**Software Test Plan Document**

CareConnect

University of Maryland Global Campus

SWEN 670 - Software Engineering Capstone

Dr. Mir Assadullah

August 3rd, 2025

Contributors: Diane Angeles, Torie Bias, Juan Gaucin Calderon, Fang Chen, Luke Curran, Ashenafi Grbreegziabhere, Alyssa Marielle Harding, Astha Malla-Paudel, Maria Ramirez, Edwenson Raphael, Dat Truong, Alex Vecchioni, Christian Yawn **Revision History**

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 06/14/2025 | 1.0 | Initial Release | CareConnect Team |
| 06/27/2025 | 2.0 | Update based on feedback | CareConnect Team |
| 08/03/2025 | 3.0 | Update tables and formatting | CareConnect Team |

**Sign-off Sheet**

| **Role** | **Name** | **Signature** | **Date** |
| --- | --- | --- | --- |
| Project Manager | Alireza Minagar |  |  |
| Team Lead | Alyssa Harding |  |  |
| Lead Tester | Juan Gaucin |  |  |
| Client | Dr. Mir Assadullah |  |  |

Table of Contents

[1. Introduction 1](#_Toc205127977)

[1. Purpose 1](#_Toc205127978)

[2. Scope 1](#_Toc205127979)

[3. Audience 2](#_Toc205127980)

[4. Project Documents 2](#_Toc205127981)

[5. Roles and Responsibilities 3](#_Toc205127982)

[6. Acronyms, Definitions, and Abbreviations 4](#_Toc205127983)

[7. References 5](#_Toc205127984)

[2. Tools & Environment 6](#_Toc205127985)

[3. Features to Be Tested 6](#_Toc205127986)

[4. Features Not to be Tested 8](#_Toc205127987)

[5. Test Strategy 8](#_Toc205127988)

[5.1 Types of Testing 8](#_Toc205127989)

[5.1.1 Unit Testing 8](#_Toc205127990)

[5.1.2 Integration Testing 9](#_Toc205127991)

[5.1.3 Performance Testing 9](#_Toc205127992)

[5.1.4 Security Testing 10](#_Toc205127993)

[5.1.5 Usability Testing 10](#_Toc205127994)

[5.2 Test Design Approach 11](#_Toc205127995)

[5.1.6 Risk-Based Testing 11](#_Toc205127996)

[5.1.7 Data-Driven Testing 11](#_Toc205127997)

[5.1.8 Boundary Value Analysis 11](#_Toc205127998)

[5.3 Test Execution Approach 11](#_Toc205127999)

[6. Testing Process 12](#_Toc205128000)

[6.1 Testing Tasks 12](#_Toc205128001)

[6.1.1 Authentication and Onboarding Test Suite 12](#_Toc205128002)

[6.1.2 Dashboard and Navigation Test Suite 15](#_Toc205128003)

[6.1.3 User and Role Management Test Suite 18](#_Toc205128004)

[6.1.4 Scheduling and Notifications Test Suite 22](#_Toc205128005)

[6.1.4.2 Caregiver Shift Scheduling 24](#_Toc205128006)

[6.1.5 Health Data Tracking Test Suite 26](#_Toc205128007)

[6.1.6 AI Integration Test Suite 29](#_Toc205128008)

[6.1.7 Communcation and Media Test Suite 32](#_Toc205128009)

[6.1.8 Device and Third-Party Integration Test Suite 39](#_Toc205128010)

[6.1.9 Billing and Subscription Test Suite 43](#_Toc205128011)

[6.1.10 User Password Reset & Recovery 45](#_Toc205128012)

[6.1.1 Patient Dashboard 47](#_Toc205128013)

[6.1.2 Gamification Module Test Suite 48](#_Toc205128014)

[6.1.3 Social Networking Test Suite 48](#_Toc205128015)

[6.2 Traceability Matrix 48](#_Toc205128016)

[7. Entry and Exit Criteria 50](#_Toc205128017)

[7.1.1 Entry Criteria 50](#_Toc205128018)

[7.1.2 Exit Criteria 50](#_Toc205128019)

[8. Risk and Mitigation 51](#_Toc205128020)

[8.1 Risk Matrix 51](#_Toc205128021)

[8.2 Risk Register 52](#_Toc205128022)

[9. Change Management Procedures 53](#_Toc205128023)

[9.1.1 Initiation 53](#_Toc205128024)

[9.1.2 Review 53](#_Toc205128025)

[9.1.3 Authorization and Implementation 54](#_Toc205128026)

[9.1.4 CareConnect Change Request Form 54](#_Toc205128027)

[10. Defect Management 55](#_Toc205128028)

[Overview 55](#_Toc205128029)

[Defect Tracking Process 55](#_Toc205128030)

[Severity and Priority Levels 56](#_Toc205128031)

[Defect Reporting Tools 56](#_Toc205128032)

[Metrics Tracked 57](#_Toc205128033)

[1. Total number of defects (per sprint) 57](#_Toc205128034)

[2. Number of open vs. closed issues 57](#_Toc205128035)

[3. Average time to resolution 57](#_Toc205128036)

[4. Defect leakage (bugs found in production that escaped testing) 57](#_Toc205128037)

[5. Reopened defect count 57](#_Toc205128038)

[Conclusion 57](#_Toc205128039)

# Introduction

## Purpose

CareConnect is a HIPAA-compliant mobile and web-based platform designed to support communication and collaboration between caregivers and patients. The platform integrates a wide range of features—from user management, health tracking, and medication reminders to emergency response, AI-powered assistance, and subscription billing. A software test plan is necessary to define the testing strategy, objectives, scope, and responsibilities necessary to ensure the successful validation of the CareConnect application.

The Software Test Plan Document will guide the verification and validation activities that confirm whether CareConnect meets its functional, non-functional, and regulatory requirements. It aims to detect defects early, ensure compliance with healthcare standards, and validate that the system performs reliably, securely, and effectively across supported devices and user roles. This document serves as a reference for developers, testers, project managers, and stakeholders throughout the 11-week development cycle, ensuring that the final product meets quality expectations and is ready for production deployment.

## Scope

The scope of this Software Test Plan includes the testing of all core features and subsystems of the CareConnect platform across mobile and web interfaces. It includes the validation of key features and system behavior to ensure the application functions correctly, securely, and reliably across supported devices. Testing will cover both functional and non-functional requirements, with a focus on reliability, usability, performance, and security in compliance with HIPAA standards. The plan also defines what will and will not be tested during the project timeline, helping the team prioritize efforts and manage expectations. This ensures that the delivered product meets quality standards and is ready for deployment in a healthcare environment.

The following test suites define the core functional areas that fall within the scope of this Software Test Plan:

* **Authentication and Onboarding:** Verifies user authentication, registration, and initial onboarding workflows to ensure secure account creation, proper role assignment, and seamless entry into the application environment.
* **Dashboard and Navigation:** Verifies that the dashboard accurately displays patient data and allows users to perform key actions such as viewing, editing, archiving, and adding patient records. It also tests the functionality and usability of the main menu to ensure it provides clear, actionable navigation.
* **User and Role Management:** Verifies the functionality of creating and managing caregiver and patient profiles, including support for linking multiple caregivers to a single patient. Also tests role-based access controls and permissions to ensure users only have access to appropriate features based on their role.
* **Scheduling and Notifications:** Validates the use of pre-defined care templates for scheduling recurring tasks and verifies caregiver shift scheduling, including daily/weekly shift creation, conflict resolution, availability settings, and assignment of caregivers to patient care windows.
* **Health Data Tracking:** Verifies the accurate entry, storage, and display of manually logged health metrics, and ensures that basic reporting and trend visualization features function as intended.
* **AI Integration:** Verifies AI-driven features such as virtual assistant and mood detection during video interactions, including safeguards and disclaimers.
* **Communication and Media:** Validates in-app messaging, audio and video calling, virtual check-in rounds, media uploads, and voice-activated commands. It also verifies the functionality of the Telehealth Bridge and ensures the Emergency SOS mode can be triggered and routed appropriately within the communication framework.
* **Device and Third-Party Integration:** Verifies integration with various platforms and devices to ensure accurate data synchronization and reliable connectivity. This includes wearable devices such as Fitbit, mobile health data platforms like Apple Health and Health Connect, as well as smart home systems and other external health-related services.
* **Billing and Subscriptions:** Validates subscription tier selection, secure payment processing via Stripe or PayPal, and billing lifecycle events such as upgrades, renewals, and cancellations.

## Audience

The audience for this Software Test Plan Document includes the Clients, Team Lead, Project Manager, Developers, Testers & Test Leaders, Business Analyst & Business Analyst Lead, Technical Lead/Architect, Lead UI/UX Designer, UI/UX Designer, MS Teams & GitHub Mentor.

## Project Documents

The Technical Design Document is just one artifact in a suite of documents comprising the software development life cycle for CareConnect. These documents contain important information regarding the project planning, implementation, testing, and more about the CareConnect product. A table with the list of documents, their current versions, and delivery dates can be viewed below.

**Table 1**

*List of Project Documents for CareConnect*

| **Document** | **Version** | **Date** |
| --- | --- | --- |
| Project Plan | 1.0 | 05/31/2025 |
| Software Requirements Document | 1.0 | 05/31/2025 |
| Technical Design Document | 1.0 | 06/14/2025 |
| Software Test Plan | 1.0 | 06/14/2025 |
| Deployment and Operations Guide | 1.0 | TBD |
| Programmer Guide | 1.0 | TBD |
| User Guide | 1.0 | TBD |
| Test Report | 1.0 | TBD |

*Note*. Documents with Date labeled TBD have not been delivered yet.

## Roles and Responsibilities

The CareConnect project is organized into a team of people with different roles and responsibilities. The roles are responsibilities are listed in the table below.

