**Project Plan Document**

CareConnect

University of Maryland Global Campus

SWEN 670 - Software Engineering Capstone

Dr. Mir Assadullah

July 26th, 2025

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

Table of Contents

[Revision History v](#_Toc205129108)

[1.Executive Summary 1](#_Toc205129109)

[1.1 Project Name 1](#_Toc205129110)

[1.2 Purpose 1](#_Toc205129111)

[1.3 Scope 1](#_Toc205129112)

[2. Objectives and Goals 1](#_Toc205129113)

[2.1 Goals 1](#_Toc205129114)

[2.2 Objectives 1](#_Toc205129115)

[3. Scope of Work 2](#_Toc205129116)

[3.2 Out-of-Scope 4](#_Toc205129117)

[4. Stakeholders 4](#_Toc205129118)

[4.1 Internal: 4](#_Toc205129119)

[4.2 External: 5](#_Toc205129120)

[5. Timeline & Milestone 6](#_Toc205129121)

[5.1 Timeline and Milestone Table 6](#_Toc205129122)

[5.2 Gantt Chart 7](#_Toc205129123)

[6. Technical Architecture 8](#_Toc205129124)

[6.1 Architectural Objectives 8](#_Toc205129125)

[6.2 Logical System Architecture 9](#_Toc205129126)

[6.3 High-Level System Architecture Diagram 11](#_Toc205129127)

[6.4 Architectural Rationale 13](#_Toc205129128)

[6.5 Runtime Flows 13](#_Toc205129129)

[6.6 Technology Stack Summary 14](#_Toc205129130)

[6.7 Deployment Topology Highlights 14](#_Toc205129131)

[6.8 Frontend 15](#_Toc205129132)

[6.9 Backend 15](#_Toc205129133)

[6.10 Database 17](#_Toc205129134)

[6.11 APIs 18](#_Toc205129135)

[6.11.1 Fitbit Web API 18](#_Toc205129136)

[6.11.2 Apple HealthKit API 18](#_Toc205129137)

[6.11.3 Google Health Connect API 19](#_Toc205129138)

[6.11.4 Google Smart Device Management (SDM) API 19](#_Toc205129139)

[6.11.5 Alexa Smart Home Skill API 19](#_Toc205129140)

[6.11.6 OpenFDA API 19](#_Toc205129141)

[6.12 AI Services 20](#_Toc205129142)

[6.12.1 Purpose and User Flow 20](#_Toc205129143)

[6.12.2 Technical Architecture 20](#_Toc205129144)

[6.12.3 Security and data Protection 20](#_Toc205129145)

[6.12.4 Compliance and Governance 21](#_Toc205129146)

[6.12.5 Summary and Future Enhancements 21](#_Toc205129147)

[6.13 Hosting 21](#_Toc205129148)

[6.13.1 Cloud Deployment Model 21](#_Toc205129149)

[6.13.2 Containerization and Orchestration 21](#_Toc205129150)

[6.13.3 Serverless-First & Scale-to-Zero 22](#_Toc205129151)

[6.13.4 Security and Reliability 22](#_Toc205129152)

[6.14 Security 22](#_Toc205129153)

[6.14.1 Data Protection 22](#_Toc205129154)

[6.14.2 Authentication and Access Control 23](#_Toc205129155)

[6.14.3 Secure APIs 23](#_Toc205129156)

[6.14.4 Infrastructure Hardening 23](#_Toc205129157)

[6.14.5 Monitoring and Logging 23](#_Toc205129158)

[6.14.6 Vulnerability Management 24](#_Toc205129159)

[6.14.7 Compliance Measures 24](#_Toc205129160)

[7. Project Deliverables 24](#_Toc205129161)

[8. Risk & Mitigation 25](#_Toc205129162)

[8.1 Risk Probability Matrix 25](#_Toc205129163)

[8.2 Risk Identification, Probability, and Mitigation 25](#_Toc205129164)

[9. Budget Estimate 27](#_Toc205129165)

[9.1 Labor Cost 27](#_Toc205129166)

[9.2 Materials Cost 27](#_Toc205129167)

[9.3 Infrastructure Cost 28](#_Toc205129168)

[9.3.1 AWS Core Services 31](#_Toc205129169)

[9.3.2 Third-party / One-time Fees 31](#_Toc205129170)

[9.3.3 Total Budget for MVP Phase (13 weeks ≈ 3 months) 31](#_Toc205129171)

[9.3.4 Cost-Control & Optimization Plan 31](#_Toc205129172)

[9.3.5 Alignment to Financial Policies 32](#_Toc205129173)

[9.3.6 Budget Summary for Project Plan 32](#_Toc205129174)

[10. Roles & Responsibilities 33](#_Toc205129175)

[10.1 Roles 33](#_Toc205129176)

[10.2 RACI Matrix 36](#_Toc205129177)

[11. Monitoring and Reporting 38](#_Toc205129178)

[11.1 Intra-Team Monitoring and Reporting 38](#_Toc205129179)

[11.2 Project Communications 38](#_Toc205129180)

[12. Assumptions 38](#_Toc205129181)

[13. Constraints 39](#_Toc205129182)

[14. Work Breakdown Structure (WBS) 39](#_Toc205129183)

[14.1 Structure 39](#_Toc205129184)

[15. Change Management Procedures 40](#_Toc205129185)

[15.1 Initiation 40](#_Toc205129186)

[15.2 Review 40](#_Toc205129187)

[15.3 Authorization and Implementation 41](#_Toc205129188)

[15.4 Change Request Form 41](#_Toc205129189)

[16. Appendices 42](#_Toc205129190)

[16.1 Glossary of Terms 42](#_Toc205129191)

[16.2 References 43](#_Toc205129192)

[15.5 UI/UX wireframe preview 42](#_Toc205129193)

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Team Name** | **Date** | **Reason for Changes** | **Version** |
| CareConnect Team | 05/31/2025 | Initial document submission. | 1.0 |
| CareConnectTeam | 06/14/2025 | Revisions based on 1.0 feedback | 2.0 |
| CareConnect Team | 07/26/2025 | Revisions based on 2.0 feedback | 3.0 |

# **1.Executive Summary**

## Project Name

The name of this project is CareConnect**.**

## Purpose

The purpose of this project is to develop a viable mobile application that provides a functional and user-friendly interface that connects caregivers and patients on a single platform.

The purpose of the CareConnect project is to design and develop a robust, intuitive mobile application that seamlessly connects caregivers and patients through a unified digital platform. This application aims to enhance the quality of care by enabling caregivers to efficiently monitor, organize, and manage the day-to-day health-related activities of their patients.

## Scope

The CareConnect platform empowers caregivers to deliver personalized, high-quality support through features such as patient profile management, task scheduling, and real-time health tracking. At the same time, it promotes patient autonomy by allowing individuals to actively engage in their own care—tracking on symptoms, recording meals, managing notifications, and initiating emergency alerts when needed. Through this bidirectional functionality, CareConnect fosters a collaborative care environment that improves outcomes, ensures continuity, and strengthens the patient-caregiver relationship.

# **2. Objectives and Goals**

## 2.1 Goals

The goal of the project is to build an intuitive interface where users can easily understand the application and effortlessly navigate through it.

## 2.2 Objectives

The objective of the project is outlined below:

* Provide an integrated platform where patients and caregivers can connect, coordinate, and communicate.
* Provide all the necessary tools for caregivers to manage, track, and support their care recipients.
* Provie all the necessary tools for the patient to manage their daily health tasks and activate emergency alerts at the time of need.

# **3. Scope of Work**

The scope of work for this project is listed below in In-Scope and Out of Scope:

**3.1 In-Scope***3.1.1. User Management and Access*

* The application shall allow caregivers to register, log in, and create a profile using email and password.
* The application shall allow caregivers to manage patient profiles.

*3.1.2. Communication Integration*

* The system shall support in-app messaging, audio/video calling, and virtual check-in rounds.

*3.1.3. Tasking & Medication Management*

* The application shall allow caregivers to assign daily tasks to patients (such as medication and meal reminders).
* The application shall allow patients to mark the task complete.

*3.1.4. Health Data Logging and Tracking*

* The application shall allow caregivers and patients to manually record health data (e.g., vital signs) and generate reports for review.

*3.1.5. Note & Documentation Application*

* The application shall allow caregivers to share notes related to their care recipient(patient).

*3.1.6. Emergency Assistance*

* The application shall allow patients to activate an emergency signal, notifying caregivers and SOS during urgent situations.

*3.1.7. Gamification*

* The application shall have Badges, rewards, motivation messages, and compliance leaderboards.
* These features will enhance user engagement by recognizing consistent activity and promoting healthy behaviors through in-app visuals.

*3.1.8. Billing & Subscription Management*

* The application shall support subscription-based billing for caregivers, including the ability to define pricing tiers (e.g., $20/patient/month), activate subscriptions upon user onboarding or patient linking, and securely collect payment via credit card or PayPal using Stripe integration.

