities. The guided tour is accessible from the system's interface and provides a step-by-step tutorial on how to use the system's key features. |
| **Requirements** | * The application shall display a guided tour to users when they tap the “Guided Tour” button on the CogniOpen application’s Home Screen. * When the user taps the green arrow facing right, the application shall progress the user to the next guided tour screen. * When the user taps the green arrow facing left, the application shall take the user back to the previous screen. |
| **Prerequisites** | * The user is logged in to the CogniOpen application. * The profile screen is accessible from the home screen. |
| **Test Data** | * User’s biometric authentication information |
| **Test Steps** | 1. Launch the CogniOpen application from the device. 2. User taps the “Login in Here” button. 3. Users enter their biometric authentication information. 4. Navigate to the Guided Tour screen from the home screen. 5. User taps the “Green Arrow” facing to the right 6. User taps the “Green Arrow” facing left 7. Repeat the step 7 two times 8. Repeat step 8 one time 9. Repeat steps 7 until the tour is complete. |
| **Test Environment** | * + Android Emulator: Pixel 7 Pro API |
| **Pass/Fail Criteria** | Pass:   * + The guided tour feature allows users to navigate through the tutorial easily.   + When a user taps the green arrow to the right, they are automatically redirected to the next page of the tutorial.   + When a user taps the green arrow to the left, they are redirected back to the previous screen.   Fail:   * + The guided tour feature does not allow users to navigate through the tutorial.   + When a user taps the green arrow to the right, they do not automatically redirect to the next page of the tutorial.   + When a user taps the green arrow to the left, they do not redirect back to the previous screen. |
| **Assumptions** | * The user has previously registered with the application * The user is connected to the internet * The application is active and able to receive requests * The application is able to communicate with the backend database services |

Table 16: View Guided Tour Test Case

#### 3.1.12.2 Cancel Guided Tour

**Test Case Name:** Cancel Guided Tour

|  |  |
| --- | --- |
| **Description** | A user can decide to cancel the guided tour, they can do so at any point during the tutorial by clicking on the cancel button provided on the screen. |
| **Requirements** | * The application shall display a guided tour to users when they tap the “Guided Tour” button on the CogniOpen application’s Home Screen. * When the user taps the green arrow facing right, the application shall progress the user to the next guided tour screen. * When the user taps the green arrow facing left, the application shall take the user back to the previous screen. * When the user taps the “X” button on the top right of the screen, the Guided Tour will exit and take the user back to the Home Screen. |
| **Prerequisites** | * The user is logged in to the CogniOpen application. * The Guided Tour screen is accessible from the home screen. |
| **Test Data** | * User’s biometric authentication information |
| **Test Steps** | 1. Launch the CogniOpen application from the device. 2. User taps the “Login in Here” button. 3. Users enter their biometric authentication information. 4. Navigate to the Guided Tour screen from the home screen. 5. User taps the “Green Arrow” facing to the right 6. User taps the “X” button to cancel the tour. 7. The screen redirects to the home screen. |
| **Test Environment** | * + Android Emulator: Pixel 7 Pro API |
| **Pass/Fail Criteria** | Pass: When a user taps the “X” button, the Guided Tour is closed out and they are automatically redirected to the home screen.  Fail: When a user taps the “X” button, the Guided Tour is not closed out and they do not automatically redirect to the home screen. |
| **Assumptions** | * The user has previously registered with the application * The user is connected to the internet * The application is active and able to receive requests * The application can communicate with the backend database services |
| **Actual Result** | The back arrow was clicked, and the Home Screen was displayed. |
| **Pass/Fail** | Pass |

Table 17: Cancel Guided Tour Test Case

## 3.2 Nonfunctional Tests

Nonfunctional requirements have been established in conjunction with Team A. Team A is establishing the nonfunctional requirements for performance, reliability, and usability. These realms will be considered out of scope for the purposes of this document but can be accessed on the Team A testing plan. Team B is responsible for establishing the nonfunctional requirements for security and scalability. These realms will be considered in scope for the purposes of this document.