**Table 2**

*Table of roles and responsibilities for the CareConnect project.*

| **Roles** | **Responsibilities** |
| --- | --- |
| **Client**-Dr. Assadullah, Roy Gordon  Ashley Wayne | They are responsible for explaining what requirements are needed and providing feedback on progress. |
| **Team Lead**-Alyssa Marielle Harding | Oversees tasks are being completed. Reports to Project Manager. Attend Lead Meetings. |
| **Project Manager**- Alireza Minagar | Report to Dr. Assadullah., Oversees project progress |
| **Technical Lead/Architect-** Ashenafi Grbreegziabhere, Edwenson Raphael | Lead Developers and Design project architecture |
| **Developers-** Ashenafi Grbreegziabhere, Edwenson Raphael, Alex Vecchioni, Maria Ramirez, Alyssa Marielle Harding, Christian Yawn, Fang Chen, Luke Curran | Execute coding, refers to Technical Lead/Architect |
| **Business Analyst Lead-** Astha Malla-Paudel | Use cases identification, Assembles with stakeholders, overseas the Business Analysts |
| **Business Analyst-** Diane Angeles, Ashenafi Grbreegziabhere, Edwenson Raphael, Dat Truong, Luke Curran, Alyssa Marielle Harding, Juan Gaucin | Refers to the Business Analyst Lead, attends meetings and fulfills functional requirements, assist team lead to determine use cases |
| **Lead Tester-** Juan Gaucin | Collaborate with testers to create testing scenarios, creates test matrix, refers with team leads |
| **Testers**- Dat Truong, Fang Chen, Alyssa Marielle Harding, Torie Bias, Luke Curran | Assist in creating and implementing testing scenarios. Document results of testing. |
| **Lead UI/UX Designer -** Dat Truong  Luke Curran | Collaborate with leads and design wireframes for product mockups |
| **UI/UX Designer-** Astha Malla-Paudel, Christian Yawn, Edwenson Raphael | Wireframe designing for mockups, collaborate with developers |
| **MS Teams & GitHub Mentor -** Robert Wilson | MS Teams scheduling, Assist with GitHub and MS Teams |

## Acronyms, Definitions, and Abbreviations

**Table 3**

| **Definitions** | **Acronyms** | **Abbreviations** |
| --- | --- | --- |
| **Care Connect-** Application for this project that will be created to help manage patient healthcare needs. | **UX-** User Experience  **UI-**User Interface | **APP**-Application |
| **Flutter-** Will be used to code the UI/UX across multiple applications | **TBD-** To be determined | **"+”= To add patients** |
| **Caregiver-**person caring for the patient such as ADLs | **AI-** Artificial Intelligence | **AI Assistant (Ask)-** enables Caregivers to ask questions. |
| **Patient**- Person receiving care with utilization of the application | **ADLs**-Activities of daily living | **Ad-hoc-** needs to be addressed right away. |
| **Family Member-** Person that oversees care of the patient | **API-**Allows communication between different software applications. | **Via-**By way of. |
| **Android-**Mobile operating system by Google | **HIPPA-** Health Insurance Portability and Accountability Act | **ID-** Identification |
| **IOS-**Mobile OS by Apple | **OS-** Operating System | **Vs.** - Versus |
| **Dart Tool-**Will be used to code the UX/UI | **SMS-OTP-** Short Message Service – One Time Password | **Etc**.- Et cetera (and so on) |
| **Emergency SOS-** Notifies caregivers to track the location of a patient if they are having an emergency. If they do not respond, then two other points of contact will be reached. If they don't respond, 911 will be called. | **QA-** Quality Assurance | **REQ-** Requirements |
| **Bugs-**Mistakes, errors or faults that may cause the system to not function properly. | **NPL-**Natural Language Processing |  |
|  | **CI/CP pipelines-** Continuous Integration/Continuous Deployment |  |
|  | **TC-**Test Case |  |

*Acronyms, Definitions, and Abbreviations of the Software Test Plan*

## References

AlphaSoft. (2023, August 5). *Short-Term Memory System (STeMS) project plan* (Version 4.0) [Project documentation]. University of Maryland Global Campus.

*CareConnect High Level Requirements* (n.d.). UMGC. [Care Connect High Level Requirements.docx](https://umgcdev361.sharepoint.com/:w:/r/sites/SWEN670Summer2025/Shared%20Documents/Joint%20Collab%20(Care%20Connect)/Care%20Connect%20High%20Level%20Requirements.docx?d=w8c0ab524c4fe42eb9ac877d4962b641d&csf=1&web=1&e=YRy6gv)

*dart: The Dart command-line tool*. (n.d.). Dart. <https://dart.dev/tools/dart-tool>

*Flutter - Build apps for any screen*. (n.d.). <https://flutter.dev/>

*Health Insurance Portability and Accountability Act of 1996 (HIPAA)*. (2024, September 10). Public Health Law. <https://www.cdc.gov/phlp/php/resources/health-insurance-portability-and-accountability-act-of-1996-hipaa.html>

OpenAI. (2025). *ChatGPT (June 13 version)* [Large language model]. <https://chat.openai.com/>

# Tools & Environment

**Table 4**

*Mediums that will be used for testing*

| **Medium** | **Hardware** | **Browser** | **Operating Systems** | **Simulated Environments** | **Backend APIs** |
| --- | --- | --- | --- | --- | --- |
| Mobile Device | Tablet: Apple/ Android | Chrome (136.0.7103.114), Safari (18.2), | iOS 16–17 & Android 12–14 | Xcode, Android Studio | Restful APIs (Patient, Caregiver, Notes, SOS) |
| Mobile Device | Smartphones: Apple/Android | Chrome (136.0.7103.114), Safari (18.2), Firefox (133.0), | iOS 16–17 & Android 12–14 | Flutter emulator (iOS/Android) | Scheduling API, Auth API |
| Desktop/Laptop | Windows 11 PC, MacBook Pro | Chrome (136.0.7103.114), Safari (18.2), Firefox (133.0), Edge (131.0.2903.146) | Windows 11, macOS Ventura | Windows, MacOS | Family Access API, Notification API |
| CI/CD Pipelines | Firebase Cloud, GitHub Actions | None | Docker Containers | Firebase Cloud, GitHub Actions | API Automation Hooks, |

# Features to Be Tested

**Table 5**

*Table of features to be tested for the CareConnect project.*

| **Feature** | **Description** | **Priority** |
| --- | --- | --- |
| User Registration | * Allow Caregivers and Patients to create accounts using email/password. | * High |
| Login/Logout | |  | | --- | |  |      |  | | --- | | * Ensures successful login using credentials. * Redirects to appropriate dashboard. * Supports persistent session unless logged out | | * High |
| Patient Dashboard (Caregiver view) | * Displays a list of linked patients with name, thumbnail, and last interaction. * Includes action icons (View, Edit, Archive) and a floating “+” button to add new patients | * High |
| Add Patient | * Opens from input name, relationship and contact info. | * High |
| Edit Patient | * Allow caregivers to update patient details and medical information. | * High |
| Archive Patient | |  | | --- | |  | | * Allows to move patient cards to an inactive list. * Validates archive logic and retrieval. | | * Medium |
| Patient Dashboard (Patient View) | * Displays list of linked caregivers with their profile, availability status and options to call, message, or remove caregivers. | * High |
| Healthcare Notes & AI Assistant (Ask) | * Allow caregivers to ask AI questions. | * Medium |
| Notification Channels | * Allows caregiver to select the default channel (Push, SMS, Email) for reminders per task type. | * Medium |
| Password Reset | * Allows caregiver/Patient to reset password in the event of forgot password. | * High |
| Scheduling & Reminders | |  | | --- | |  |  |  | | --- | | * Verifies timely delivery of reminders via selected channel, and escalation logic if missed. | | * Medium |
| Family Access | * Allows caregivers to invite family members to view selected patient data with no editing privileges. * Includes customization and revoking access logic | * Medium |
| Emergency SOS | * Patients can trigger emergency alerts with location and audio. * Escalates if not acknowledged. * Caregivers get full-screen alerts and location tracking. | * High |
| Symptoms Tracking | * Patients log daily symptoms via push prompts; caregivers see trends and alerts for critical scores. | * Medium |
| Gamification | * Positive messages or encouragement are displayed after task completion. | * Medium |
| Mood and Wellness Logging | * Patients log in moods using emoji shots. * Weekly trend reports are sent via email. | * Medium |

# Features Not to be Tested

**Table 6**

*Table of features not to be tested for the CareConnect project.*

| **Feature** | **Reason for Exclusion** |
| --- | --- |
| Stress testing under very heavy load | * Not in scope for this functional QA phase. |
| Integration Failures in Third-Party Services | * Only positive test cases will be executed. |
| UI Design consistency | * UI consistency will be covered under UI/UX testing separately. |
| Advanced Voice Command Testing | * While voice features are included, detailed testing of all voice command variations will be deferred until voice NLP integration stabilizes |
| Analytical Dashboard Accuracy | * Out of Scope for a functional test. |

# Test Strategy

## 5.1 Types of Testing

### Unit Testing

Definition: Unit Testing is a testing methodology where testing is performed on an individual unit /component of the application independently to verify that the unit/component functions correctly as expected.

Purpose:

* To validate that each function performs as expected independent of the rest of the system.
* Catch the error early in the development cycle
* Helps to test automation and CI/CP pipelines

Performer: Developers perform unit testing before integration or system testing.

Tools used:

* Flutter test: It’s a built-in test package in Flutter SDK
* Test: It’s Dart’s general-purpose unit test framework.

Coverage Goal: The coverage goal is to have 90% coverage to make sure almost every function and condition is tested.

Testing Techniques used:

**Table 7**

*Table of testing techniques used for the CareConnect project.*

| **Technique Used** | **Description** |
| --- | --- |
| Black Box Testing | Test the function without knowing the internal code |
| White Box Testing | Test the function using developer knowledge |
| Automated Testing | Use the test frame to execute test automatically |

### Integration Testing

### Performance Testing

Performance testing is critical for CareConnect due to the nature of supported features like real-time communication and the need for reliable emergency response capabilities. The application must maintain optimal performance under varying load conditions while handling sensitive health data transactions, medication reminders, and emergency SOS alerts without degradation. Performance testing validates that the system can support the target user base while maintaining response times that ensure patient safety and caregiver efficiency. Given the healthcare context, performance failures could directly impact patient care quality, making this testing domain essential for system validation and user confidence.

Key areas that are to be tested under performance are as follows:

* Load Testing
  + Normal load – test under normal or average conditions with an average set of concurrent users.
  + Peak load- test under intensive usage with an above average number of concurrent users.
  + Response time – test system function’s response time against pre-established metrics.
  + Database performance – test using realistic data volumes.
* Stress Testing
  + Load limits – test an ever-increasing load to determine failure point.
  + Resource utilization – test the host system’s resource utilization by the app.
  + Scalability – test the system’s ability for future growth.

### Security Testing

Security testing represents one of the most critical testing domains for CareConnect, given the application's responsibility for protecting sensitive information and ensuring compliance with stringent healthcare regulations including HIPAA. Robust security validation is essential for protecting patient privacy and maintaining regulatory compliance. Security vulnerabilities in healthcare applications can result in severe consequences including data breaches exposing patient medical records, financial penalties from regulatory bodies, loss of patient trust, and potential legal liability. The multi-user nature of CareConnect, with caregivers, patients, and family members accessing shared health data, creates complex security requirements that must be thoroughly validated through comprehensive testing approaches.