*3.1.9. Scheduling & Notifications*

* The platform shall allow caregivers to create both template-based and custom care tasks and deliver alerts and reminders via push notifications, email, or SMS based on user preferences and priority.

*3.1.10. Analytics & Reports*

* The system shall generate real-time dashboard metrics and shall allow export of health and care data in CSV or PDF format for external reporting or consultation purposes.

*3.1.11. AI Integration*

The system shall include AI-powered features such as:

* Ask AI: An assistant that answers health-related queries using the patient’s records.
* Mood Detection: Real-time emotion analysis during video calls using on-device facial and voice recognition.
* These features shall include disclaimers and links to the original data sources

*3.1.12. Social Networking*

* The application shall support patient-caregiver communication through private messaging and a basic acitivy feed for posting care-related updates.

Users may form group discussions for caregiving communities or condition specific support groups.

*3.1.13. Multilingual Support*

* The System shall include multilingual capabilities to support users from diverse linguistic backgrounds.
* The System shall enable users to switch the language in the app ensuring the accurate appropriate translation.

## 

## 3.2 Out-of-Scope

*3.2.1. Clinical Diagnosis*

* The system shall not offer medical diagnosis or prescribe treatments. AI features are strictly informational with required disclaimers

*3.2.2. Accessibility Enhancements*

* The application shall not include features such as voice control, screen reader compatibility, or high-contrast mode.

*3.2.3. Advance Home Automation*

* Only essential triggers (e.g., motion alerts) are in scope. Full smart home automation routines (e.g., lighting, climate control) are excluded.

*3.2.4. Custom Medical Device Development*

* The project shall not involve creating new wearables or sensors. Integration is limited to existing APIs and platforms.

*3.2.5. Public Social Media Integration*

* The application shall not support public social media integration such as Facebook or Twitter cross-posting

*3.2.6. Community Forums*

* The application shall not include open community forums or large-scale public networking features.

# **4. Stakeholders**

## 4.1 Internal:

The Internal Stakeholders involved in this project are listed below. The details about the roles and responsibilities of the internal stakeholders are outlined in Section 10 of this document.

* Client
* Project Mentor
* MS Teams & GitHub Mentor
* Class Project Manager
* Team lead
* Technical Lead/Architect
* Software Developers
* Lead Business Analyst
* Business Analysts
* Lead UI/UX Designer
* UI/UX Designers
* Lead Tester
* Testers

## 4.2 External:

The following are the key external stakeholders for the CareConnect project:

* **Patients**  
   End users of the application who will manage their health tasks, receive reminders, and request emergency help via the platform.
* **Caregivers**  
   Individuals, whether professional or family-based, will utilize the platform to manage patient schedules, tasks, and wellness data.
* **Health Professionals**  
   Licensed providers such as nurses or case managers who may use CareConnect for remote monitoring and coordination of care tasks.
* **Family Members**  
   Extended family who supports the care of patients and may be secondary users of the system to stay informed or provide supplemental care.
* **Clients/Mentors**  
   Advisors providing project development feedback and insight into real-world healthcare technology integration.
* **University Faculty (Professor)**  
   Faculty overseeing and evaluating the capstone project deliverables and adherence to software engineering best practices.
* **Platform Providers**  
   External service providers including Apple App Store, Google Play Store, and third-party integration platforms such as Stripe or Firebase host, process, or extend the functionality of the CareConnect app.

**5. Timeline & Milestone**

## 5.1 Timeline and Milestone Table

**Table 1**

*Milestone and Timeline Table*

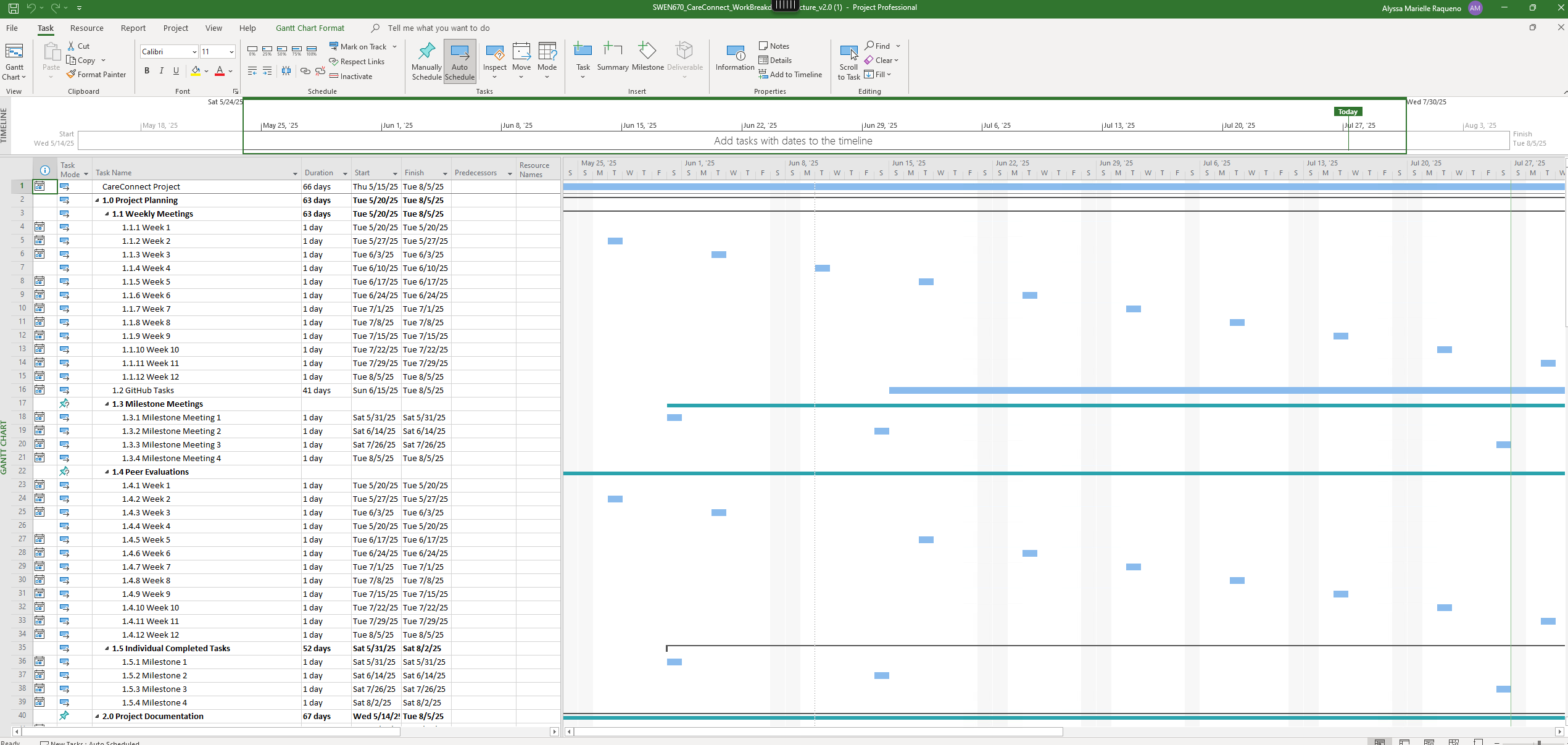
| **Milestone** | **Deliverables Due** | **Due Date** |
| --- | --- | --- |
| Milestone 1 | * Project Plan * Software Requirements Specification | May 31st, 2025 |
| Milestone 2 | * Technical Design Document * Software Test Plan | June 14th, 2025 |
| Checkpoint | * PowerPoint of Progress | July 5th, 2025 |
| Milestone 3 | * Programmer Guide * Deployment and Operations Guide * Software Test Plan | July 26th, 2025 |
| Milestone 4 | * User Guide * Test Report | August 5th, 2025 |

*Note.* This table outlines the major milestones and deliverable due dates for the CareConnect project.

## 5.2 Gantt Chart

**Figure 1**

*Gantt Chart of Milestones for CareConnect Project*



*Note.* This table outlines the major milestones and deliverable due dates for the CareConnect project.

This Gannt chart offers a clear, visual overview of the entire project timeline. It helps identify what is currently in progress, what is coming next, and what has been completed. It complements our written schedule by making the plan easier to understand at a glance, and it will also serve as a living reference for future updates and progress tracking.

An external Microsoft Project file has been included as a supplement of the overall documents for a more detailed view of the CareConnect project Gantt chart.

# **6. Technical Architecture**

The purpose of this section is to translate the functional and non-functional requirements defined in the SRS into an implementable, cloud-native architecture. It explains how CareConnect’s caregiver/patient features, especially Billing & Subscription Management and Analytics & Reporting—will be realized on AWS using Flutter, Spring Boot micro-services, and managed platform services. The accompanying deployment and system architecture diagrams visualize component boundaries, data flow, and runtime hosting topology.