|  |  |  |  |
| --- | --- | --- | --- |
| Nonfunctional Requirement | Domain | Description | Team Responsible |
| Response Time | Reliability | The application should respond to user interactions within 5 seconds for tasks such as navigation and setting reminders | Team A |
| Uptime and Availability | Reliability | The application should have a minimum uptime of 99% to ensure consistent access for users, caregivers, and medical professionals. | Team A |
| Backup and Recovery | Reliability | Regular automated backups should be performed to facilitate data recovery in case of system failures or data breaches. | Team A |
| Offline Use | Reliability | Functions such as recording and retrieval of data, which are considered essential to the system, shall be available in case of low connectivity or offline use. | Team A |
| Speech Processing | Usability | Speet to text processing is complete within 5 seconds of initiation. | Team A |
| Video Processing | Usability | Video/imaging is processed within 30 seconds of initiation. | Team A |
| Data Transfer | Security | All site data transferred to external services must be conducted via encrypted tunnels. | Team B |
| Data Encryption | Security | User data, including personal and medical data, should be encrypted both during transmission and storage to prevent unauthorized access. | Team B |
| Authentication | Security | Robust user authentication mechanisms, including biometric options, should be implemented to ensure only authorized users can access the application. | Team B |
| Authorization Levels | Security | Different levels of authorization should be established to manage access to sensitive features and data, such as caregiver accounts. | Team B |
| Privacy Compliance | Security | The application should adhere to relevant data protection regulations such as HIPPA, depending on the jurisdiction and data processed. | Team B |
| Secure Development | Security | Secure coding practices should be observed to minimize vulnerabilities and weak points for cyber-attacks. | Team B |
| User Growth | Scalability | The applications should handle 10,000 concurrent users without performance degrading past established response times | Team B |
| Server Scalability | Scalability | Back-end infrastructure should be designed to scale with demand, appropriating additional resources | Team B |
| Database Scalability | Scalability | The database should be capable of handling increased data load and user interactions without compromising performance. | Team B |

Table 3 – Nonfunctional Requirements

3.2.1 Security

#### 3.2.1.1 NF-Sec-1 - Data Encryption

**Description:** User data, including personal data and medical information, should be encrypted both during transmission and storage to prevent unauthorized access.

**Requirements:** The application shall encrypt data before transmitting or storing personal or medical information.

**Prerequisites:**

* The user has previously registered with the application
* The application is in use on the user’s phone

**Pass/Fail Criteria:**

* Pass: All data is stored and transmitted in an encrypted manner.
* Fail: Data is not encrypted before storage or transmission.

**Assumptions:**

* The user has previously registered with application
* The user is connected to the internet
* The application is active and able to receive requests
* The application can communicate with back-end services

#### 3.2.1.2 NF-Sec-2 – Data Transfer

**Test Case Name:** Data Transfer

**Description:** All data transferred to external services must be conducted utilizing encrypted tunnels such as HTTPS.

**Requirements:** The application shall transfer data to external services using encrypted tunnels.

**Prerequisites:**

* The user has previously registered with the application
* The user has logged into the application
* The user’s device is connected to the internet
* The application is active and able to receive requests
* The application can communicate with back-end services

**Test Data:**

* Email Address: test\_user@testemail.com
* Password: C0mplexPa$$word2@

**Pass/Fail Criteria:**

* Pass: Data is transferred via encrypted tunnels.
* Fail: Data is transferred via unencrypted tunnels.

**Assumptions:**

* The user has previously registered with application
* The user is connected to the internet
* The application is active and able to receive requests
* The application can communicate with back-end services

#### 3.2.1.3 NF-Sec-3 – Authentication

**Description:** Robust user authentication mechanisms, including biometric options, should be implemented to ensure only authorized users can access the application.