As part of security testing, the following areas will be tested:

* Authentication Systems
  + Validate access controls.
  + Validate session management and timeouts.
* Data Protection
  + Verify data security at transit and at rest.
  + Continuously validate regulatory compliance.
* Vulnerability Assessments
  + Conduct routing scans for vulnerabilities.
  + Ensure input is sanitized and validated.

### Usability Testing

Usability testing is fundamental to CareConnect's success, as the application serves a diverse user base including elderly patients, family caregivers with varying technical expertise, and professional healthcare providers who require efficient workflows. The healthcare context demands that the interface be intuitive and accessible, as usability barriers can directly impact medication adherence, emergency response times, and overall care quality. Poor usability in healthcare applications can lead to user abandonment, medication errors, delayed emergency responses, and increased caregiver burden. The application must accommodate users who may be experiencing stress, physical limitations, or cognitive challenges while managing health-related tasks. Additionally, the cross-platform approach requires validation that all critical functions remain accessible and efficient across different screen sizes and interaction methods, ensuring that caregivers can provide effective support regardless of their device or technical proficiency level.

Usability will be validated through the following test areas:

* Test accessibility features.
* Test for error clarity, helpfulness, and error recovery procedures.
* Test for navigation and task completion intuitiveness.

## 5.2 Test Design Approach

The CareConnect test design approach employs a systematic methodology that ensures comprehensive coverage while optimizing testing efficiency and effectiveness. Our approach integrates multiple testing design techniques to address the complex requirements of healthcare applications, regulatory compliance needs, and diverse user scenarios.

### Risk-Based Testing

Test case prioritization is driven by impact assessment and complexity. Critical functions such as Emergency SOS, medication management, and patient data access receive the highest testing priority. Risk assessment considers potential patient safety impacts, regulatory compliance requirements, and system availability needs.

### Data-Driven Testing

Because healthcare applications require validation across multiple data combinations and edge cases. Our test design incorporates comprehensive data sets. Test cases utilize this data input to validate system behavior across different functions.

### Boundary Value Analysis

Ttest cases specifically target boundary conditions such as medication dosage limits, vital sign thresholds, and system capacity limits. This approach identifies potential failure points where the app performance and reliability could be compromised.

## 5.3 Test Execution Approach

The CareConnect test execution strategy employs several strategies to maximize efficiency while ensuring adequate testing across all domains. Parallel testing will enable simultaneous testing across different domains and on different platforms by the different teams in charge of each functional area of CareConnect.

Continuous testing integration and automated testing will ensure that all changes are tested and vetted before being incorporated into the base software system. Versioning control will also aid in maintaining a testing environment during development.

# Testing Process

## 6.1 Testing Tasks

The testing tasks are organized into comprehensive test suites covering all aspects of the CareConnect application. Each test suite contains multiple detailed test cases with full requirement traceability.

### 6.1.1 Authentication and Onboarding Test Suite

**Scope**: Complete user authentication, registration, and onboarding workflows  
**Total Test Cases**: 8  
**Requirements Coverage**: REQ-3.1.1 - REQ-3.2.8

|  |  |
| --- | --- |
| **Prerequisites for this test:**  Fresh app installation (no user logged in)  Internet connectivity  Test email/password credentials  Mobile device ready to receive SMS  Google account available for SSO | **Software Versions:**  Application: CareConnect  Software Version: 1.0  Operating System: iOS 18 |
| **Priority:** High |
| **Requirements validated:** FR-3.1.1, FR-3.1.2, FR-3.1.4–3.1.7, FR-3.1.9–3.1.10, FR-3.2.1–FR-3.2.8 | |
| **TEST EXECUTOR:** | |
| **TEST SCRIPT STEPS/RESULTS** | |

| **STEP** | **TEST STEP/INPUT** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/FAIL** |
| --- | --- | --- | --- | --- |
| **1.1: Welcome Screen Display and Navigation (Test case name)** | | | | |
| 1. | Install CareConnect application on target device | Welcome screen displays only on first launch |  |  |
| **1.2: User Registration – Email and Password** | | | | |
| 1. | Launch the app. On the welcome screen, tap Register. | Registration screen appears with fields for email and password. |  |  |
| 2. | Enter a valid email and a strong password; tap Submit. | Account is created; user sees a confirmation or is taken to dashboard. The app sends a confirmation email/SMS (per FR-3.1.10). |  |  |
| 3. | Close and relaunch the app; attempt to log in with the new credentials. | Login succeeds immediately using the created account (per FR-3.1.9). |  |  |
| **1.2: User Registration - SMS OTP** | | | | |
| 1. | Launch the app; tap Register, then choose Register with OTP option. | OTP registration screen opens, prompting for phone number. |  |  |
| 2. | Enter a valid mobile number and tap Send OTP. | An SMS OTP is sent to the number; OTP input field appears (per FR-3.1.6). |  |  |
| 3. | Enter the received OTP code and submit. | Account is created; user sees confirmation or dashboard. |  |  |
| 4. | Attempt to log in with the associated email/phone and password (or OTP). | Login succeeds; new account is active (per FR-3.1.7 and FR-3.1.10). |  |  |
| **1.3: User Registration – Google SSO** | | | | |
| 1. | From the welcome or register screen, tap Register with Google. | Google OAuth flow starts. |  |  |
| 2. | Authenticate using a valid Google account. | User is returned to the app, and account is created (per FR-3.1.4–3.1.5). |  |  |
| 3. | Verify the user lands on the appropriate dashboard/home screen. | The app shows the user’s dashboard; a confirmation email/SMS is sent if required (per FR-3.1.10). |  |  |
| **1.4: User Registration – Invalid Email Format** | | | | |
| 1. | Launch the app; tap Register. | Registration screen appears. |  |  |
| 2. | Enter an invalid email (e.g. “user@domain”, missing “.com”) and a password; tap Submit. | The app blocks submission and displays an error about invalid email format (per FR-3.1.2). |  |  |
| 3. | Correct the email and submit it again. | Registration then succeeds as in Test Case 1 (account is created and user proceeds). |  |  |
| **1.5: Login/Logout Flows – Email and Password Login** | | | | |
| 1. | Launch the app; on the login screen enter a valid email and password; tap Login. | User is authenticated (session created per FR-3.2.3). The app navigates to the user’s dashboard. |  |  |
| 2. | Close and reopen the app (or switch tabs); ensure the session is persisted (if allowed by idle timeout). | The user remains logged in without re-entering credentials (per FR-3.2.3). |  |  |
| **1.6: Login/Logout Flows – Login w/ Invalid Credentials** | | | | |
| 1. | Launch the app; enter an unregistered email or wrong password; tap Login. | Authentication fails. An error message appears (e.g. “Invalid credentials”). No navigation to dashboard (session is not created, per FR-3.2.1). |  |  |
| 2. | Correct the credentials and tap Login. | Login succeeds and navigates to dashboard. |  |  |
| **1.7: Login/Logout Flows – Login w/ Google SSO** | | | | |
| 1. | On the login screen, tap Login with Google. | Google OAuth flow starts. |  |  |
| 2. | Authenticate using a valid Google account. | User is returned to the app, and a session is created (per FR-3.2.2 and FR-3.2.3). |  |  |
| 3. | Verify the user lands on the dashboard screen. | Dashboard is shown and user can use the app normally. |  |  |
| **1.8: Login/Logout Flows – Logout** | | | | |
| 1. | While logged in, tap the Logout button/menu. | User is logged out immediately; session data and tokens are cleared (per FR-3.2.6). |  |  |
| 2. | Attempt to navigate back or access a protected screen. | The app redirects to the login screen (per FR-3.2.7). No access to previous sessions. |  |  |

### 6.1.2 Dashboard and Navigation Test Suite

**Scope**: displays accurate patient data, allows interaction (view, edit, archive), support adding new patients; Main menu consists actionable navigation patient management lists.

**Total Test Cases:** 6

**Requirements Coverage**: REQ-5.4.1-REQ-5.4.5

|  |  |
| --- | --- |
| **Prerequisites for this test:**   * Linked caregiver and patient profiles. * Functioning backend for patient * Stable internet connection. | **Software Versions:**  Application: CareConnect  Software Version: 1.0  Operating System: iOS 18 |
| **Priority:** High |
| **Requirements validated:** REQ-5.4.1, REQ 5.4.2, REQ 5.4.3, REQ 5.4.4. REQ 5.4.5 (SRS) | |
| **TEST EXECUTOR:** | |
| **TEST SCRIPT STEPS/RESULTS** | |

| **STEP** | **TEST STEP/INPUT** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/FAIL** |
| --- | --- | --- | --- | --- |
| **Dashboard loads upon Caregiver login** | | | | |
| 1. | * Launch app * Login as caregiver | * Dashboard loads displaying list of patients.      * Dashboards display correct patient info. |  |  |
| **Add Patient using** | **“+”** |  |  |  |
| 2 | * Tap the floating icon “+” button. * Fill in patient name, relationship, contact info, upload media and click submit. | * Patient input from appears. * Patient is added and appears in patient list in Dashboard. |  |  |
| **Edit Patient** |  |  |  |  |
| 3 | * Tap “Edit” on patient card. * Modify the fields as needed. * Tap “Save button” | * Updated patient info appears in Dashboard. |  |  |
| **Archive Patient** |  |  |  |  |
| 4 | Tap ‘Archive” icon and confirm the dialog. | * Archive Patient is removed from the active list. * Patient is added to the archive. |  |  |
| **Real-time update** |  |  |  |  |
| 5 | * Refresh dashboard. * Or wait for update. | * Patient list updates in real-time. |  |  |
| **Menu Items** |  |  |  |  |
| 6 | * Displays list of Menu items (Billing and Subscription Management, Schedule Management, Tracking & Monitoring, Notification Setting, Telehealth Bridge, Health Care Notes, SOS Notification). | * Caregiver is able to navigate around the menu lists. |  |  |

### 6.1.3 User and Role Management Test Suite

**Scope**: All aspects of user and role management including caregiver profile, patient profile, multi-caregiver support, profile linking, and access control & permissions.

