SOFTWARE REQUIREMENTS SPECIFICATION

WinterGreen

Version 2.0

Team CloudSmiths -

Jaimin Savaliya, Jaynesh Bhandari, Krunal Savaj, Nithin Murthy, Rajath Gokhale

2025-03-18

Team Member Document Review

Member Name	Signature or Initial	Reviewed on (Date)
Jaimin Savaliya	JS	17th March 2025
Jaynesh Bhandari	JB	17th March 2025
Krunal Savaj	KS	17th March 2025
Nithin Murthy	NM	17th March 2025
Rajath Gokhale	RG	17th March 2025

Table 1: Member Signature and Review Table

Document Revision History

Version	Changes	Date
1.0	First Draft	13th Feb 2025
1.1	Updated Document to reflect changes in final deliverable based on Client's needs.	23rd Feb 2025
2.0	Added more details regarding the deliverables and streamlined the document to clearly convey the project scope and overall requirements.	17th March 2025

Table 2: Document Revision History

Contents

Team Member Review and Sign-Off ii

Document Revision History iii

1 Introduction 1
- 1.1 Purpose . . . 1
- 1.2 Project Scope . . . 1
- 1.3 Definitions, Acronyms and Abbreviations . . . 1
- 1.4 References . . . 1
- 1.5 Overview . . . 1

2 Description 2
- 2.1 Product Perspective . . . 2
- 2.2 Product Functions . . . 2
- 2.3 User Characteristics . . . 2
- 2.4 General Constraints . . . 2
- 2.5 Assumptions and Dependencies . . . 2

3 Specific Requirements 3
- 3.1 Functional Requirements . . . 3
  - 3.1.1 Input and Storage of Healthcare Provider and EHR Details . . . 3
  - 3.1.2 Fetching Data from EHR Systems . . . 3
  - 3.1.3 Processing and Storing Fetched Data . . . 3
  - 3.1.4 Scheduled Data Refresh Every 30 Days . . . 3
  - 3.1.5 Logging and Error Handling . . . 4
- 3.2 Non-functional Requirements . . . 4
  - 3.2.1 Performance and Scalability . . . 4
  - 3.2.2 Security and Compliance . . . 4
  - 3.2.3 Data Integrity . . . 4
- 3.3 Design Constraints . . . 4

Appendix A: Use cases and requirements gathering work 5

Appendix B: Overall Flow Diagram 6

1 Introduction

1.1 Purpose

The purpose of this document is to clearly and comprehensively define the scope of the project, ensuring all stakeholders are aligned on what the software should do and how it should perform. This document offers a comprehensive overview of the problems we will address, the potential solutions, and the functions that these solutions will incorporate. The document also specifies the users of the solution.

1.2 Project Scope

Primarily, the scope pertains to the creation of a HealthCare provider on-boarding application that includes a front-end and a back-end. The front end provides WinterGreen an intuitive way to add new Healthcare providers, and the back-end automatically creates a pipeline to retrieve data from Healthcare providers' EHR systems(specifically Cerner, eCW and Athena health EHRs) and process it on AWS. The goal is to develop a HIPAA-compliant application that facilitates the on-boarding of healthcare providers for WinterGreen. The product will establish connections with healthcare providers' EHR system and then securely fetch and store data.

1.3 Definitions, Acronyms and Abbreviations

- **HealthCare Provider:** Clients of WinterGreen (Example - Clinics or Hospitals).
- **AWS:** Amazon Web Services, a suite of cloud computing services.
- **EHR:** Electronic Health Record, a digital version of a patient's medical history.
- **FHIR:** Fast Healthcare Interoperability Resources, a standard for electronic healthcare data exchange.
- **HIPAA:** Health Insurance Portability and Accountability Act, U.S. national standards for protecting the privacy and security of patient health information.
- **Cerner:** Cerner, an Electronic Health Record platform.
- **eCW:** eClinicalWorks, an Electronic Health Record platform.
- **AWS S3 Bucket:** Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance.
- **Healthlake:** AWS HealthLake is a HIPAA-eligible service enabling healthcare and life sciences (HCLS) companies to securely store and transform their data into a consistent and query-able fashion.
- **Amazon RDS:** Amazon RDS is a managed database service.