**Prerequisites:**

* The user has previously registered with the application
* The user has logged into the application
* The user’s device is connected to the internet
* The application is active and able to receive requests
* The application can communicate with back-end services
* The user has agreed to provide biometric data

**Pass/Fail Criteria:**

* Pass: The user is provided options to authenticate their log in including biometric or two factor authentication.
* Fail: The user is not provided options to authenticate their log in.

**Assumptions:**

* The user has previously registered with application
* The user is connected to the internet
* The application is active and able to receive requests
* The application can communicate with back-end services
* The user opts into authentication methods

#### 3.2.1.4 NF-Sec-4 – Authorization Levels

**Description:** Different levels of authorization should be established to manage access to sensitive information, features, and data, ensuring only authorized users have access.

**Prerequisites:**

* The user has previously registered with the application
* The user has logged into the application
* The user’s device is connected to the internet
* The application is active and able to receive requests
* The application can communicate with back-end services
* The authorized party has previously registered with the application
* The authorized party has been approved for access

**Pass/Fail Criteria:**

* Pass: The authorized party is able to access the primary user account with restrictions on the information they can view.
* Fail: The authorized party is able to access the primary user account with no restrictions on their ability to view information.

**Assumptions:**

* The user has previously registered with application
* The user is connected to the internet
* The application is active and able to receive requests
* The application can communicate with back-end services

#### 3.2.1.6 NF-Sec-5 – Privacy Compliance

**Description:** The application adheres to data protection regulations, such as HIPAA, depending on the jurisdiction and data being processed

**Prerequisites:**

* The user has previously registered with the application
* The user has logged into the application
* The user’s device is connected to the internet
* The application is active and able to receive requests
* The application can communicate with back-end services
* Data protection regulations are available for the region the user is accessing the application from

**Pass/Fail Criteria:**

* Pass: Data protection procedure is in compliance with the local jurisdiction.
* Fail: Data protection procedure is not in compliance with the local jurisdiction.

**Assumptions:**

* The user has previously registered with application
* The user is connected to the internet
* The application is active and able to receive requests
* The application can communicate with back-end services
* Data protection policies are known

#### 3.2.1.7 NF-Sec-6 – Secure Development

**Description:** Secure coding practices should be observed to minimize vulnerabilities and weak points for cyber-attacks.

**Prerequisites:**

* Secure coding practices are established

**Pass/Fail Criteria:**

* Pass: Coding practices are executed according to established expectations
* Fail: Coding practices are not executed according to established expectations

**Assumptions:**

* Secure coding practices are agreed upon by development teams

### 3.2.2 Scalability

#### 3.2.2.1 NF-Sca-1 – User Growth

**Description:** The applications should handle 10,000 concurrent users without performance degrading past established response times

**Prerequisites:**

* Permissible response times have been established
* 10,000 users have registered with the application
* 10,000 users actively send requests through the application

**Pass/Fail Criteria:**

* Pass: The user experience does not degrade past the established permissible response time.
* Fail: The user experience degrades past the established permissible response time.

**Assumptions:**

* Hardware solutions do not bottleneck application performance
* Server solutions have been secured that can scale horizontally with demand

#### 3.2.2.2 NF-Sca-2 – Server Scalability

**Description:** Back-end infrastructure should be designed to scale with demand, appropriating additional resources

**Prerequisites:**

* A server solution has been secured that allows for scaling of resources
* User requests exceed the capacity that is available at a baseline instance

**Pass/Fail Criteria:**

* Pass: The server solution is able to scale with user demand, allowing for a response time that stays within the established permissible range.
* Fail: The server solution is unable to scale with user demand and user experience degrades past the established permissible range.

**Assumptions:**

* The user volume exceeds 10,000 users
* The upscaling of servers is a temporary response to unforeseen user volume
* The servers will scale back down once user volume dissipates

3.2.2.3 NF-Sca-3 – Database Scalability

**Description:** The database should be capable of handling increased data load and user interactions without compromising performance.