**Total Test Cases**: 7  
**Requirements Coverage**: REQ-5.3.1.1- REQ-5.3.5.3

|  |  |
| --- | --- |
| **Prerequisites for this test:**   * Caregiver logged in * Patient logged in * Functioning backend for caregiver * Functioning backend for patient * Stable internet connection. | **Software Versions:**  Application: CareConnect  Software Version: 1.0  Operating System: iOS 18 |
| **Priority:** High |
| **Requirements validated:** REQ-5.3.1.1, REQ-5.3.1.2, REQ-5.3.1.3, REQ-5.3.1.4, REQ-5.3.2.1, REQ-5.3.2.2, REQ-5.3.2.3, REQ-5.3.2.4, REQ-5.3.2.5, REQ-5.3.2.6, REQ-5.3.3.1, REQ-5.3.3.2, REQ-5.3.4.1, REQ-5.3.4.2, REQ-5.3.5.1, REQ-5.3.5.2, REQ-5.3.5.3 | |
| **TEST EXECUTOR:** | |
| **TEST SCRIPT STEPS/RESULTS** | |

| **STEP** | **TEST STEP/INPUT** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/FAIL** |
| --- | --- | --- | --- | --- |
| **Caregiver Profile information updated** | | | | |
|  | 1. Open Caregiver Dashboard 2. Open My Profile 3. Update profile information 4. Press “Save” | 1. Information is updated in the database 2. Dashboard information reflects changes to profile information |  |  |
| **Patient Profile information updated** | | | | |
| 2. | * Patient Profile information updated * Open My Profile * Update profile information * Press “Save” | * Patient Profile information updated * Patient Profile information updated |  |  |
| **Allow multiple Caregivers to link to a Patient** | | | | |
| 3. | * *Caregiver 1* opens Caregiver Dashboard * *Caregiver 1* presses “+” to open Add New Patient form * *Caregiver 1* links to patient * *Caregiver 2* opens Caregiver Dashboard * *Caregiver 2* presses “+” to open Add New Patient form * *Caregiver 2* links to patient | * Patient appears on *Caregiver 1*’s Caregiver Dashboard * Patient appears on *Caregiver 2*’s Caregiver Dashboard * Both Caregivers appear on patient’s Patient Dashboard |  |  |
| Allow Profile Linking between Caregiver and Patient | | | | |
| 4. | * The caregiver opens the Caregiver Dashboard * The caregiver presses “+” to open the Add New Patient form * The caregiver fills in the patient’s information in the form and presses “Submit” | * The caregiver shall have access to all of the patient’s records. * The patient appears on the caregiver’s Caregiver Dashboard * The caregiver appears on the patient’s Patient Dashboard |  |  |
| 5. | * The caregiver opens the Caregiver Dashboard * The caregiver presses “+” to open the Add New Patient form * The caregiver opens the hamburger menu for more options * The caregiver selects the QR Code option to scan a patient’s QR Code * The caregiver scans a patient’s unique QR Code * The patient’s information is auto filled into the form and the caregiver presses “Submit” | * The caregiver shall have access to all of the patient’s records. * The patient appears on the caregiver’s Caregiver Dashboard * The caregiver appears on the patient’s Patient Dashboard |  |  |
| **All caregiver features and linked patient information are shown to caregiver** | | | | |
| 6. | * The caregiver opens the Caregiver Dashboard | * The caregiver is shown a list of all linked patients * The caregiver has access to all linked patient records * The caregiver has access controls to all caregiver functionalities. |  |  |
| **All patient features are shown to patient** | | | | |
| 7. | * The patient opens the Patient Dashboard | * The patient is shown a list of all linked caregivers * The patient has access controls to all patient functionalities. |  |  |

### 6.1.4 Scheduling and Notifications Test Suite

**Scope**: This test suite assigns custom care templates and pre-defined tasks to patients.  
**Total Test Cases**: 4  
**Requirements Coverage**: REQ-5.4.1.1 - 5.4.2.5

|  |  |
| --- | --- |
| **Prerequisites for this test:**  Fresh app installation (user logged in)  Internet connectivity  Test email/password credentials  Mobile device ready to receive SMS | **Software Versions:**  Application: CareConnect  Software Version: 1.0  Operating System: iOS 18 |
| **Priority:** High |
| **Requirements validated:** REQ-5.4.1.1 - 5.4.2.5 | |
| **TEST EXECUTOR:** | |
| **TEST SCRIPT STEPS/RESULTS** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STEP** | **TEST STEP/INPUT** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/FAIL** | **Defects Found** |
| **Caregiver can assign custom task to a patient** | | | | |  |
| 1. | Create a new patient. | New patient is created. |  |  |  |
| 2. | Navigate to the Tasks section of the new patient. | The Tasks section can be viewed from the patient dashboard and is empty. |  |  |  |
| 3. | Caregiver selects ‘Add Task’ button | Assign Task Menu is shown, containing ‘Custom Task’ and task templates. |  |  |  |
| 4. | Caregiver selects ‘Custom Task’ | The ‘Add Custom Task’ form is displayed with empty fields. |  |  |  |
| 5. | Caregiver enters task details and selects ‘Save’ | The new task is displayed under the Tasks section. |  |  |  |
| **Caregiver can assign task from template to a patient** | | | | |  |
| 1. | Create a new patient. | New patient is created. |  |  |  |
| 2. | Navigate to the Tasks section of the new patient. | The Tasks section can be viewed from the patient dashboard and is empty. |  |  |  |
| 3. | Caregiver selects ‘Add Task’ button | Assign Task Menu is shown, containing ‘Custom Task’ and task templates. |  |  |  |
| 4. | Caregiver selects ‘Sleep’ | The ‘Add Task’ form is displayed with pre-filled fields. |  |  |  |
| 5. | Caregiver reviews task details and selects ‘Save’ | The new task is displayed under the Tasks section. |  |  |  |
| **Caregiver can edit a task assigned to a patient** | | | | |  |
| 1. | Caregiver assigns a task to a patient | The new task is displayed under the Tasks section. |  |  |  |
| 2. | Caregiver selects ‘Edit’ for newly created task. | The ‘Add Task’ form is displayed with pre-filled fields for the selected task. |  |  |  |
| 3. | Caregiver edits a field and selects ‘Save’ | The edited task is displayed under the Tasks section. |  |  |  |
| 4. | Caregiver selects the edited task. | The task details are displayed reflecting the changes made. |  |  |  |
| **Caregiver can edit a task assigned to a patient** | | | | |  |
| 1. | Caregiver assigns a task to a patient | The new task is displayed under the Tasks section. |  |  |  |
| 2. | Caregiver selects ‘Delete Task’ for newly created task. | The ‘Delete Task’ form is displayed asking for confirmation to delete the task. |  |  |  |
| 3. | Caregiver selects ‘Delete’ | The selected task is no longer displayed under the Tasks section. |  |  |  |

## 6.1.4.2 Caregiver Shift Scheduling

Scope:  
 This test suite ensures caregiver shift scheduling allows for defining daily/weekly shifts, conflict resolution, caregiver availability settings, and linking caregivers to specific patient care windows.

Total Test Cases: 5

Requirements Coverage: REQ-6.2.1 to REQ-6.2.6

Prerequisites:  
 - At least two caregivers are registered and linked to a patient  
 - Calendar module and availability service are integrated  
 - UI for shift scheduling is implemented

Software Versions:  
 - Application: CareConnect  
 - Software Version: 1.0  
 - Operating System: iOS 18 / Android 14

Priority: High

Test Executor:

Requirements Validated: FR-6.2.1 through FR-6.2.6

| STEP | TEST STEP/INPUT | EXPECTED RESULTS | ACTUAL RESULTS | PASS/FAIL |
| --- | --- | --- | --- | --- |
| 2.1 | Caregiver opens the “Shift Scheduling” screen from the menu | Weekly view of caregiver availability is displayed |  |  |
| 2.2 | Add a new shift from 8AM–12PM on Monday | Shift is added and saved under caregiver profile |  |  |
| 2.3 | Attempt to add overlapping shift for same caregiver (e.g., 10AM–2PM) | App shows validation message: “Overlapping shift detected” |  |  |
| 2.4 | Assign the created shift to a patient | Patient calendar reflects caregiver assigned for that time block |  |  |
| 2.5 | Remove a scheduled shift | Shift is removed from caregiver profile and patient calendar updates |  |  |

### 6.1.5 Health Data Tracking Test Suite

**Scope**: Track patient symptoms in CareConnect application  
**Total Test Cases**: 16  
**Requirements Coverage**: REQ-5.6.1.1 - 5.6.14.8

|  |  |
| --- | --- |
| **Prerequisites for this test:**  Fresh app installation (user logged in)  Internet connectivity  Test email/password credentials  Mobile device ready to receive SMS | **Software Versions:**  Application: CareConnect  Software Version: 1.0  Operating System: iOS 18 |
| **Priority:** High |
| **Requirements validated:** REQ-5.6.1.1 - 5.6.14.8 | |
| **TEST EXECUTOR:** | |
| **TEST SCRIPT STEPS/RESULTS** | |

| **STEP** | **TEST STEP/INPUT** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/FAIL** |
| --- | --- | --- | --- | --- |
| **1.1: Table exists and symptoms are new (OpenAI, 2025)** | | | | |
| 1. | Ensure symptoms table exists and is empty. | Symptom table is empty. |  |  |
| **1.2 Table exists with some symptoms already present (OpenAI, 2025)** | | | | |
| 1. | Populate symptoms table with a subset of default symptoms. | Only missing symptoms are inserted. |  |  |
| **1.3 Table does not exist (OpenAI, 2025)** | | | | |
| 1. | Delete/disable symptoms table before deployment. | Deployment exits gracefully without inserting symptoms. |  |  |
| **1.4 Attach Default Symptoms to Patient (OpenAI, 2025)** | | | | |
| 1. | Create a new patient. | New patient is created. |  |  |
| 2. | Navigate to the symptom section. | Default symptoms are displayed with “+ Start Date” option. |  |  |
| **1.5 Add Custom Symptoms (OpenAI, 2025)** | | | | |
| 1. | Click “Add Symptom”. | Add symptom page appears. |  |  |
| 2. | Enter unique name, category, and critical level. | Symptom is added globally and associated with patient. |  |  |
| **1.6 Symptom Notification Push (OpenAI, 2025)** | | | | |
| 1. | Wait for scheduled notification time. | Nothing happens until scheduled notification time. |  |  |
| 2. | Notification time occurs. | Patient receives symptom update notification, which remains visible until acknowledged. |  |  |
| **1.7 Respond to Symptom Notifications (OpenAI, 2025)** | | | | |
| 1. | Receive notification. | Notification received. |  |  |
| 2. | Submit response. | Response is placed in service. |  |  |
| **1.8 View Patient’s Symptoms (OpenAI, 2025)** | | | | |
| 1. | Open patient profile > symptoms. | System shows all active symptoms and latest responses. |  |  |
| 2. | No symptoms. | Display message: “No active symptoms found.” |  |  |
| **1.9 View Symptom Graph (OpenAI, 2025)** | | | | |
| 1. | Select symptom graph. | Line chart shows historical data, supports filtering, and tooltips. |  |  |
| **1.10 Alert Caregiver of Critical Symptoms (OpenAI, 2025)** | | | | |
| 1. | Patient submits high-severity symptom. | System alerts caregiver with patient details and logs the alert. |  |  |
| **1.11 Insert Default Meal Questions (OpenAI, 2025)** | | | | |
| 1. | Trigger first-time deployment. | Questions are inserted if table exists and isn’t already filled. |  |  |
| **1.12 Add Custom Meal Question (OpenAI, 2025)** | | | | |
| 1. | Caregiver adds a question not in the system. | Question is added globally and linked to patient. |  |  |
| 2. | Prevent duplication. | Duplicate association is blocked with a message. |  |  |
| **1.13 Patient Logs Meal Entry (OpenAI, 2025)** | | | | |
| 1. | * Patient submits meal using text. * Patient submits voice entry. * Patient uploads meal image. |  |  |  |
| **1.14 Caregiver Views Meal Logs (OpenAI, 2025)** | | | | |
| 1. | * View full meal log timeline. * Filter logs by date. | Entries are shown with time, question, and input format. |  |  |
| **1.15 Mood vs Medication/Symptom Trends (OpenAI, 2025)** | | | | |
| 1. | Access dashboard, select filters. | Synchronized, filterable, comparative graphs. |  |  |
| **1.16 Notify Caregiver of Negative Mood Streak (OpenAI, 2025)** | | | | |
| 1. | Patient logs 3 consecutive sad moods. | Caregiver is notified with date range and context. |  |  |