## 6.1 Architectural Objectives

**Table 2**

*Architectural Objectives Table*

| **Objective** | **Rationale** |
| --- | --- |
| **Scalability** | Must support adding thousands of patients without downtime while ingesting real-time wearable data. |
| **Regulatory Compliance** | HIPAA/GDPR drive our choices for encryption, audit logging, and isolated VPC design. |
| **Time-to-Market** | The eleven-week delivery window requires managed services (Fargate, Cognito, MSK) to minimize ops toil. |
| **Cost-Efficiency** | MVP phase should remain <$300 / month; autoscaling and spot pricing mitigate idle spend. |
| **Extensibility** | Event-driven, domain-oriented services allow future modules (e.g., Tele-health Bridge) to subscribe without direct coupling. |

*Note.* This figure provides a visual timeline of the project's milestone deadlines using a Gantt chart.

## 6.2 Logical System Architecture

**Table 3**

*Logical System Architecture Table*

| **Tier** | **Logical Component** | **Responsibilities** | **Tech Choice** |
| --- | --- | --- | --- |
| **Client** | Flutter Mobile + Web | UI, device APIs (camera, microphone), local cache, Stripe Elements, WebSocket listener | Dart / Flutter |
| **Edge** | Amazon CloudFrontWAF | TLS termination, SPA hosting, DDoS & OWASP rules | CloudFront, AWS WAF |
| **API Gateway** | REST & WebSocket endpoints | Routing, JWT validation (Cognito), request throttling | Amazon API Gateway |
| **BFF (Service Mesh)** | Spring Boot API-BFF | Aggregates downstream calls for mobile performance | ECS Fargate (Java 17) |
| **Domain Micro-services** | User-Service, Billing-Service, Scheduling-Service, Communication-Service, Analytics-Service | Each owns its schema; publishes Kafka topics | Spring Boot -> ECS Fargate |
| **Data** | Relational DB, Document DB, Time-series DB, Object Storage | RDS (PostgreSQL), DynamoDB, Timestream, S3 | Managed AWS |
| **Integration** | Stripe, Fitbit, Nest, FCM/APNs | External REST / Webhooks | HTTPS, JWT mutual-TLS |
| **Streaming** | Event Bus (“CareBus”) | Domain events, audit trail | Amazon MSK (Kafka) |
| **Analytics** | Athena / QuickSight | Ad-hoc queries, dashboards | Nightly ETL via Glue-Jobs |
| **Security & Ops** | Cognito, KMS, IAM, CloudWatch, EventBridge | AuthN/Z, secrets, logs, scheduled jobs | AWS native |

*Note.* This table identifies the key architectural objectives and rationale for the CareConnect platform.

6.3 High-Level System Architecture Diagram

**Figure 2**

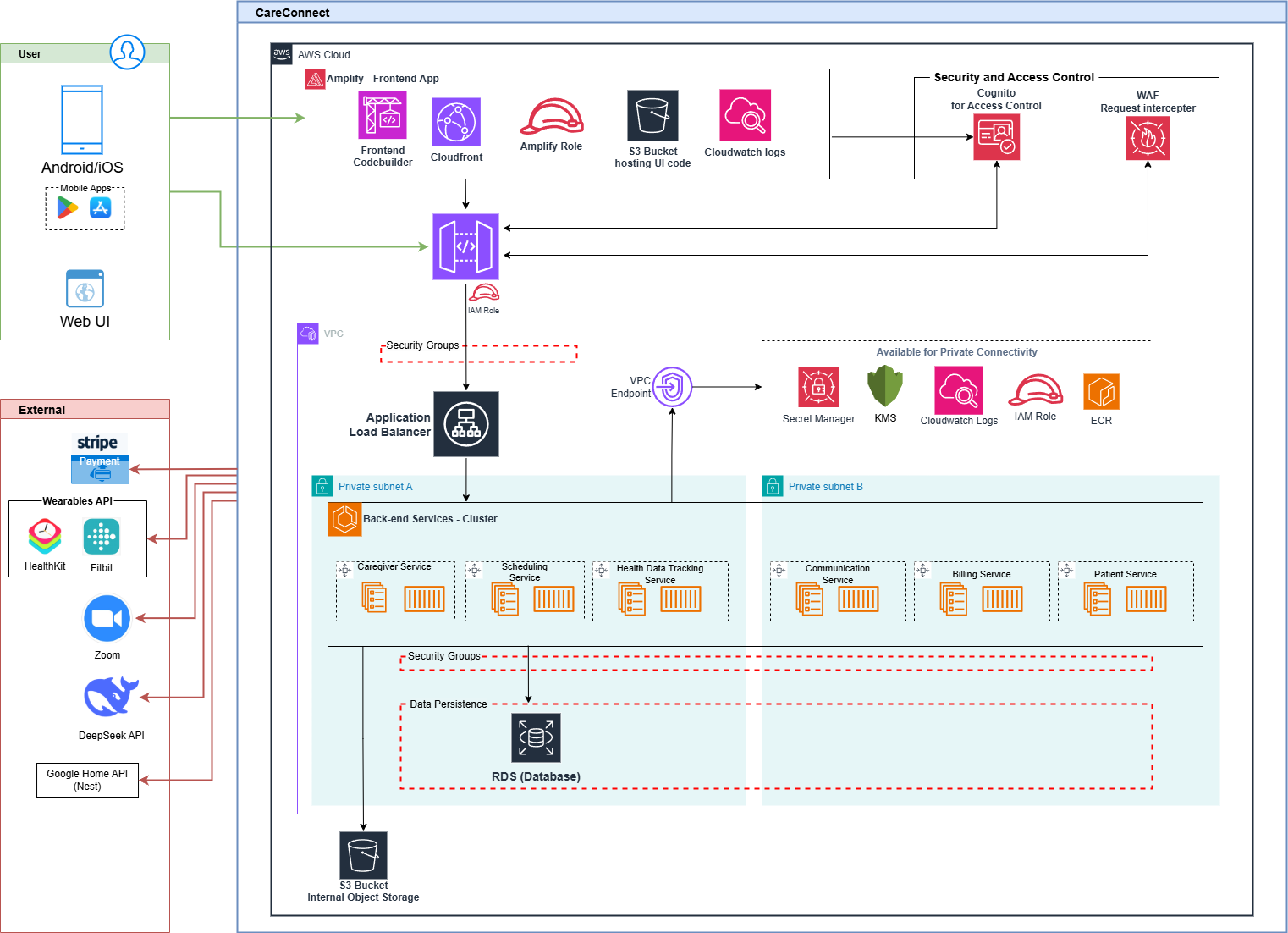
*High-Level System Architecture Diagram 1*

A diagram of a company

AI-generated content may be incorrect.*Note.* This figure provides a visual timeline of the project's milestone deadlines using a Gantt chart.

**Figure 3**

*High-Level System Architecture Diagram 1*



*Note.* This table identifies the key architectural objectives and rationale for the CareConnect platform.

## 6.4 Architectural Rationale

**Table 4**

*Architectural Rationale Table*

| **Concern** | **Design Decision** | **Justification** |
| --- | --- | --- |
| **Scalability** | Stateless Spring Boot services on ECS Fargate; event-driven via Kafka (MSK). | Auto-scales by CPU or queue lag; zero server management. |
| **Latency** | API-BFF aggregates calls; CloudFront edge caching; WebSocket for live metrics. | Reduces mobile roundtrips; meets <300 ms UX goal. |
| **Security & HIPAA** | Cognito JWT, TLS 1.3, at-rest AES-256 (KMS), WAF, audit to Lake Formation. | Satisfies HIPAA/GDPR encryption and audit mandates. |
| **Cost-control** | Fargate spot where possible; scale-to-zero Lambda ETL; S3 tiering to Glacier. | Keep MVP hosting below $300 a month during dev. |
| **Extensibility** | Clean domain services; events on MSK, and Timestream for IoT expansion. | New micro-services subscribe without coupling. |
| **Dev Velocity** | Single Git repo, multi-module Maven; Shared DTOs; OpenAPI contracts. | Reduces cross-team merge pain; enables code-gen stubs. |

*Note.* This table details the logical system architecture, including component tiers, responsibilities, and technology choices.

## 6.5 Runtime Flows

1. **Payment Happy-Path**  
    Patient → API-BFF → Billing-Svc → Stripe Charge → RDS commit → Push “ACTIVE.”
2. **Real-time Vital Update**  
    Fitbit Sync Lambda → Timestream → Analytics-Svc publishes event → WebSocket to client (<5 s).
3. **Nightly Batch**  
    EventBridge 02:00 UTC trigger → Glue Job aggregates TS → summary metrics → QuickSight refresh.