1.4 References

1. IEEE Std 830. "IEEE Guide for Software Requirements Specifications". In: *IEEE Std 830-1984* (1984), pp. 1–26. DOI: 10.1109/IEEESTD.1984.119205

1.5 Overview

This Software Requirements Specification (SRS) document outlines the functional and non-functional requirements for the Healthcare provider on-boarding software system. It serves as a comprehensive guide for developers, designers, and stakeholders to understand the scope and objectives of the system, ensuring that all requirements are properly documented and understood.

The document provides a detailed description of the system's intended capabilities, user interactions, performance goals, and constraints. It also includes the system constraints and any other factors that may influence development or deployment. The goal is to ensure that the developed software meets the specified requirements.

2 Description

2.1 Product Perspective

WinterGreen is a Maine-based rural healthcare advisory firm that helps hospital and clinic leaders achieve their goals through simple but impactful decisions. WinterGreen hopes to create a system that allows any healthcare provider to have a dashboard to visualize trends in data collected by them in a secure and meaningful way to help them make impactful decisions. To create data visualization, WinterGreen needs to use FHIR APIs to fetch data from healthcare providers' EHR software. After fetching the data, the data needs to be stored in AWS Healthlake for processing of data before querying it to get meaningful data that can be visualized.

Currently, WinterGreen has developed a manual code to fetch data from clients' EHRs but the code has several key limitations. The code contains hard-coded healthcare provider details, requiring it to be manually recreated each time a new provider is on-boarded and the code is specific to Cerner EHR which means any healthcare provider that does not use Cerner cannot be on-boarded. WinterGreen has also developed an in house Data visualization Web-app where the queried data can be taken as input.

The project is only focusing on fetching, storing and querying the data. Data visualization is out of scope the for this project.

2.2 Product Functions

- User should be able log on the system.
- User should be able to use add new Healthcare providers in the system.
- User should be able to fetch Healthcare providers data.
- System should be able to re-fetch the data once a month.
- System should be able to store the data on AWS Healthlake.

2.3 User Characteristics

- **Wintergreen Associate:** System administrators responsible for on-boarding Healthcare provider and system maintenance. Refer to the [Appendix A] for Use case diagram of the user.

2.4 General Constraints

- Data will be retrieved from EHR systems once per month.
- Data will be sourced from sandbox EHR environments during development.
- The product must comply with HIPAA regulations. (Out of Scope)
- The development process will exclusively use AWS services such as Lambda and Healthlake.

2.5 Assumptions and Dependencies

- Healthcare provider has provided their details including their EHR credentials to WinterGreen for on-boarding.
- User of the system is Wintergreen Associate and so does not require to re-login to verify their credentials.

3 Specific Requirements

3.1 Functional Requirements

3.1.1 Input and Storage of Healthcare Provider and EHR Details

Description : The system should allow authorized users to input healthcare provider location details along with their respective EHR system information (Cerner, eCW or Athena health). This data will be validated and stored in AWS RDS database for further processing.

Priority: High

Stimulus: An authorized user enters healthcare provider details along with the respective EHR information via the system's user interface.

Response: The system validates the input for completeness and correctness before storing it in the RDS database. If validation fails, an error message is returned with the appropriate details asking user to verify and re-enter the details.

3.1.2 Fetching Data from EHR Systems

Description : The system should fetch healthcare data from the specified EHR systems associated with each healthcare provider. This ensures accurate and up-to-date information is retrieved for processing.

Priority: High

Stimulus:

Case 1: The system initiates a request to fetch data from an EHR when a new entry is added in the System.

Case 2: When a scheduled refresh occurs.

Response: The system establishes a connection with the respective EHR system, retrieves the relevant data, stores it in AWS S3 bucket (refer to Appendix B: Overall Flow Diagram to see the overall Flow of data). If the EHR system is not supported, an error is logged, and no data is fetched.

3.1.3 Processing and Storing Fetched Data

Description : Once data is retrieved from an EHR system, it should be processed to ensure consistency. Processing includes structuring the data for further analysis.

Priority: High

Stimulus: New data is fetched from an EHR system.

Response: The system processes the data using AWS Healthlake, and then stores it back in AWS S3 bucket (refer to Appendix B: Overall Flow Diagram to see the overall Flow of data). If processing fails, an error is logged, and the issue is reported.

3.1.4 Scheduled Data Refresh Every 30 Days

Description : To ensure stored data remains up-to-date, the system shall automatically refresh data from the respective EHR system every 30 days. This refresh will overwrite outdated information with the latest available data.