### 6.1.6 AI Integration Test Suite

**Scope**: AI Integrations in CareConnect application  
**Total Test Cases**: 10  
**Requirements Coverage**: REQ-5.7.1.1 - REQ-5.7.2.8

|  |  |
| --- | --- |
| **Prerequisites for this test:**  Fresh app installation (user logged in)  Internet connectivity  Test email/password credentials  Mobile device ready to receive SMS  AI Integration Complete | **Software Versions:**  Application: CareConnect  Software Version: 1.0  Operating System: iOS 18 |
| **Priority:** High |
| **Requirements validated:** REQ-5.7.1.1 - REQ-5.7.2.8 | |
| **TEST EXECUTOR:** | |
| **TEST SCRIPT STEPS/RESULTS** | |

| **STEP** | **TEST STEP/INPUT** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/FAIL** |
| --- | --- | --- | --- | --- |
| **1.1: Submit Patient Mood Entry via Image (OpenAI, 2025)** | | | | |
| 1. | Patient uploads an image via the mobile app. | * Image is securely stored. * System extracts facial landmarks and infers mood with a confidence score. * Entry is timestamped and logged in mood history. |  |  |
| **1.2 Submit Patient Mood Entry via Video (OpenAI, 2025)** | | | | |
| 1. | Patient records and submits a short video. | * System processes facial expressions and voice (if applicable). * AI assigns inferred mood with timestamp and confidence score. |  |  |
| **1.3 Submit Mood via Text Entry (OpenAI, 2025)** | | | | |
| 1. | Patient types a journal entry. | * Text is analyzed for sentiment. * System infers mood, logs it with a confidence score and timestamp. |  |  |
| **1.4 Caregiver Views AI-Inferred Mood on Dashboard (OpenAI, 2025)** | | | | |
| 1. | Caregiver opens the mood dashboard. | * Both AI-inferred and manually submitted mood logs are shown. * AI entries include mood, timestamp, and confidence score. |  |  |
| **1.5 Data Privacy Compliance Check (OpenAI, 2025)** | | | | |
| 1. | Submit a mood entry (text, image, or video). | * Entry is processed in compliance with HIPAA/GDPR. * No sensitive data is exposed to unauthorized users. |  |  |
| **1.6 Administrator Updates AI Model (OpenAI, 2025)** | | | | |
| 1. | * Admin logs in to AI settings. * Uploads a new model version. | * System accepts the new model. * Inference results improve accordingly. |  |  |
| **1.7 Caregiver Submits AI Question via Text (OpenAI, 2025)** | | | | |
| 1. | Caregiver types a non-clinical question (e.g., "What are common signs of dehydration?"). | AI responds clearly and conversationally. |  |  |
|  |  |  |  |  |
| **1.8 Caregiver Submits AI Question via Voice (OpenAI, 2025)** | | | | |
| 1. | Caregiver speaks a question into the app. | * System converts voice to text. * Text is processed and AI provides a relevant answer. |  |  |
| **1.9 Submit Clinical Question in AI Feature (e.g., "Should I adjust medication?") (OpenAI, 2025)** | | | | |
| 1. | Caregiver submits a question needing medical judgment. | * System flags the question. * Displays message: “Please consult a licensed medical professional.” |  |  |
| **1.10 View Interaction History (OpenAI, 2025)** | | | | |
| 1. | Access AI assistant log as an admin. | Display of all submitted questions and AI responses with timestamps. |  |  |

### 6.1.7 Communcation and Media Test Suite

**Scope**: In-App Messaging, Audio and Video Calling, Emergency SOS Mode, Virtual Check-in Rounds, Media Uploads, Voice-Activated Commands, and Telehealth Bridge  
**Total Test Cases**: 31  
**Requirements Coverage**: REQ-5.8.1.3.1–5.8.1.3.5, REQ-5.8.2.3.1–5.8.2.3.6

|  |  |
| --- | --- |
| **Prerequisites for this test:**  Two registered users (one caregiver, one patient) linked as contacts  Both users have internet, microphone and camera access enabled  Both users are signed in on two devices  Both users have an active internet connection | **Software Versions:**  Application: CareConnect  Software Version: 1.0  Operating System: iOS 18 |
| **Priority:** High |
| **Requirements validated:** REQ-5.8.1.3.1–5.8.1.3.5, REQ-5.8.2.3.1–5.8.2.3.6 | |
| **TEST EXECUTOR:** | |
| **TEST SCRIPT STEPS/RESULTS** | |