## 6.6 Technology Stack Summary

**Table 5**

*Technology Stack Summary Table*

| **Layer** | **Tech** | **Notes** |
| --- | --- | --- |
| **Front-end** | Flutter 3.x (Dart 3) | Single code-base iOS/Android/Web |
| **BFF & Services** | Java 17, Spring Boot 3, Maven multi-module | Micrometer, Resilience4j |
| **Data** | RDS (Postgres 15), DynamoDB, Timestream 1.0 | Flyway migrations |
| **Messaging** | MSK (Kafka 3.x, TLS) | Topic per domain (billing, vitals, alerts) |
| **CI/CD** | GitHub Actions → AWS CodeBuild / CodeDeploy | Blue green ECS; semantic-version tags |
| **Observability** | CloudWatch Logs, Prometheus ECS sidecar, Grafana Cloud | Traces via AWS X-Ray |
| **Security** | AWS Cognito (OIDC), KMS, WAFv2, IAM-least privilege | S3 Block Public Access |

*Note.* This figure presents a high-level system architecture diagram showing data flow and component integration.

## 6.7 Deployment Topology Highlights

* All Spring services run as container images in an ECS Fargate cluster inside a private subnet; only API Gateway and CloudFront are publicly routable.
* Secrets (DB passwords, Stripe keys) are stored in AWS Secrets Manager; tasks assume IAM roles with the least-privilege policy generated via AWS SAM.
* Observability: structured JSON logs are shipped to CloudWatch; metrics and traces are exported via OpenTelemetry to a managed Grafana workspace.
* Blue/Green deployments are orchestrated by CodeDeploy. Rollback is automatic on canary health-check failures (<2 min).

This section outlines a secure, scalable, serverless and cloud-native technical architecture for CareConnect. We will be using Amazon Web Services (AWS) to build the infrastructure and Flutter to build the user interface.

This setup is well-suited for modern web and mobile applications that require secure access control, high availability while being able to scale to zero and integrate with internal AWS services and external third-party seamlessly.

## 6.8 Frontend

**CareConnect’s frontend** is primarily designed for ease-of-use by both patients and caregivers, with configurable features integrated in the code to allow for user preference of the UX. The frontend will be comprised of code using Flutter and Dart, allowing for support from a wide range of platforms. The UI was designed with role-specific functionality in mind. This allows for separate features dependent on the user’s role – patient or caregiver.

**Main Pages:**

* Welcome/Login page
* Caregiver/Patient dashboard
* Admin panel

## 6.9 Backend

The backend will constitute a set of AWS services for the core infrastructure and integrate external third-party resources and API(s). We will be using Terraform for infrastructure as code (IaC) management that would help to spin up, update, and take down all the resources needed in our infrastructure.

1. Amazon Cognito – *User Authentication and Authorization*

* Purpose: Manages user sign-up, sign-in, and access control.
* Features:
  + Supports OAuth2.0, SAML, and OpenID Connect.
  + Integrates with API Gateway to secure endpoints.
  + Enables multi-factor authentication (MFA) and social identity providers (Google, Apple, etc.).

1. Amazon API Gateway

* Purpose: Act as the entry point for all requests
* Features:
* Routes requests to backend services.
* Integrate with Cognito for authentication and authorization.
* Supports rate limiting, throttling, and monitoring.

1. Amazon Elastic Container Service (ECS)

* Purpose: Host and orchestrate containers for the application, the backend.
* Features:
  + Supports Fargate for serverless type of launch
  + Integrate with Application Load Balancer (ALB) for traffic distribution
  + Scalability
  + Secure networking to block all external communications

1. SpringBoot – For backend development

* Purpose: Develop Rest API with Java
* Feature:
* Robust and secure
* Uses Java
* Supports and integrates countless external technologies for all aspects.

1. Amazon Relational Database Service (RDS) – Managed SQL Database Server

* Purpose: Stores structured applications data.
* Features:
* Supports engines like MySQL and PostgreSQL.
* Offers automated backups, and read replicas
* Encrypted at rest and in transit

For CareConnect, we will be using PostgreSQL.

1. Amazon S3 – for Object Storage

* Purpose: Stores static assets like image, documents.
* Features:
* Cheap
* Durable
* Scalable

1. Amazon CloudWatch – for Monitoring and Logging

* Purpose: Provides observability into application and infrastructure
* Features:
* Collects logs and metrics from ECS, API Gateway, RDS.

1. AWS Identity and Access Management (IAM) – Security and Permissions

* Purpose: Manage permission between AWS services, resources, and users.
* Features:
* Enforces least privilege access (LPA)
* ECS tasks assume IAM roles to access RDS, S3 and more

## 6.10 Database

The CareConnect database is designed to securely store and manage application data related to users, tasks, health records, and system operations. The database will ensure data integrity, support HIPAA compliance, and provide reliable access for both mobile and backend components.

**Database Technology Stack:**

* **Database Type:** Relational (SQL-based)
* **Preferred Platform:** PostgreSQL (hosted on AWS RDS, a cloud service)
* **ORM/Access Layer:** Dart packages compatible with Flutter, such as sqflite or drift for local caching if offline support is needed.

**Key Database Components:**

* **Users Table**  
   Stores account information for patients, caregivers, and admins. Includes authentication credentials, contact details, and role-based metadata.
* **Patients Table**  
   Contains patient-specific profiles, medical notes, emergency contacts, and linked caregiver relationships.
* **Care Tasks Table**  
   Stores both pre-defined and custom tasks related to care, such as medications, meals, and appointments, with timestamps, frequency, and completion status.
* **Health Logs Table**  
   Records manually entered data like vitals, symptoms, and mood tracking for patients, linked by patient ID and timestamped.
* **Notifications Table**  
   Logs system-generated alerts and reminders for both patients and caregivers, tracking delivery and response status.
* **Billing Table**  
   Manages billing events, payment history, subscription status, and Stripe transaction references.

**Security & Compliance Features:**

* Role-based access enforced through middleware
* Data encryption at rest and in transit
* Regular backups and automated failover support
* Data retention and purging policies for compliance

**Scalability Considerations:**

* Support for multi-tenancy for different organizations or caregiver groups
* Optimized indexes for common query patterns (e.g., task retrieval by date or caregiver)
* Future-readiness for migration to a distributed database if user volume scales

## 6.11 APIs

The CareConnect platform requires robust and secure API integrations to support its modular feature set. Each subsection describes the purpose, integration model, and relevant considerations for the respective API.

### 6.11.1 Fitbit Web API

* **Purpose:** Retrieve heart rate and step count data from Fitbit devices.
* **Authentication:** OAuth 2.0
* **Data Accessed:** Heart rate, step count
* **Platform Support:** Android, iOS
* **Considerations:** Requires user account linking and periodic data sync.

### 6.11.2 Apple HealthKit API

* **Purpose:** Retrieve heart rate and step count data on Apple Health on iOS devices.
* **Authentication:** Native iOS permission-based access
* **Data Accessed:** Heart rate, step count
* **Platform Support:** iOS only
* **Considerations:** Requires explicit user consent for each metric.

### 6.11.3 Google Health Connect API

* **Purpose:** Retrieve heart rate and step count data onAndroid devices through a unified API for Google Fit and partner apps.
* **Authentication:** Android permission-based access
* **Data Accessed:** Heart rate, step count
* **Platform Support:** Android only
* **Considerations:** Health Connect to be installed and permitted.

### 6.11.4 Google Smart Device Management (SDM) API

* **Purpose:** Interact with Google Nest smart home devices for environmental monitoring.
* **Authentication:** OAuth 2.0
* **Data Accessed:** Motion events, device status, live camera availability
* **Platform Support:** Cloud-based
* **Considerations:** Must be carefully managed for privacy.

### 6.11.5 Alexa Smart Home Skill API

* **Purpose:** Enable control and monitoring of Alexa-compatible smart home devices.
* **Authentication:** OAuth 2.0
* **Data Accessed/Controlled:** Device state, control directives**.**
* **Platform Support:** Cloud-based
* **Considerations:** Integration requires Smart Home Skill certification and user account linking.

### 6.11.6 OpenFDA API

* **Purpose:** Retrieve official medication information using NDC codes.
* **Authentication:** None (public API)
* **Data Accessed:** Medication name, dosage, manufacturer, instructions
* **Platform Support:** Web-based

**Considerations:** Used to auto-populate medication entries from scanned NDC codes.

6.11.8 Jitsi API

* Purpose: Use audio and video calls for various communication functions in CareConnect.
* Authentication: JWT Token
* Data Accessed: Voice, video, calendar schedule.
* Platform Support: Web-based. Apple iOS. Android.
* Considerations: Used for in-app messaging services. HIPPA compliant with set-up.

## 6.12 AI Services

### 6.12.1 Purpose and User Flow

* Patients use the “Ask” feature in the app to type or ask questions about their treatment plans or symptoms.
* Utilize DeepSeek, a HIPAA/GDPR-compliant LLM, to power the “Ask” feature for patients to pose personalized care questions.
* If a patient’s question looks risky or outside the AI’s safe zone, it’s held back for a real clinician to review before the patient sees it.

## 6.12.2 Technical Architecture

#### 6.5.2.1 Frontend Integration

* Secure REST calls from the mobile app to an AI Gateway.