**Prerequisites:**

* A database solution has been secured that allows for scaling of resources
* User requests exceed the capacity that is available at a baseline instance

**Pass/Fail Criteria:**

* Pass: The database solution processes above average user interactions without response times falling below the established permissible range.
* Fail: The database solution cannot support above average user interactions and response times fall degrade beyond the established permissible range.

**Assumptions:**

* The user load exceeds 10,000 users
* The upscaling of the database is temporary response to unexpected user volume
* The database will scale back down once user volume dissipates

# 4 Testing Phases

The testing for the CogniOpen application can be broken down into five phases – unit testing, functional testing, integration testing, regression testing, and user acceptance testing (UAT). These phases will ensure that the software written by the two teams developing the application will be resilient, take user input appropriately, and satisfy the agreed upon requirements. Each phase of testing is a critical step in the process of delivering a finished product.

## 4.1 Unit Testing

Unit testing will be used within the CogniOpen application as a tool for developers to ensure that the code that they wrote is functioning as intended. There are two main uses for unit testing for the CogniOpen application – testing UI rendering and testing functionality.

When writing unit tests for the UI, developers are checking to ensure that the UI can load; the buttons, labels, and images are present; and the application is serving them correctly. While not a functional test, it will ensure that the UI has a good chance of looking the way the developer wants it to.

Functional unit tests are more involved. These unit tests should provide an exhaustive test suite for the classes and their associated functions. Every class written for the CogniOpen application shall have a set of unit tests that stress the functions and their algorithmic and input handling. The goal for these tests is to ensure that user data is being sanitized, the logic is functioning as expected, and future updates do not break the code.

## 4.2 Functional Testing

Functional testing will be done by test engineers prior to integration with the greater system. In coordination with the developer, a test engineer will develop a suite of test cases that will ensure that the feature they are developing is going to function according to the system requirements and is user friendly. When the feature is completed, a test engineer can check out the branch where the development was done, run through the suite of tests, and either pass or fail the tests. Depending on the output of functional testing, either the feature will be integrated with the rest of the system, or the developer will have a list of bugs that need to be fixed. This testing will ensure that features are working in a sandbox, and that they are meeting specific system and user needs. The impact that this feature has on other features will not be addressed at this time.

## 4.3 Integration Testing

Integration testing will be done by test engineers after features have been integrated into the main project. The goal for integration testing is to ensure that new features have not broken the interaction between existing features, new features, and the core functionality of the application. While the two teams are trying to develop in isolated swim lanes, it is almost impossible to write code completely in isolation. Inevitably, new code will have an impact on other functions of the system. Integration testing will ensure that new features have meshed successfully and there are no obvious bugs identified. If bugs are identified, developers will fix them, get them merged, and the test engineer will resume execution of their integration testing.

## 4.4 Regression Testing

Regression testing is the last phase of product testing before delivery to the customer. During regression testing, every possible feature within the application will be exercised. Features that have not had any impactful code changes, new features just integrated, and all non-functional requirements will be tested to the furthest possible extent. During regression testing, all buttons, forms, switches, and variations of user input should be used to stress the system. Real-world scenarios should be prioritized during this time of testing. Ideally, very few bugs would show up during this testing phase, but only the most critical bugs should halt delivery. If blocking or critical bugs are identified, they will be fixed by a developer, verified by a test engineer, and testing may resume.

## 4.5 User Acceptance Testing

Finally, after the code has been unit tested, functionally tested, integration tested, and fully regression tested, the software can be delivered to the customer. At this time, the customer will do the final phase of testing – UAT. During UAT, the customer gets a chance to use the system, perform their own testing, and provide direct feedback to the development team. The goal of UAT is to ensure that the two teams developing the CogniOpen application have successfully realized the requirements provided by the customer. If the customer is happy with the software, it will get deployed to production. If the customer is unhappy with certain components of the application, the teams will discuss a path forward to resolving the issues and providing a new version of the software.