| **STEP** | **TEST STEP/INPUT** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/FAIL** |
| --- | --- | --- | --- | --- |
| **8.1: In-App Messaging - Send and Receive Text Message** | | | | |
| 1. | On User A’s device, open the conversation with User B (or start a new chat). | Conversation screen is displayed. |  |  |
| 2. | In the message input field, type a message (e.g. “Hello”) and tap Send. | Message appears immediately in User A’s chat window (with current timestamp). |  |  |
| 3. | On User B’s device, confirm receipt of the new message in the same conversation. | User B sees “Hello” appear in real-time with timestamp (per REQ-5.8.1.3.3–3.4). |  |  |
| **8.2: In-App Messaging - Conversation List Updates** | | | | |
| 1. | On User A’s device, send a new message to User B (as above). | New message is sent. |  |  |
| 2. | Navigate back to the inbox/chat list view. | A conversation entry for User B shows the latest message snippet and timestamp (confirm stored message, per REQ-5.8.1.3.4–3.5). |  |  |
| 3. | On User B’s device, close and reopen the chat. | The previously sent message still appears in the conversation (persisted storage). |  |  |
| **8.3 In-App Messaging - Message Persistence After Restart** | | | | |
| 1. | On User A’s device, send a message to User B. | Message is delivered to User B. |  |  |
| 2. | Close the app completely on both devices, then reopen and log in again. | Conversation history including the sent message is still present on both sides. |  |  |
| **8.4 Audio and Video Calling – Initiate and Connect Audio Call** | | | | |
| 1. | On User A’s device (e.g. patient), open User B’s (caregiver) profile or chat and tap the Audio Call icon. | An outgoing audio call is placed; User B’s device shows an incoming call alert. |  |  |
| 2. | On User B’s device, tap Accept on the call alert. | The audio call connects; both users can hear each other (per REQ-5.8.2.3.1). |  |  |
| 3. | Verify that an End Call button is visible on both devices. | Users can end the call by tapping it. |  |  |
| **8.5 Audio and Video Calling – Initiate and Connect Video Call** | | | | |
| 1. | On User A’s device, open User B’s profile/chat and tap the Video Call icon. | An outgoing video call is placed; User B’s device shows an incoming video call alert. |  |  |
| 2. | On User B’s device, tap Accept on the call alert. | The video call connects; both users see each other’s video feed (per REQ-5.8.2.3.1). |  |  |
| 3. | During the call, on one device tap the Mute icon and the Disable Video icon. | Audio is muted on that device, and the video stream stops; UI indicates the mute/video-off state (per REQ-5.8.2.3.5). |  |  |
| **8.6 Audio and Video Calling – Reject Incoming Call** | | | | |
| 1. | On User A’s device, initiate an audio or video call to User B (as in Test Case 1 or 2). | User B’s device shows an incoming call alert. |  |  |
| 2. | On User B’s device, tap Reject (decline the call). | The call ends immediately. User A sees a “call ended” message or hears a disconnect tone (per REQ-5.8.2.3.3). |  |  |
| 3. | Verify that neither device remains in a call state. | Both users return to their previous screen; no active call is present. |  |  |
| **8.7 Audio and Video Calling – End Active Call** | | | | |
| 1. | Following Test Case 1 or 2, with an active call in progress, tap the End/Hang Up button. | The call disconnects (per REQ-5.8.2.3.3); both users return to the app’s previous screen (e.g. conversation or dashboard). |  |  |
| 2. | Attempt to resume the call without initiating a new one. | The call cannot be resumed; users must place a new call if needed. |  |  |
| **8.8 - Emergency SOS – Patient sees and taps Emergency SOS button (OpenAI, 2025)** | | | | |
| 1. | Patient logs in and views the dashboard. | SOS button is visible and tappable. |  |  |
| **8.9 - Emergency SOS – Patient sends an Emergency SOS alert (OpenAI, 2025)** | | | | |
| 1. | Following success of test case 8.8, patient taps SOS button | Emergency alert confirmation notification is visible. |  |  |
| 2. | Patient selects “Yes” to Application request for confirmation of SOS alert with location and audio. | Emergency SOS notification is sent to scheduled caregiver. |  |  |
| 3. | Patient starts audio recording. | Application saves audio. |  |  |
| **8.10 - Emergency SOS – Patient cancels Emergency SOS alert (OpenAI, 2025)** | | | | |
| 1. | Following success of test case 8.8, patient taps Emergency SOS button. | Emergency alert confirmation notification is visible. |  |  |
| 2. | Patient taps “Cancel” button on application confirmation for SOS. | Emergency SOS cancel notification appears. No notification is sent. |  |  |
| 3. | Patient taps “Close” button. | Patient is returned to dashboard. |  |  |
| **8.11 - Emergency SOS – Caregiver receives Emergency SOS alert (OpenAI, 2025)** | | | | |
| 1. | Following success of test case 8.9, a patient sends an emergency SOS alert. | Caregiver receives Emergency SOS alert pop up on their screen. |  |  |
| **8.12 - Emergency SOS – Caregiver acknowledges Emergency SOS alert within 2 minutes (OpenAI, 2025)** | | | | |
| 1. | Following success of test case 8.11, Caregiver taps “Acknowledge” button within 2 minutes. | Application directs user to Emergency SOS alert details screen. |  |  |
| **8.13 - Emergency SOS – Live location and audio are shared with caregiver (OpenAI, 2025)** | | | | |
| 1. | Following success of test case 8.12, application shows Emergency SOS alert details screen. | Patient’s live location and audio play options are present on details screen. |  |  |
| 2. | Caregiver taps “Play” icon. | Application plays audio sent from patient. |  |  |
| **8.14 - Emergency SOS – Caregiver ignores Emergency SOS Alert (OpenAI, 2025)** | | | | |
| 1. | Following success of test case 8.10, the caregiver ignores the emergency SOS alert for over 2 minutes. | Patient is notified that caregiver has not acknowledged Emergency SOS alert. |  |  |
| **8.15 Emergency SOS – Backup caregiver is notified (OpenAI, 2025)** | | | | |
| 1. | Following success of test case 8.14, patient selects “Notify Backup Caregiver” button. | Backup caregiver receives Emergency SOS alert. |  |  |
| **8.16 - Virtual Check-In – Caregiver navigate to Check-In Creation Page (OpenAI, 2025)** | | | | |
| 1. | Caregiver taps “Create Check-In" button. | Caregiver is directed to the “Create Check-In Questionnaire” screen. |  |  |
| **8.17 - Virtual Check-In – Caregiver sets-up virtual check-in process (OpenAI, 2025)** | | | | |
| 1. | Caregiver inputs questions and selects cadence (e.g., hourly, daily) | Questions and frequency saved to session; “Next” is enabled. |  |  |
| 2. | Caregiver taps “Next” to review, then taps “Confirm”. | Check-in is finalized and stored; scheduled alerts are queued. |  |  |
| **8.18 - Virtual Check-In - Cancel Check-In Creation (OpenAI, 2025)** | | | | |
| 1. | Caregiver taps “Cancel” before submitting check-in setup. | Inputs discarded; caregiver returned to dashboard |  |  |
| **8.19 - Virtual Check-In - Patient Receives Scheduled Notification (OpenAI, 2025)** | | | | |
| 1. | Check-in time arrives for patient to be notified. | Notification appears on patient’s device to begin check-in. |  |  |
| **8.20 - Virtual Check-In - Patient Opens and Fills Check-In (OpenAI, 2025).** | | | | |
| 1. | Patient taps notification, answers questions, taps “Submit”. | Confirmation prompt appears; data temporarily stored in S3 table. |  |  |
| **8.21 - Virtual Check-In - Patient Cancels Submission (OpenAI, 2025).** | | | | |
| 1. | Patient taps “Cancel” on confirmation prompt. | No data sent; patient is returned to dashboard screen. |  |  |
| **8.22 - Virtual Check-In - Caregiver Receives Notification of Completed Check-In (OpenAI, 2025)** | | | | |
| 1. | Patient completes and submits check-in. | Caregiver receives notification with access to submitted answers. |  |  |
| **8.23 - Virtual Check-In - Caregiver Receives Alert for Missed Check-In (OpenAI, 2025)** | | | | |
| 1. | Patient does not complete check-in within required time. | Caregiver receives alert indicating the check-in was missed. |  |  |
| **8.24 - Telehealth Bridge – Jitsi Integration Established - Verify that the application can successfully authenticate and communicate with the Jitsi Meet API (HIPAA-compliant) (OpenAI, 2025)** | | | | |
| 1. | Valid API credentials configured in the system. | Jitsi meeting created; Jitsi Meet API returns valid meeting link and metadata. |  |  |
| **8.25 - Telehealth Bridge - Caregiver Adds Telehealth Event to Calendar - Confirm that a logged-in caregiver can create a telehealth event from the calendar (OpenAI, 2025)** | | | | |
| 1. | Caregiver selects date/time and confirms creation. | Event appears in calendar with placeholder or actual Jitsi link. |  |  |
| **8.26 - Telehealth Bridge - API Call Made to Jitsi- Ensure the application sends a correctly formatted API request to Jitsi Meet upon event creation (OpenAI, 2025)** | | | | |
| 1. | Event creation action triggered. | API call sent, response code 200 OK with meeting link. |  |  |
| **8.27 - Telehealth Bridge - Meeting Link Added to Calendar - Verify the Jitsimeeting link is attached to the calendar event upon API success (OpenAI, 2025)** | | | | |
| 1. | Jitsi returns a meeting URL. | Event displays join link for both caregiver and patient. |  |  |
| **8.28 - Telehealth Bridge – Patient/Caregiver Prepares Questions for Session (OpenAI, 2025)** | | | | |
| 1. | Patient Prepares Questions for Session with text input. | Questions captured in text response and ready to save. |  |  |
| 2. | Patient Prepares Questions for Session with audio input. | Questions captured in audio response and ready to save. |  |  |
| 3. | Questions saved with tap of “Submit” button. | Questions are saved in local storage. |  |  |
| **8.29 - Telehealth Bridge – Caregiver and Patient can access previously saved notes (OpenAI, 2025)** | | | | |
| 1. | Caregiver and/or patient views telehealth details screen. | Questions/notes are retrieved from local storage and presented in the details screen. |  |  |
| **8.30 - Telehealth Bridge - Join Meeting via “Join” Button (OpenAI, 2025)** | | | | |
| 1. | Patient or caregiver taps the “Join Meeting” button on the calendar. | Jitsi meetings open using provided meeting link. |  |  |
| **8.31 - Telehealth Bridge - End Session Redirect (OpenAI, 2025)** | | | | |
| 1. | End call button or Jitsi session termination. | App redirects to caregiver or patient dashboard screen. |  |  |

### 6.1.8 Device and Third-Party Integration Test Suite

**Scope**: Wearable devices, smart home integration, and external service connectivity  
**Total Test Cases**: 7  
**Requirements Coverage**: REQ-5.6.1.2.1, REQ-5.6.1.2.2, REQ-5.6.1.2.3, REQ-5.6.1.2.4,REQ**-5.6.1.2.5,** REQ-5.9.1.2.6, REQ-5.9.1.2.7, REQ-5.9.1.2.8, REQ-5.9.2.2.1, REQ-5.9.2.2.2, REQ-5.9.2.2.3, REQ-5.9.2.2.4, REQ-5.9.2.2.5, REQ-5.9.3.2.6, REQ-5.9.3.2.7, REQ-5.9.3.2.1, REQ-5.9.3.2.2, REQ-5.9.3.2.3, REQ-5.9.3.2.4, REQ-5.9.3.2.5, REQ-5.9.3.2.6, REQ-5.9.3.2.7

|  |  |
| --- | --- |
| **Prerequisites for this test:**  Valid Fitbit account and device  Google Nest camera configured  Amazon Alexa device linked  Stable internet connectivity  OAuth credentials configured | **Software Versions:**  Application: CareConnect  Software Version: 1.0  Operating System: iOS/Android |
| **Priority:** High |
| **Requirements validated:** REQ-5.6.1.2.1, REQ-5.6.1.2.2, REQ-5.6.1.2.3, REQ-5.6.1.2.4,REQ**-5.6.1.2.5,** REQ-5.9.1.2.6, REQ-5.9.1.2.7, REQ-5.9.1.2.8, REQ-5.9.2.2.1, REQ-5.9.2.2.2, REQ-5.9.2.2.3, REQ-5.9.2.2.4, REQ-5.9.2.2.5, REQ-5.9.3.2.6, REQ-5.9.3.2.7, REQ-5.9.3.2.1, REQ-5.9.3.2.2, REQ-5.9.3.2.3, REQ-5.9.3.2.4, REQ-5.9.3.2.5, REQ-5.9.3.2.6, REQ-5.9.3.2.7 | |
| **TEST EXECUTOR:** | |
| **TEST SCRIPT STEPS/RESULTS** | |

| **STEP** | **TEST STEP/INPUT** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/FAIL** |
| --- | --- | --- | --- | --- |
| **1.1: Fitbit Integration** | | | | |
| 1. | Access wearable integration settings from user profile | Wearable integration page displays with "Connect Fitbit" option visible |  |  |
| 2. | Select "Connect Fitbit" option | App redirects to Fitbit OAuth authorization page |  |  |
| 3. | Log in with valid Fitbit credentials | Fitbit login successful, permissions page appears |  |  |
| 4. | Grant required permissions (heart rate, steps, sleep data) | All permission toggles are visible and can be selected |  |  |
| 5. | Click "Allow" to complete authorization | Redirected back to CareConnect with success message |  |  |
| 6. | Verify connection status in settings | Fitbit shows as "Connected" with last sync timestamp |  |  |
| **1.2: Apple Health Integration** | | | | |
| 1. | On iOS device, access wearable settings | "Connect Apple Health" option visible |  |  |
| 2. | Tap "Connect Apple Health" | iOS permission dialog appears for HealthKit |  |  |
| 3. | Toggle ON permissions (heart rate, steps, sleep data) | Permissions granted |  |  |
| 4. | Tap "Allow" to confirm | Returns to app with "Apple Health Connected" status |  |  |
| 5. | Verify data appears in dashboard | Apple Health shows as "Connected" with last sync timestamp |  |  |
| **1.3: Google Health Connect Integration** | | | | |
| 1. | On Android device, access wearable settings | "Connect Health Connect" option visible |  |  |
| 2. | Tap "Connect Health Connect" | Android Health Connect permission screen appears |  |  |
| 3. | Toggle ON permissions (heart rate, steps, sleep data) | Permissions toggles work, "Allow" button enabled |  |  |
| 4. | Complete authorization | Google Health shows as "Connected" with last sync timestamp |  |  |
| **1.4: Google Nest Camera Integration** | | | | |
| 1. | Navigate to Home Monitoring settings | Smart camera options displayed |  |  |
| 2. | Select "Connect Google Nest" | Redirects to Google account authorization |  |  |
| 3. | Sign in with Google account linked to Nest | Shows Nest devices associated with account |  |  |
| 4. | Select camera(s) to connect | Selected cameras show checkmarks |  |  |
| 5. | Grant CareConnect permissions | Success message, returns to app |  |  |
| 6. | View live camera feed | Live stream loads |  |  |
| 7. | Test motion detection alerts | Motion triggers notification |  |  |
| **1.5: Amazon Alexa Integration //TODO** | | | | |
| 1. | Launch the app; tap Register. | Registration screen appears. |  |  |
| 2. | Enter an invalid email (e.g. “user@domain”, missing “.com”) and a password; tap Submit. | The app blocks submission and displays an error about invalid email format (per FR-3.1.2). |  |  |
| 3. | Correct the email and submit again. | Registration then succeeds as in Test Case 1 (account is created and user proceeds). |  |  |
| **1.6: Medication Management** | | | | |
| 1. | Navigate to Medications section | Add Medication button visible |  |  |
| 2. | Tap "Scan NDC Barcode" | Camera opens with barcode scanner overlay |  |  |
| 3. | Scan test NDC code | |Barcode recognized, processing message appears |  |  |
| 4. | Wait for OpenFDA API response | Medication name auto-populates |  |  |
| 5. | Verify auto-filled fields | Manufacturer, strength, and form fields filled |  |  |
| 6. | Add dosage frequency | Custom fields accept input |  |  |
| 7. | Save medication | Medication added to list with reminder scheduled |  |  |