#### 6.12.2.2 AI Gateway

#### Handles authentication/authorization, rate limiting, and request routing to DeepSeek.

#### 6.12.2.3 DeepSeek Service

* Retrieves patient context from the backend, invokes LLM, and returns structured JSON answers.

#### 6.12.2.4 Review & Escalation

* Flag any high-risk queries for human clinician review before delivering to the patient.

### 6.12.3 Security and data Protection

* + Encryption: All PHI (Personal Health Information) in transit and at rest must use AES-256 encryption, and TLS 1.2+ for network calls.
  + Data Minimization: Only supply DeepSeek with the minimal necessary attributes (e.g., plan ID, symptom entries) and discard logs after compliance retention periods.
  + Audit Logging: Maintain immutable logs of AI interactions, including timestamps, user IDs, and query transcripts, stored in a secure, access-controlled ledger.

### 6.12.4 Compliance and Governance

* + Privacy by Design: Embed compliance controls into each component (e.g., tokenization, PII redaction) to ensure HIPAA and GDPR adherence from day one.
  + Human-in-the-Loop: Escalate any AI responses outside predefined safety envelopes to clinicians for manual approval.
  + Periodic Review: Conduct quarterly risk assessments and model audits to validate compliance and performance.

### 6.12.5 Summary and Future Enhancements

#### 6.12.5.1 AI Service Summary

* The ability for patients to easily and efficiently get the information they need without having to wait for a human caregiver that might not be available is something that could significantly increase patient quality of life. The AI integration piece for the ‘Ask’ function is a great way to achieve this, though takes some careful planning to ensure we are implementing strict security best-practices and making sure we keep the quality-of-care high through human review when necessary.

#### 6.12.5.2 Future Enhancements

* Potential avenues for further AI integration would be automated care alerts for caregivers based on aggregating information on vitals and medicine reception by the patients and running LLM inference against the data to find anomalies. These anomalies would be reported back to caregivers in real-time as they are found to ensure the quickest triage and remedy.

## 6.13 Hosting

### 6.13.1 Cloud Deployment Model

* Adopt API Gateway as the entry point for all client requests, enforcing authentication and throttling.
* Use AWS Fargate with ECS to run containerized services (backend API, AI Gateway) in a serverless-managed cluster, eliminating server management overhead.

### 6.13.2 Containerization and Orchestration

* Docker Images: Build stateless microservices, each with immutable Docker images stored in Amazon’s ECR (Elastic Container Registry)
* ECS Task Definitions: Define CPU/memory limits and Amazon IAM (Identity and Access Management) roles per task, ensuring least-privilege access to AWS resources.
* Auto Scaling: Configure ECS Service Auto Scaling to adjust desired tasks based on CPU/memory metrics.

### 6.13.3 Serverless-First & Scale-to-Zero

* Leverage AWS Lambda to achieve scale-to-zero and cost efficiency.

### 6.13.4 Security and Reliability

* Containers live inside a private network segment with no direct public access, and outbound calls go through controlled gateways.
* We protect the front door with a web firewall and standard defenses against denial-of-service attacks.
* Data backups and multi-zone deployments ensure that, even if one data center goes down, the service stays up, and data stays safe.

## 6.14 Security

Security is a core principle in the design and operation of the CareConnect application. Our approach integrates industry best practices to protect sensitive health information and ensure compliance with regulations such as the Health Insurance Portability and Accountability Act (HIPAA) and the General Data Protection Regulation (GDPR). Security measures are implemented across all components of the system—including the user interface, backend services, data storage, APIs (Application Programming Interfaces), and AI (Artificial Intelligence) features—through a combination of encryption, access controls, monitoring, and secure development practices.

### 6.14.1 Data Protection

#### 6.14.1.1 Encryption

* All Personal Health Information (PHI) is encrypted both when stored and during transmission, using the strong industry encryption standards of AES 256 for data being stored and TLS 1.2 for any data in transit between systems.

#### 6.14.1.2 Data Minimization

* We limit the collection and use of personal data to what is necessary for providing services. Identifiable information is replaced with tokens where possible, and logs are anonymized to protect user privacy.

### 6.14.2 Authentication and Access Control

#### 6.14.2.1 User Authentication

* Secure authentication mechanisms are in place, supporting multi-factor authentication (MFA) and social login options, to verify user identities effectively.

#### 6.14.2.2 Role-Based Access Control (RBAC)

* Access to system features and data is determined by user roles (such as patient, caregiver, or administrator), ensuring users can only access information pertinent to their responsibilities.

#### 6.14.2.3 Principle of Least Privilege

* Access permissions are granted based on the minimum necessary access required for a user's role, reducing potential security risks, and will be managed through AWS IAM (Identity & Access Management)

### 6.14.3 Secure APIs

#### 6.14.3.1 API Gateway Protection

* All API traffic is managed through AWS API Gateway, a secure gateway that enforces rate limiting and input validation to prevent misuse and attacks.

### 6.14.4 Infrastructure Hardening

#### 6.14.4.1 Network Isolation

* System components are deployed within isolated virtual networks, preventing direct exposure to the internet and reducing the attack surface.

#### 6.14.4.2 Web Application Firewall (WAF)

* A firewall is in place to protect against common web exploits such as SQL injection and cross-site scripting (XSS).

#### 6.14.4.3 Firewall Rules

* Strict firewall configurations control traffic between system components, allowing for only necessary communication.

### 6.14.5 Monitoring and Logging

#### 6.14.5.1 Real-Time Monitoring

* CareConnect will leverage the capabilities of AWS CloudWatch to monitor threats to the system continuously and actively, monitoring administrators whenever an anomaly is detected.

#### 6.14.5.2 Audit Logging

* All critical actions, including logins, data modifications, and AI interactions, are recorded in secure and unchangeable logs to support auditing and compliance efforts.

### 6.14.6 Vulnerability Management

#### 6.7.6.1 Interval Scanning

* The system undergoes routine scans to identify and address vulnerabilities in both backend and mobile components.

#### 6.14.6.2 Automated Patching

* Security patches are applied automatically to system components as part of our continuous integration and deployment processes, ensuring timely updates.

### 6.14.7 Compliance Measures

#### 6.14.7.1 Regulatory Compliance

* The system is designed with privacy and security considerations from the outset, aligning with HIPAA and GDPR requirements.

#### 6.14.7.2 Data Retention and Deletion

* Personal data is retained only as long as necessary for service provision, and users can request data deletion in accordance with legal guidelines.

# **7. Project Deliverables**

* Fully functional mobile apps (Caregiver & Patient)
  + A modern and well-designed, developed, and tested mobile application intended to enhance the overall patient/caregiver experience.
* Web dashboard for caregivers
  + Whether in a professional or family setting, the caregiver will have the ability to add new, view a list of, and meet one-on-one with their patients.
* Admin panel for operations
  + Ability for admins to add/edit/delete entries to the database.
  + Allows access to all features of the UI.
* Technical documentation
  + Developer comments throughout the code for readability.
  + Update the GitHub README file with any important instructions/details.
* User manuals & onboarding guides
  + Includes description of all UI pages and features.

# **8. Risk & Mitigation**

## 8.1 Risk Probability Matrix

**Table 6**

*Risk Probability Matrix Table*

|  | **Probability** | | | |
| --- | --- | --- | --- | --- |
| **Impact** | **1 = High (80% ≤ x ≤ 100%)** | **2 = Medium High (60% ≤ x ≤ 80%)** | **3 = Medium Low (30% ≤ x ≤ 60%)** | **4 = Low (0% ≤ x ≤ 30%)** |
| A = High (Rating 100) | 100 –Very High Exposure | 80 –Very High Exposure | 60 –High Exposure | 30 – Moderate Exposure |
| B = Medium (Rating 50) | 50 – High Exposure | 40 – Moderate Exposure | 30 – Moderate Exposure | 15 – Low Exposure |
| C = Low (Rating 10) | 10 – Low Exposure | 8 – Low Exposure | 6 – Low Exposure | 3 – Very Low Exposure |

*Note.* Risk Probability Matrix table created with reference from previous 2023 SWEN 670 AlphaSoft project (AlphaSoft, 2023).

## 8.2 Risk Identification, Probability, and Mitigation

Table 7

*Risk Probability and Mitigation Table*

| **Risk Description** | **Probability** | **Impact** | **Score** | **Mitigation** |
| --- | --- | --- | --- | --- |
| Project – Team members not completing assigned work. | 1 | A | 60 | High probability high impact. We will have open channels of communication to keep members accountable. If members do not complete their work, then we will reassign project parts to other team members. |
| Project – Team members do not know necessary technology to implement CareConnect | 1 | B | 50 | High probability and high impact. We will allot time for team members to learn necessary technology to implement CareConnect features. |
| Product – Lawsuit | 1 | C | 10 | Low probability and low impact. Until the first deployment and public use of CareConnect, a lawsuit is unlikely to occur. |
| Product – Client changes the scope of the project | 2 | B | 40 | Moderate probability and moderate exposure. We will use the change process to record change in scope and plan accordingly. |
| Product - Compliance failure in ensuring HIPPA procedure for data | 4 | A | 30 | Low probability and high impact. Conduct HIPPA audit and legal review. Update impacted service to regain HIPPA compliance. |
| Product - Failure to integrate 3rd party services with CareConnect Application | 4 | A | 30 | Moderate probability and high impact. Identify alternative 3rd parties with working API integrations; add failure handling when API calls do not work. |

*Note.* Risk Probability Matrix table created with reference from previous 2023 SWEN 670 AlphaSoft project (AlphaSoft, 2023).