### 6.1.9 Billing and Subscription Test Suite

**Scope**: Complete billing and subscription management for CareConnect application  
**Total Test Cases**: 4  
**Requirements Coverage**: FR-B1-FR-B4

|  |  |
| --- | --- |
| **Prerequisites for this test:**   * Fresh app installation (no user logged in) * Internet connectivity * Test email/password credentials * Mobile device ready to receive SMS * Google account available for SSO * Stripe integration | **Software Versions:**  Application: CareConnect  Software Version: 1.0  Operating System: iOS 18 |
| **Priority:** High |
| **Requirements validated:** FR-B1-FR-B4 | |
| **TEST EXECUTOR:** | |
| **TEST SCRIPT STEPS/RESULTS** | |

| **STEP** | **TEST STEP/INPUT** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/FAIL** |
| --- | --- | --- | --- | --- |
| **1.1: Display Plan Catalog During Signup (OpenAI, 2025)** | | | | |
| 1. | Open the signup page. | Signup page loads. |  |  |
| 2. | View available subscription plans. | * Two plans are displayed: * Standard: $20/patient/month * Premium: $30/patient/month * Pricing reflects latest backend configuration. |  |  |
| **1.2 Activate Subscription on Successful Payment (OpenAI, 2025)** | | | | |
| 1. | Select a subscription plan. | Subscription plan is selected in UI |  |  |
| 2. | Enter valid credit card or PayPal info. | Information appears on screen as it is filled out. |  |  |
| 3. | Submit payment. | * Payment is processed via Stripe. * New account is created with ACTIVE status in billing\_subscriptions. * No trial period is shown or triggered. |  |  |
| **1.3 Retry Credit Card Payment and Fallback to PayPal (OpenAI, 2025)** | | | | |
| 1. | Simulate a credit card failure during billing. | Payment fails with credit card. |  |  |
| 2. | Observe retry behavior and fallback sequence. | * Retry occurs up to 3 times (1h → 4h → 24h). * If all retries fail, system attempts PayPal. * Each step is logged and a “payment retrying” notice is pushed. |  |  |
| **1.4 Secure Tokenization of Payment Details (OpenAI, 2025)** | | | | |
| 1. | Enter card details and tap "Pay". | Payment is processed successfully. |  |  |
| 2. | Inspect UI behavior and back-end logs. | * UI shows loading spinner (≤ 500 ms). * Stripe Elements tokenizes payment. * Only token ID is posted to the server. * Backend does not store raw card data. * Backend responds in < 2 seconds. |  |  |

### 6.1.10 User Password Reset & Recovery

**Scope**: Reset user password and help with account recovery  
**Total Test Cases**: 1  
**Requirements Coverage**: REQ-5.1.4.6.1 to REQ-5.1.4.6.4

|  |  |
| --- | --- |
| **Prerequisites for this test:**   * Fresh app installation (no user logged in) * Internet connectivity * Test email/password credentials * Mobile device ready to receive SMS * Google account available for SSO | **Software Versions:**  Application: CareConnect  Software Version: 1.0  Operating System: iOS 18 |
| **Priority:** High |
| **Requirements validated:** REQ-5.1.4.6.1 to REQ-5.1.4.6.4 | |
| **TEST EXECUTOR:** | |
| **TEST SCRIPT STEPS/RESULTS** | |

| **STEP** | **TEST STEP/INPUT** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/FAIL** |
| --- | --- | --- | --- | --- |
| **1.7: Password Reset & Account Recovery** | | | | |
| 1. | On login screen, click “Forgot Password/Account Recovery” | Navigates to account recovery page |  |  |
| 2. | Enter registered email and click “Next” | Proceeds to security question screen |  |  |
| 3. | Answer security questions correctly, click “Submit” | Navigates to DOB and address input page |  |  |
| 4. | Input correct date of birth and address | Navigates to password creation screen |  |  |
| 5. | Enter a **new password** (same as the previous password), confirm and click “Submit” | System displays error: "New password must be different from the old password" *(REQ-5.1.4.6.2)* |  |  |
| 6 | Enter a **new valid password**, confirm with **non-matching entry**, click “Submit” | System displays error: "Passwords do not match" *(REQ-5.1.4.6.3)* |  |  |
| 7. | Enter a valid and unique new password, confirm correctly, click “Submit” | The display screen will confirm a succesful password. |  |  |
| 8. | Check registered email inbox | Email received confirming password change *(REQ-5.1.4.6.1)* |  |  |
| 9. | Log in using the new password | Login successful |  |  |
| 10. | Attempt to change password again via the same process | Password change allowed again without limits *(REQ-5.1.4.6.4)* |  |  |

### Patient Dashboard

| **Test Case ID** | **Description** | **Input** | **Expected Output** | **Requirement ID** |
| --- | --- | --- | --- | --- |
| TC-PD-01 | Display Daily care tasks | User login | Dashboard loads with today’s task list | REQ-5.2.4.1, REQ-UI-1 |
| TC-PD-02 | Mark task as completed | Checkbox tap | Task visually marked, progress bar updates, backend reflects completion | REQ-5.2.4.2 |
| TC-PD-03 | Record mood using face icons | Tap mood icon | Mood logged and UI confirmation shown | REQ-5.2.4.3, IUF-WS-1 |
| TC-PD-04 | Contact caregiver via message | Tap message icon | Chat screen opens | EUF-WS-1 |
| TC-PD-05 | Trigger emergency SOS alert | Tap SOS button | Alert sent to caregiver; toast confirmation appears | REQ-5.2.4.5 |
| TC-PD-06 | Archive Caregiver | Tap moon icon | Caregiver profile is removed from dashboard | REQ-5.4.3.3.4 |
| TC-PD-07 | Call Caregiver | Tap phone icon | Caregiver is called and screen displays profile picture and name | REQ -5.4.4.2 |
| TC-PD-08 | Details | Tap person icon | Details about the Caregiver is displayed | REQ-5.4.4.3.1 |
| TC-PD-09 | Edit/Remove | Tap pencil/trashcan | A screen will display allowing editing and deleting | REQ-5.4.4.3.1 |

### Gamification Module Test Suite

| **Test Case ID** | **Description** | **Input** | **Expected Output** | **Requirement ID** |
| --- | --- | --- | --- | --- |
| TC-GF-01 | Award points upon task completion | Complete a daily task | Points awarded and visible on profile | REQ-5.10.1.1 |
| TC-GF-02 | View total accumulated points | Open profile | Total points displayed in dashboard | REQ-5.10.1.2 |
| TC-GF-03 | Unlock badge after reaching milestone | Complete enough tasks | Badge shown in achievements panel | REQ-5.10.1.3 |
| TC-GF-04 | Display celebratory message after earning points | Task completed | UI displays animation or congratulatory message | UI-UX Requirement |

### Social Networking Test Suite

| **Test Case ID** | **Description** | **Input** | **Expected Output** | **Requirement ID** |
| --- | --- | --- | --- | --- |
| TC-SN-01 | View community forum posts | Navigate to social tab | List of posts appears with username, timestamp | REQ-5.11.1.1 |
| TC-SN-02 | Submitting a new comment | Enter text and post | Comment visible under corresponding topic | REQ-5.11.1.2 |
| TC-SN-03 | React to another user’s post | Tap like/emoji | Reaction count increases, visual update shown | REQ-5.11.1.3 |
| TC-SN-04 | View caregiver activity feed | Open caregiver profile | Timeline-style updates appear (e.g., “Task acknowledged”) | REQ-5.11.1.4 |
| TC-SN-05 | Privacy check for patient posts | Post with restricted visibility | Only assigned caregiver sees post | REQ-5.11.1.5 |

## 6.2 Traceability Matrix

The test cases and functional requirements can be mapped with the following traceability matrix to keep items organized.

**Table 8**

*Traceability Matrix for the CareConnect project.*

| **Requirement ID** | **Requirement Description** | **Test Case IDs** | **Verification Method** | **Requirement Source** |
| --- | --- | --- | --- | --- |
| REQ-3.1.1 - REQ-3.2.8 | User Registration and Authentication | Authentication and Onboarding Test Suite | Functional Test |  |
| REQ-5.4.1-REQ-5.4.5 | Dashboard and Navigation | Dashboard and Navigation Test Suite | Functional Test |  |
| REQ-5.3.1.1- REQ-5.3.5.3 | User and Role Management | User and Role Management Test Suite | Functional Test |  |
| REQ-6.1.1 to REQ-6.1.5 | Scheduling and Notifications Test Suite | Scheduling and Notifications Test Suite | Functional Test |  |
| REQ-5.6.1.1 - 5.6.14.8 | Health Data Tracking Test Suite | Health Data Tracking Test Suite | Functional Test |  |
| REQ-5.7.1.1 - REQ-5.7.2.8 | AI Integration Test Suite | AI Integration Test Suite | Functional & Integration Test |  |
| REQ-6.1.1 to REQ-6.2.6 | Communcation and Media Test Suite | Communcation and Media Test Suite | Functional & Integration Test |  |
| REQ-5.6.1.2.1 to REQ-5.9.3.2.7 | Device and Third-Party Integration Test Suite | Device and Third-Party Integration Test Suite | Functional Test |  |
| FR-B1-FR-B4 | Billing and Subscription Test Suite | Billing and Subscription Test Suite | Functional Test |  |
| REQ-5.1.4.6.1 to REQ-5.1.4.6.4 | Password Reset & Recovery | Password Reset & Recovery | Functional Test |  |

# Entry and Exit Criteria

The following tables display the entry and exit criteria we will require to determine a working CareConnect product that may be delivered to the clients.