# **9. Budget Estimate**

## 9.1 Labor Cost

The main component for the budget cost of the CareConnect project will be the cost of labor per team member. An estimate of labor costs can be viewed in Table 1 below:

**Table 8**

*Budget Estimate for Labor Costs*

| **Team Member Name** | **Project Role** | **Cost Per Hour** | **Work Hours/Week** | **Weekly Salary** | **Monthly Salary** | **3- Month Cost** |
| --- | --- | --- | --- | --- | --- | --- |
| Alyssa Marielle Harding | Team Lead | $48.44 | 20 | $968.80 | $3,875.20 | $11,625.60 |
| Ashenafi Grbreegziabhere | Technical Lead/Architect | $63.20 | 20 | $1,264.00 | $5,056.00 | $15,168.00 |
| Edwenson Raphael | Technical Lead/Architect | $63.20 | 20 | $1,264.00 | $5,056.00 | $15,168.00 |
| Dat Truong | UX Designer | $45.85 | 20 | $917.00 | $3,668.00 | $11,004.00 |
| Juan Gaucin | Test Engineer | $47.44 | 20 | $948.80 | $3,795.20 | $11,385.60 |
| Christian Yawn | Software Developer | $47.44 | 20 | $948.80 | $3,795.20 | $11,385.60 |
| Diane Angeles | Business Analyst | $47.44 | 20 | $948.80 | $3,795.20 | $11,385.60 |
| Luke Curran | UX Designer | $45.85 | 20 | $917.00 | $3,668.00 | $11,004.00 |
| Astha Malla-Paudel | Business Analyst | $47.44 | 20 | $948.80 | $3,795.20 | $11,385.60 |
| Torie Bias | Test Engineer | $47.44 | 20 | $948.80 | $3,795.20 | $11,385.60 |
| Alex Vecchioni | Software Developer | $47.44 | 20 | $948.80 | $3,795.20 | $11,385.60 |
| Maria Ramirez | Software Developer | $47.44 | 20 | $948.80 | $3,795.20 | $11,385.60 |
| Alireza Minagar | Group Project Manager | $100.00 | 10 | $1,000.00 | $4,000.00 | $12,000.00 |
|  | **Totals** | $646.00 | 260 | $13,921.20 | $55,684.80 | $167,054.40 |

*Note.* Cost per hour rates retrieved from the U.S. Bureau of Labor Statistics

## 9.2 Materials Cost

Given this project is to be completed in the scope of a capstone course, team members will be responsible for their hardware technology used for development (laptops, keyboards, mice, etc.). Thus, hardware costs are estimated to be $0.

## 9.3 Infrastructure Cost

This subsection converts the target technical architecture (Section 6) into a quantified operating-expense forecast so that project sponsors can gauge cash burn and approve the required AWS spend-limit. The estimate follows a “bottom-up, usage-based” methodology commonly used in U.S. SaaS cost modelling:

1. Reference Architecture Mapping – Each logical component (API Gateway, Fargate tasks, RDS, MSK, S3, etc.) is mapped to its corresponding billable SKU in the AWS Pricing Catalog (Amazon Web Services, 2024).
2. Sizing Assumptions – Workload drivers—concurrent users, API calls, data-ingest volume, and retention—are derived from the SRS (user counts) and the Performance NFRs (<200 ms p99, 99.9 % SLA). Conservative headroom (+20 %) is applied to prevent cost overruns during load spikes.
3. Unit-Price Collection – April 2025 on-demand (or free-tier/spot) prices for the us-east-1 region were pulled from the AWS Pricing API and validated against the AWS Pricing Calculator worksheets archived in project SharePoint (rev 1.2). Stripe, Apple, and Google fees reference each vendor’s public tariff sheet (Stripe, 2025; Apple Developer Program, 2025).
4. Estimation Technique – For recurring services, costs are projected as

MRC = Unit Price × Monthly Usage Qty   
Project Burn = MRC × 3 months (11-week MVP + 1-week hyper-care)

One-off items (app-store enrollments, TLS certificates) are treated as Cap-Ex charges in Month 1.

1. Contingency Allocation – A 10% buffer is added to cover unseen data-egress or long–retention growth, consistent with company FP&A guidelines for cloud projects.
2. Validation – Numbers were peer-reviewed by the CloudOps FinOps analyst and reconciled against the previous quarter’s reference workloads to ensure no line item deviates by >15 % from historical benchmarks.

The resulting figure ≈ USD 1.3 K for the entire MVP period—represents less than 1 % of total project budget, affirming the decision to use managed, serverless AWS services rather than self-hosted infrastructure (Stripe, Inc., 2025; Apple Inc., 2025).

**Table 9**

*Infrastructure Cost Table*

| **#** | **Service** | **Sizing Assumption** | **Unit Price\*** | **Monthly Qty** | **MRC USD** |
| --- | --- | --- | --- | --- | --- |
| 1 | **ECS Fargate (CPU+RAM)** | 4 tasks × (0.25 vCPU / 0.5 GB) avg 50 % util | 0.04048 $/vCPU-h0.004445 $/GB-h | 288 vCPU-h288 GB-h | **16** |
| 2 | **API Gateway REST** | 3 M API calls/mo | 3.50 $ per 1 M | 3 | **11** |
| 3 | **API Gateway WebSocket** | 500 active conn. · 720 h | 0.25 $/M mins | 216 000 mins | **54** |
| 4 | **AWS Cognito MAU** | 6 000 unique users | 0.0055 $/MAU | 6 000 | **33** |
| 5 | **Amazon RDS-PostgreSQL** | db.t4g.medium Multi-AZ | 0.101 $/h × 730 h | 730 | **74** |
| 6 | **RDS storage & I/O** | 40 GB gp3 + 3 M I/O | 0.115 $/GB-mo + 0.20 $/M I/O | – | **8** |
| 7 | **Amazon DynamoDB** | 15 GB + 2 M WCUs/RCUs | 1.25 $/GB + 1.25 $/M WCU | – | **25** |
| 8 | **Amazon Timestream** | 20 GB ingestion + 200 M queries | 0.50 $/GB + 0.02 $/M | – | **20** |
| 9 | **Amazon MSK Serverless** | 100 GB in/out; 50 GB-h storage | 0.11 $/GB-in/out + 0.01 $/GB-h | – | **16** |
| 10 | **S3 Standard + PUT/GET** | 200 GB data + 1 M ops | 0.023 $/GB + API | – | **8** |
| 11 | **CloudFront Data Out** | 400 GB/mo | 0.085 $/GB (Tier 1) | – | **34** |
| 12 | **AWS Lambda (Glue ETL)** | 1 M req, 80 GB-s | 0.20 $/M + 0.00001667 $/GB-s | – | **4** |
| 13 | **CloudWatch & X-Ray** | 50 GB logs, 5 M traces | 0.50 $/GB + 5 $/M | – | **30** |
| **Subtotal – AWS Recurring** | | | | | **333 $/month** |

*Note.* This table defines a matrix to assess and categorize risks based on probability and impact.

### **9.3.1 AWS Core Services**

\* Prices: AWS public on-demand rates as of April 2025, us-east-1; Fargate uses **spot** where idle utilization <35 %.

### **9.3.2 Third-party / One-time Fees**

**Table 10**

*Fees Table*

| **Service** | **Type** | **Cost USD** |
| --- | --- | --- |
| Apple Developer Program | Annual | **99** |
| Google Play Console | One-time | **25** |
| Stripe – Standard Plan | Usage-based | < 0.3 % per txn (pass-through) |
| Sectigo TLS Wildcard (if not using ACM) | 1-year cert | **80** |

*Note.* This table lists specific project risks along with their likelihood, potential impact, and mitigation plans.

### **9.3.3 Total Budget for MVP Phase (13 weeks ≈ 3 months)**

**Table 11**

*Total Infrastructure Table*

| **Category** | **Monthly** | **3-Month** | **Notes** |
| --- | --- | --- | --- |
| AWS Core (Table 1 subtotal) | **333** | **999** | uses spot + free tier credits |
| λ / Glue ETL data transfer | Incl. | Incl. | fold into AWS subtotal |
| App-store & TLS fees | — | **204** | 99 + 25 + 80 |
| **Subtotal** |  | **1,203** |  |
| Contingency 10 % |  | **120** | burst traffic, data-out |
| **Grand Total (Infrastructure)** |  | **≈ 1,320 USD** |  |

*Note.* This table outlines estimated labor costs based on role, hourly rate, and time commitment.