### Entry Criteria

**Table 9**

*Entry Criteria to begin testing CareConnect*

| **Criteria** | **Detail** |
| --- | --- |
| Requirements Approved | All requirements and acceptance criteria are documented and approved, ensuring clarity on expected behavior. |
| Backend Service Ready | The backend service is implemented and accessible. |
| UI/UX Implementation | The UI is completed |
| Test Plan and Test Cases Completed | Test Plan and test suites are documented, reviewed, and traceable to requirements. |
| Test Data and Credentials Prepared | Test accounts, datasets, API tokens, and mock integrations are ready and validated. |
| Defect Tracking System Active | GitHub Issues is active with predefined issue types, severity levels, and workflows. |

### Exit Criteria

**Table 10**

*Exit Criteria to sign-off testing and deliver CareConnect*

| **Criteria** | **Detail** |
| --- | --- |
| All Test Cases Executed | 100% of functional, integration, regression, and performance test cases are executed at least once |
| No Critical Defects | All critical or high-severity defects in the software are fixed and verified. |
| Test Pass Rate ≥ 95% | At least 95% of executed test cases pass; failures have acceptable mitigations or are low priority |
| Regression Testing Completed | All previously working features re-validated; no new breakages introduced by recent changes |
| Performance, Security, and Usability Validated | Benchmarks met for load, HIPAA compliance, across supported platforms and roles. |
| Stakeholder Sign-off Acquired | QA Lead, Project Manager, and Product Owner have formally approved release readiness. |

# Risk and Mitigation

This section outlines our structured approach to identifying, prioritizing, and managing potential risks that could impact the CareConnect testing process. Using a simple yet effective 3×3 risk matrix and a detailed risk register, we assess the likelihood and potential impact of each risk—ranging from data privacy concerns with HIPAA compliance to video call reliability and backend service availability. By documenting each risk and its recommended mitigation actions, we ensure transparency, accountability, and proactive planning. This framework helps the QA team focus on protecting core features—such as secure AI Ask responses, stable patient symptom tracking, and embedded video conferencing—while maintaining test schedule and resource efficiency. The approach supports weekly review cycles, enabling adjustments as CareConnect evolves.

## 8.1 Risk Matrix

The risk matrix provides a structured overview of potential project risks by plotting each risk’s likelihood against its potential impact. It is included in the test plan to ensure that testing and mitigation efforts focus on the most critical areas. In this matrix, risks related to CareConnect’s unique requirements are highlighted — for example, data privacy and HIPAA compliance issues, the reliability of the AI Ask feature, video consultation quality, symptom tracking accuracy, and backend integration stability. The risk matrix summarizes these areas so the QA team can focus tests on the highest-impact scenarios. Key risk areas for CareConnect include:

* HIPAA compliance breaches (e.g., data encryption failures or unauthorized access)
* AI Ask feature errors (e.g., inaccurate or unsafe medical guidance)
* Video conferencing issues (e.g., connectivity loss or poor call quality)
* Symptom tracking data problems (e.g., lost or inconsistent patient entries)
* Backend integration failures (e.g., API Gateway outages or container scaling limits)

The risk matrix visualizes these concerns so that stakeholders and the QA team can quickly identify and prioritize the issues that require the most immediate attention.

| **Likelihood \ Impact** | **Low** | **Medium** | **High** |
| --- | --- | --- | --- |
| **Low** | Low | Low | Medium |
| **Medium** | Low | Medium | High |
| **High** | Medium | High | High |

## 8.2 Risk Register

The risk register is a detailed log of all identified project risks, including their descriptions, likelihood, impact, and mitigation strategies. It is included in the test plan to systematically document and track potential problems that could affect CareConnect. Each entry ensures that issues like HIPAA compliance failures, AI functionality problems, or video call disruptions are not overlooked and have corresponding action plans. Maintaining this register helps the team monitor the status of each risk and verify that appropriate controls are in place throughout testing. Each register entry typically includes:

* A unique ID and concise risk description
* An assessment of likelihood (probability) and impact (severity)
* Proposed mitigation or contingency actions
* Any relevant status notes or current controls

**Table 11**

*Risk Matrix used for the CareConnect project.*

| **ID** | **Risk Description** | **Likelihood** | **Impact** | **Mitigation Action** |
| --- | --- | --- | --- | --- |
| R1 | Unencrypted patient data storage leading to HIPAA compliance risk | Medium | High | Enforce encryption in transit and at rest; conduct audits |
| R2 | Integration endpoint failure between front-end and backend | Medium | High | Implement retry logic and monitor API health |
| R3 | Symptom tracking data loss due to sync errors | Medium | Medium | Use transactional APIs and verify data consistency |
| R4 | Video call connectivity loss on certain networks | High | Medium | Perform network resilience testing; optimize video settings |
| R5 | AI Ask feature returns inaccurate or unsafe medical guidance | Medium | High | Validate AI outputs with medical experts; implement disclaimers |
| R6 | AWS backend fails to scale (Fargate/API Gateway) causing downtime | Medium | High | Conduct load testing; configure auto-scaling and rate limits |
| R7 | Embedded video conferencing fails on some devices or browsers | Medium | Medium | Conduct cross-device/browser testing; provide fallback options |

# Change Management Procedures

Any modifications to the test plan, whether they are scope updates, schedule shifts, or resource changes, must follow a controlled change process. Our process involves three clear stages:

### Initiation

Any team member who identifies a needed change—such as a new testing requirement or delay—submits a standardized Change Request Form. This form includes the purpose of the change, expected impact, required resources, and timeline. Templates ensure that each change is described consistently, preventing ad hoc requests

### Review

The proposed change is reviewed by a small Change Control Board (CCB), composed of the Course Instructor/Client and clients, Team Lead and Group Project Manager, both Technical Leads/Architects, and the remaining CareConnect team members. They assess risks, alignment with goals, impact on the testing schedule, and resource availability. Low-impact changes may be fast-tracked, while higher-impact proposals require more detailed planning.

**Table 12**

*Change Control Board Members*

| **CCB Member** | **Role** | **Responsibilities** |
| --- | --- | --- |
| Dr. Assadullah  Roy Gordon  Ashley Wane | Course Instructor/ Client  Client  Client | * Final approval or rejection of change * Determines change priority |
| Alyssa Marielle Harding  Alireza Minagar | Team Lead  Group Project Manager | * Determine if changes are viable for review by CCB. |
| Ashenafi Grbreegziabhere, Edwenson Raphael | Technical Leads/Architects | * Determine if changes are viable for review from technical standpoint. |
| CareConnect Team Members | BA, Developers, Testers | * Provide analysis of change requests to produce an estimated level of effort and potential impacts to other requirements |

*Note*. Change Control Board Members Table created with reference from previous 2023 SWEN 670 AlphaSoft project (AlphaSoft, 2023).

### Authorization and Implementation

Once approved, changes are documented in the test plan baseline, including who approved them and when. The updated plan is then communicated to all stakeholders. Approved changes are implemented in development and testing environments using version control, and validation is performed through a brief smoke test or regression cycle. For major rollouts, a partial or full test pass may be scheduled. All changes and their outcomes are logged for auditability and future review.

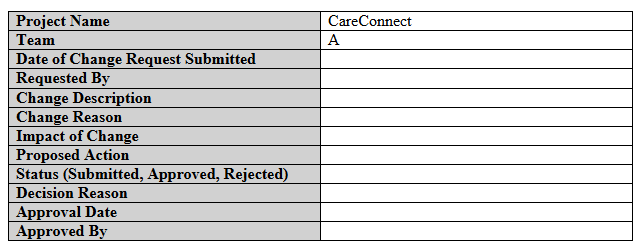
By following this structured yet adaptable change workflow, including initiation, review, and authorization, we will ensure the CareConnect testing remains aligned with project goals while minimizing disruption.

### CareConnect Change Request Form

The CareConnect Change Request Form can be located in the Teams section of the Joint Collab Microsoft Teams channel linked here: [CareConnect Change Request Form Template.docx.](https://umgcdev361.sharepoint.com/:w:/r/sites/SWEN670Summer2025/Shared%20Documents/Joint%20Collab%20(Care%20Connect)/CareConnect%20Change%20Request%20Form%20Template.docx?d=w488c50dd08794e6daccf9c30b13c967f&csf=1&web=1&e=9Fdn7f) A figure has been provided below to illustrate what the template for the Change Request Form looks like for reference.

**Figure 1**

*CareConnect Change Request Form Template*

  
*Note*. CareConnect Change Request Form Template created with reference from previous 2023 SWEN 670 AlphaSoft project (AlphaSoft, 2023).

# Defect Management

## Overview

Defect management in the CareConnect project ensures that bugs, errors, and issues identified during testing are properly tracked, prioritized, resolved, and documented. Effective defect management allows the development and QA teams to maintain a high-quality product, minimize regressions, and ensure timely resolution of any issues that impact functionality, performance, or user experience.

## Defect Tracking Process

All defects will be logged, tracked, and managed using the GitHub Issues feature, integrated with our code repository. This centralizes defect reporting, status updates, and resolution tracking within the same platform used for source control and CI/CD pipelines. Every issue will be tagged with a unique identifier, priority level, severity level, assigned owner, and a clear description of the defect.

Each defect will follow a standardized lifecycle, as outlined below:

* Defect Identification – Discovered during manual or automated testing and reported by testers, developers, or other team members.
* Defect Logging – Logged into GitHub Issues with details such as summary, steps to reproduce, screenshots/logs, severity, priority, and environment details.
* Triaging and Assignment – Reviewed during QA meetings and assigned based on priority and module ownership.
* Resolution and Verification – Fixed by the developer, pushed for review, tested by QA, and closed if resolved.
* Reopening (if needed) – If the issue persists, it may be reopened and escalated as needed.

## Severity and Priority Levels

**Table 12**

*Severity Levels used for the CareConnect project.*

| **Severity** | **Definition** |
| --- | --- |
| **Critical** | Blocks system functionality; no workaround. |
| **High** | Major feature broken; affects user workflow. |
| **Medium** | Feature issue with a workaround available. |
| **Low** | Minor UI/UX issue or cosmetic error. |

**Table 13**

*Priority Levels used for the CareConnect project.*

| **Priority** | **Definition** |
| --- | --- |
| **P1** | Must fix immediately before release. |
| **P2** | Fix in the current sprint/iteration. |
| **P3** | Fix after major features are completed. |
| **P4** | Low-priority or backlog of items. |

## Defect Reporting Tools

• Primary Tool: GitHub Issues (linked to pull requests and commits)  
 • Notification System: GitHub @mentions and email alerts for ticket updates  
 • Reporting & Metrics: Defect status, count by severity, average time to close, and open bug aging tracked in weekly QA reports

## Metrics Tracked

To monitor and continuously improve our testing and quality assurance process, the following defect metrics will be reviewed:

## Total number of defects (per sprint)

## Number of open vs. closed issues

## Average time to resolution

## Defect leakage (bugs found in production that escaped testing)

## Reopened defect count

## Conclusion

The defect management process ensures that issues are handled with transparency, accountability, and consistency. By leveraging GitHub’s built-in issue tracking system and applying structured triage procedures, the CareConnect team can deliver a reliable and user-friendly application while minimizing technical debt and deployment risk.