### **9.3.4 Cost-Control & Optimization Plan**

**Table 12**

*Cost-Control Table*

| **Lever** | **Action** | **Expected Savings** |
| --- | --- | --- |
| **Compute** | Keep QA & Feature branches on smaller 0.25 vCPU tasks; auto-stop dev DB nights/weekends. | 15–20 % |
| **Storage** | Lifecycle S3 uploads >90 days to Glacier Instant; compress CloudWatch logs with AWS Logs Insights. | 5–7 % |
| **Traffic** | Enable CloudFront Origin Shield; consolidate API endpoints to reduce DNS lookups. | 3–5 % |
| **Reserved Instances** | Purchase 1-yr RDS Standard RI after MVP proves stable. | 35–40 % on DB layer |
| **Spot** | Maintain Fargate spot capacity providers with on-demand fallback. | 50–60 % on compute bursts |

*Note.* This table presents monthly infrastructure costs for AWS and related services.

### **9.3.5 Alignment to Financial Policies**

* **Cap-Ex vs. Op-Ex.** All spend is Op-Ex (pay-as-you-go), aligning with U.S. GAAP guidance for cloud-service costs (ASC 350-40).
* **Cost Recognition.** Expenses booked monthly via consolidated AWS invoice to *Cost Center IT-R&D-CareConnect*.
* **SOX Controls.** AWS accounts integrated with enterprise AWS Organizations. Detailed billing reports exported to an immutable S3 bucket for audit.
* **HIPAA Compliance Fees.** AWS BAA services carry no surcharge; Stripe’s PCI scope remains with the vendor. No additional license cost foreseen.

### **9.3.6 Budget Summary for Project Plan**

**Table 13**

*Budget Summary Table*

| **Line Item** | **Amount (USD)** |
| --- | --- |
| Labor Cost (Section 9.1) | 167 054 |
| Materials Cost (Section 9.2) | 0 |
| Infrastructure (9.3) | 1 320 |
| **Total Project Budget** | **≈ 168 374 USD** |

*Note.* This table outlines one-time or third-party service fees related to app distribution and TLS security.

The infrastructure allocation represents <1 % of total project spends, validating a serverless, pay-as-you-go strategy that keeps capital exposure minimal while meeting security and compliance requirements.

# **10. Roles & Responsibilities**

For successful planning and execution of CareConnect, the identification of roles and responsibilities of the project team is crucial. Since there is a time limit of 3 months for the course duration, multiple team members will be taking on multiple roles to complete the project deliverables.

## 10.1 Roles

**Table 14**

*Roles & Responsibilities Table with Assigned Team Members*

| **Role** | **Team Members** | **Responsibilities** |
| --- | --- | --- |
| Client | Dr. Assadullah  Roy Gordon  Ashley Wayne | * Provide product requirements and feedback on product progress |
| Project Mentor | Roy Gordon  Ashley Wayne | * Advise student team members * Review deliverables upon request and provide constructive feedback |
| MS Teams & GitHub Mentor | Robert Wilson | * Act as Admin role for MS Teams and GitHub resources * Schedule MS Teams meetings * Provide GitHub access for course |
| Class Project Manager | Alireza Minagar | * Organize team project progress for all teams * Provide updates to course professor |
| Team Lead | Alyssa Marielle Harding | * Organize project team deliverables * Provide project progress to Class Project Manager * Provide team member communications |
| Backup Team Lead | Astha Malla-Paudel | * Help Team Lead when Team Lead is unavailable * Assume Team Lead responsibilities when Team Lead is unavailable |
| Technical Lead/Architect | Edwenson Raphael | * Design project architecture * Lead developers in implementation |
| Backup Technical Lead/Architect | Ashenafi Grbreegziabhere | * Help Technical Lead/Architect with designing project architecture |
| Lead Software Developer | Ashenafi Grbreegziabhere | * Lead Software Developer members in implementing functional code * Meet with Technical Lead/Architect to ensure components are implemented correctly |
| Software Developer | Edwenson Raphael  Alex Vecchioni  Maria Ramirez  Alyssa Marielle Harding  Christian Yawn  Luke Curran | * Implement functional code * Seeking guidance from Technical Lead/Architect * Act as Backup Lead Software Developer if need arises |
| Lead Business Analyst | Astha Malla-Paudel | * Lead business analysts * Identify use cases * Confer with leads to determine functional capabilities of use cases * Meet with stakeholders to discuss functional requirements |
| Business Analyst | Diane Angeles  Ashenafi Grbreegziabhere  Edwenson Raphael  Dat Truong  Luke Curran  Alyssa Marielle Harding  Juan Gaucin | * Help Lead Business Analyst to identify use cases * Meet with stakeholders to discuss functional requirements * Act as Backup Lead Business Analyst if need arises |
| Lead UI/UX Designer | Dat Truong  Luke Curran | * Lead UI/UX Designers * Confer with leads of UI designs * Help design wireframes for product mockups |
| UI/UX Designer | Astha Malla-Paudel  Christian Yawn  Edwenson Raphael | * Design wireframes for product mockups * Collaborate with developers to implement product designs * Act as Backup Lead UI/UX Designer if need arises |
| Lead Tester | Juan Gaucin | * Create test matrix for unit, integration, and system testing * Confer with leads of test plans * Work with testers to create tests and perform tests against product |
| Tester | Dat Truong  Alyssa Marielle Harding  Torie Bias  Luke Curran | * Help create testing scenarios for product * Implement testing scenarios * Perform testing scenarios and record results * Act as Backup Lead Tester if need arises |

*Note.* Roles table created with reference from previous 2023 SWEN 670 AlphaSoft project (AlphaSoft, 2023).

## 10.2 RACI Matrix

**Table 15**

*RACI Matrix Table*

Key:

* R – Responsible
* A – Accountable
* C – Consulted
* I - Informed

| **Task** | **Client** | **Project Manager** | **Team Lead** | **Business Analyst** | **Technical Lead/Architect** | **Developer** | **UI/UX Designer** | **Tester** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Milestone 1** |  |  |  |  |  |  |  |  |
| Project Plan | I | I | A | C | I |  |  |  |
| SRS | I | I | A | C | C | I | C | I |
| **Milestone 2** |  |  |  |  |  |  |  |  |
| TDD | I | I | A |  | R |  |  |  |
| Software Test Plan | I | I | A |  |  |  |  | R |
| Flutter/Dart |  | I | I |  | A | R | R |  |
| Mobile App Development |  | I | I |  | A | R | C |  |
| Web Service Development |  | I | I |  | A | R | C |  |
| Infrastructure setup |  | I | I |  | A | R |  |  |
| UI/UX Design |  | I | I |  |  |  | R |  |
| 3rd-Party Device Integration |  | I | I |  | A | R |  |  |
| 3rd- Party Billing Integration |  | I | I |  | A | R |  |  |
| Deployment |  | I | I |  | A | R |  |  |
| **Milestone 3** |  |  |  |  |  |  |  |  |
| Test Cases | I | I | A |  | C |  |  | R |
| Product Testing | I | I | A |  | C | C |  | R |
| Programmer Guide | I | I | A |  | R | R | C |  |
| Deployment and Operations Guide | I | I | A | R |  |  |  |  |
| **Milestone 4** |  |  |  |  |  |  |  |  |
| User Guide | I | I | A | C | R |  |  |  |
| Test Report | I | I | A |  | C |  |  | R |
| Product Delivery | I | I | A |  | R |  |  |  |

*Note.* RACI Matrix created with reference from previous 2023 SWEN 670 AlphaSoft project (AlphaSoft, 2023).

# **11. Monitoring and Reporting**

### **11.1 Intra-Team Monitoring and Reporting**

* We will track and monitor tasks and updates within the group through a few different avenues that will ensure requirements are being met as we progress through the project, and all members of the team are adequately contributing, reporting, and tracking progress:
  + Weekly sprint reviews
  + Task tracking with GitHub Project
  + KPI dashboard for feature engagement, user activity

### **11.2 Project Communications**

#### 11.2.1 Internal Meetings

* Internal meetings will be held twice weekly, when available, to ensure that all internal stakeholders are kept up to date with changes.

#### 11.2.2 Internal-External Meetings

* + Meetings between internal and external stakeholders will aim to be held once a week, so that all progress updates, reviews, and requirement changes can be discussed and maintained.

#### 11.2.3 Cross-team Meetings

* + Cross-team communications will be had on a case-by-case basis as the need for such meetings arises.

# **12. Assumptions**

The following assumptions are made as part of the development of CareConnect.

* All users will utilize a modern smartphone that supports the latest standards.
* Initial deployment of the CareConnect app will only support the English language while support for multi-lingual access is deferred.
* Medication lists, tasks and documentation will be manually input it, the initial release of CareConnect will not support external sources and integration with third-party systems that explicitly manage medications. Third-part services integration does exist in other functional areas of CareConnect.
* Notifications will initially only cover app-specific notifications. Other types of notifications such as text and email will not have support at launch.
* Due to the nature of the product, the assumptions are made that users may have limited technical skills and the user interface, and the user experience should account for this.

# **13. Constraints**

The following constraints have been identified in the development of CareConnect:

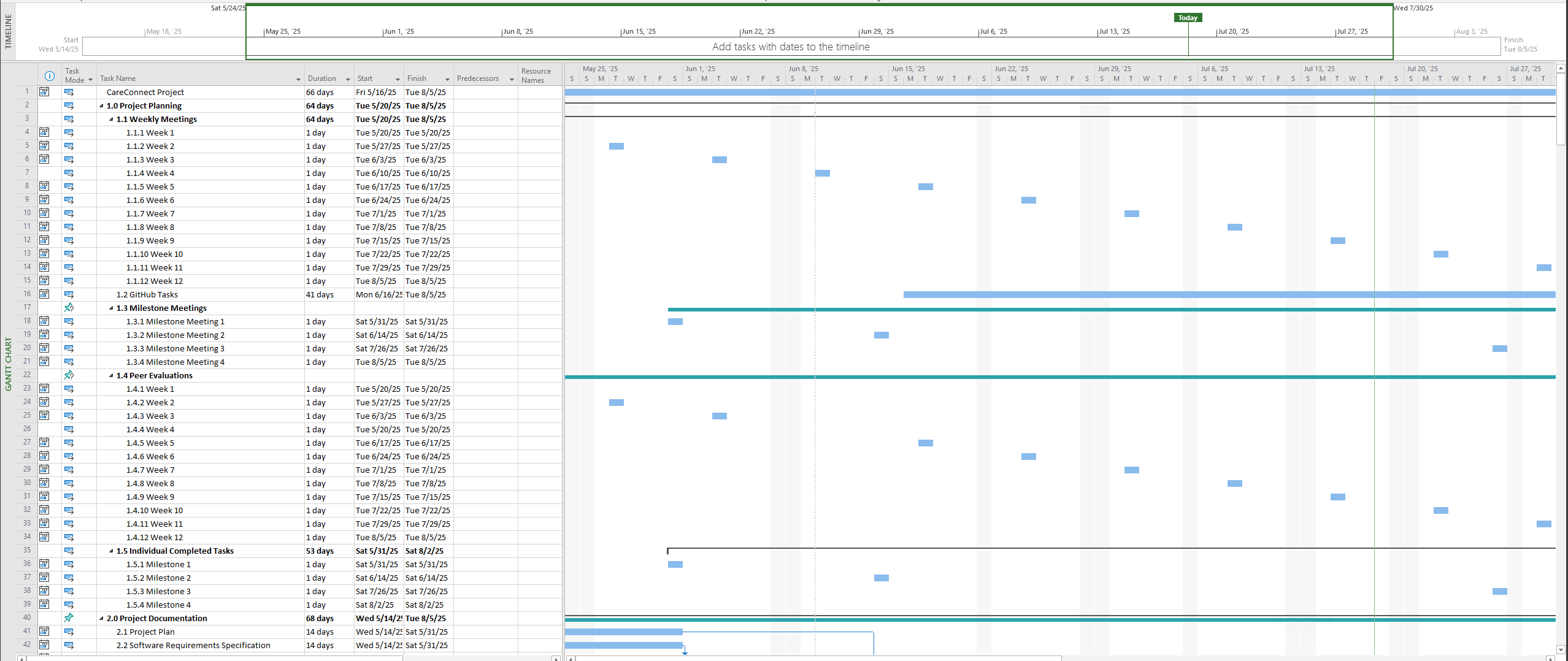
* The entire development cycle for a minimal viable product must be completed within a 12-week period. This includes a fully completed product. A beta version must be ready within a 4-week period.
* Only open-source and free tools must be leveraged. There is no budget to adopt any solutions that may incur a cost.
* A scale-to-zero and serverless architecture must be prioritized.
* Full regulatory compliance may not be initially met by the MVP as the verification process would be referred to an external division.
* CareConnect must be developed within the constraints of the chosen technology stack.
* Complete conformity to accessibility standards will not be achieved with the MVP for CareConnect. Improvements to this area are deferred to future updates.

# **14. Work Breakdown Structure (WBS)**

## 14.1 Structure

**Figure 4**

*Visual Overview WBS for CareConnect*



*Note.* WBS created with reference from previous 2023 SWEN 670 AlphaSoft project (AlphaSoft, 2023).

An external Microsoft Project file has been included as a supplement of the overall documents for a more detailed view of the CareConnect project WBS. The WBS is created with the timeline of the Gantt chart as one cohesive file.

# **15. 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:

## 15.1 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

## 15.2 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 16**

*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).

## 15.3 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.

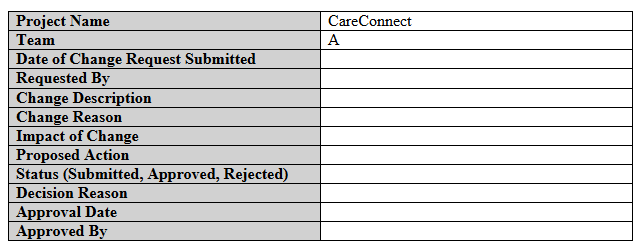
By following this structured yet adaptable change workflow, including initiation, review, and authorization, we will ensure the CareConnect testing remains aligned with project goals while minimizing disruption.

## 15.4 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 TemplateCareConnect Change Request Form Template](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=kW9Wxy). A figure has been provided below to illustrate what the template for the Change Request Form looks like for reference.

**Figure 5**

*CareConnect Change Request Form Template*

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

## 15.5 UI/UX wireframe preview

A detailed discussion of the CareConnect UI/UX wireframes can be found in the CareConnect Technical Design Document. The Technical Design Document is included in the overall group of documents delivered to the intended audience.

# **16. Appendices**

## 16.1 Glossary of Terms

**Table 16**

*Glossary Table*

| **Term** | **Definition** |
| --- | --- |
| CareConnect | |  | | --- | | This is the name of the project. It will be designed as an application to help caregivers and patients keep track of healthcare needs, such as appointments and health records. | |
| UI/UX -User Interface/User Experience | Aspects of the application that will define how users experience and interact with the application. |
| User | A person utilizing the application is either the patient, caregiver, or family member using the application. |
| API’s (Application Programming Interface) | Permits systems to communicate with one another. |
| Wireframes | A preview of a visualization of how a user interface will appear. |
| KPI (Key Performance Indicator) | Indicator that helps to gauge performance and progress. |
| Flutter/Dart | Dart is the programming language that Flutter uses. Flutter is used to be able to use a single code among different applications. |
| HIPAA (Health Insurance Portability and Accountability Act) | Laws governing the protection of patient information. |
| GDPR (General Data Protection Regulation) | European regulations governing privacy and data protection. |
| Sprints | Short period where software developers work on aspects of a project. |
| Elastic Container Registry | AWS storage for docker images. |

*Note.* This table shows a RACI matrix outlining who is Responsible, Accountable, Consulted, and Informed for key tasks.

## 16.2 References

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

Amazon Web Services. (2025, April). *AWS pricing catalog*. <https://aws.amazon.com/pricing/>

Apple Inc. (2025). *Apple Developer Program FAQ*. <https://developer.apple.com/support/programs/>

*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)

OpenAI. (2025). *ChatGPT (May 30 version)* [Large language model]. <https://chat.openai.com/>

PubNub. (n.d.). *Configuration: Authentication key (Dart SDK API Reference)*. <https://www.pubnub.com/docs/sdks/dart/api-reference/configuration#authentication-key>

Stripe, Inc. (2025). *Stripe pricing guide*. <https://stripe.com/pricing>

U.S. Bureau of Labor Statistics. (2023, September 6). *Computer and information systems managers*. Occupational Outlook Handbook. <https://www.bls.gov/ooh/management/computer-and-information-systems-managers.htm>

U.S. Bureau of Labor Statistics. (2023, September 6). *Computer programmers*. Occupational Outlook Handbook. <https://www.bls.gov/ooh/computer-and-information-technology/computer-programmers.htm>

U.S. Bureau of Labor Statistics. (2023, September 6). *Project management specialists*. Occupational Outlook Handbook. <https://www.bls.gov/ooh/business-and-financial/project-management-specialists.htm>

U.S. Bureau of Labor Statistics. (2023, September 6). *Software developers*. Occupational Outlook Handbook. <https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm>

U.S. Bureau of Labor Statistics. (2023, September 6). *Web developers*. Occupational Outlook Handbook. <https://www.bls.gov/ooh/computer-and-information-technology/web-developers.htm>

Zoom Video Communications. (n.d.). *OAuth for user authorized apps*. Zoom Developer Platform. <https://developers.zoom.us/docs/integrations/oauth/>