Priority: Low

Stimulus: A scheduled task triggers a data refresh process every 30 days.

Response: The system fetches latest data from the EHR systems for all stored healthcare providers, processes it, and updates the S3 bucket. If a provider's EHR data is unavailable, an error is logged for further review.

3.1.5 Logging and Error Handling

Description: The system should maintain logs for all data retrieval, processing, and storage operations. Any failures, including validation errors, fetch failures, or processing issues, shall be logged and reported.

Priority: Medium

Stimulus: Any operation within the system (example - data submission, fetch request, processing task) encounters an error or completes successfully.

Response: The system logs details of the operation, including timestamps, status, and error details.

3.2 Non-functional Requirements

3.2.1 Performance and Scalability

- The system should be able to at least process 100 concurrent requests for data input, fetching and processing without affecting the performance.
- Data fetching from EHR should complete within 45 seconds.
- Provide clear and concise code documentation, making the code more maintainable.

3.2.2 Security and Compliance

- Authentication and authorization should be managed using AWS IAM roles and policies, ensuring that only authorized users and services can access sensitive data.
- Encrypt all data to maintain data access and control.

3.2.3 Data Integrity

- The system must validate all verifiable input data before storing it.
- Duplicate entries should be automatically detected and flagged, preventing data duplication.

3.3 Design Constraints

- The architecture of the system must be completely on AWS while utilizing services like HealthLake, Lambda, Athena, and QuickSight.
- As the System is completely on AWS, we should minimize resource utilization to avoid extra cost.

Appendix A: Use cases and requirements gathering work

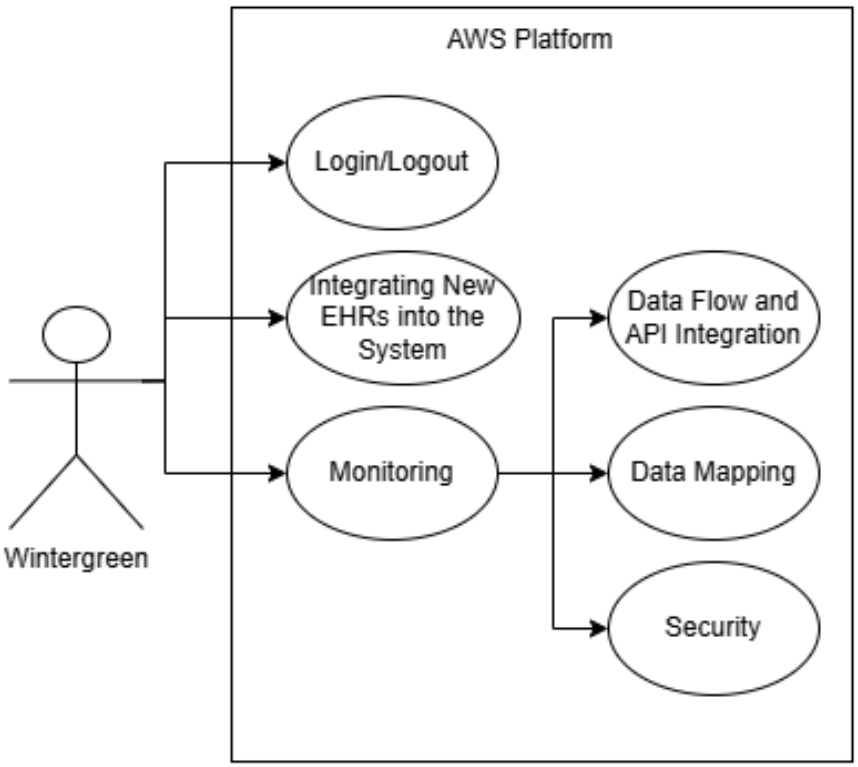

Figure 1: Use Case Diagram

Explanation: This use case diagram shows the interactions between the "AWS Platform" and "Wintergreen" (an administrator managing the AWS Platform). It essentially explains the platform's capabilities for the user. In a nutshell, it informs the user about the platform's characteristics. The features of the AWS Platform are displayed in the diagram. Features like platform monitoring, data flow and API administration, data mapping, user authentication, EHR integration, and security are all connected by utilizing AWS services. In conclusion, it illustrates how Wintergreen uses the AWS Platform to handle important system operations and health care administration activities.

Appendix B: Overall Flow Diagram

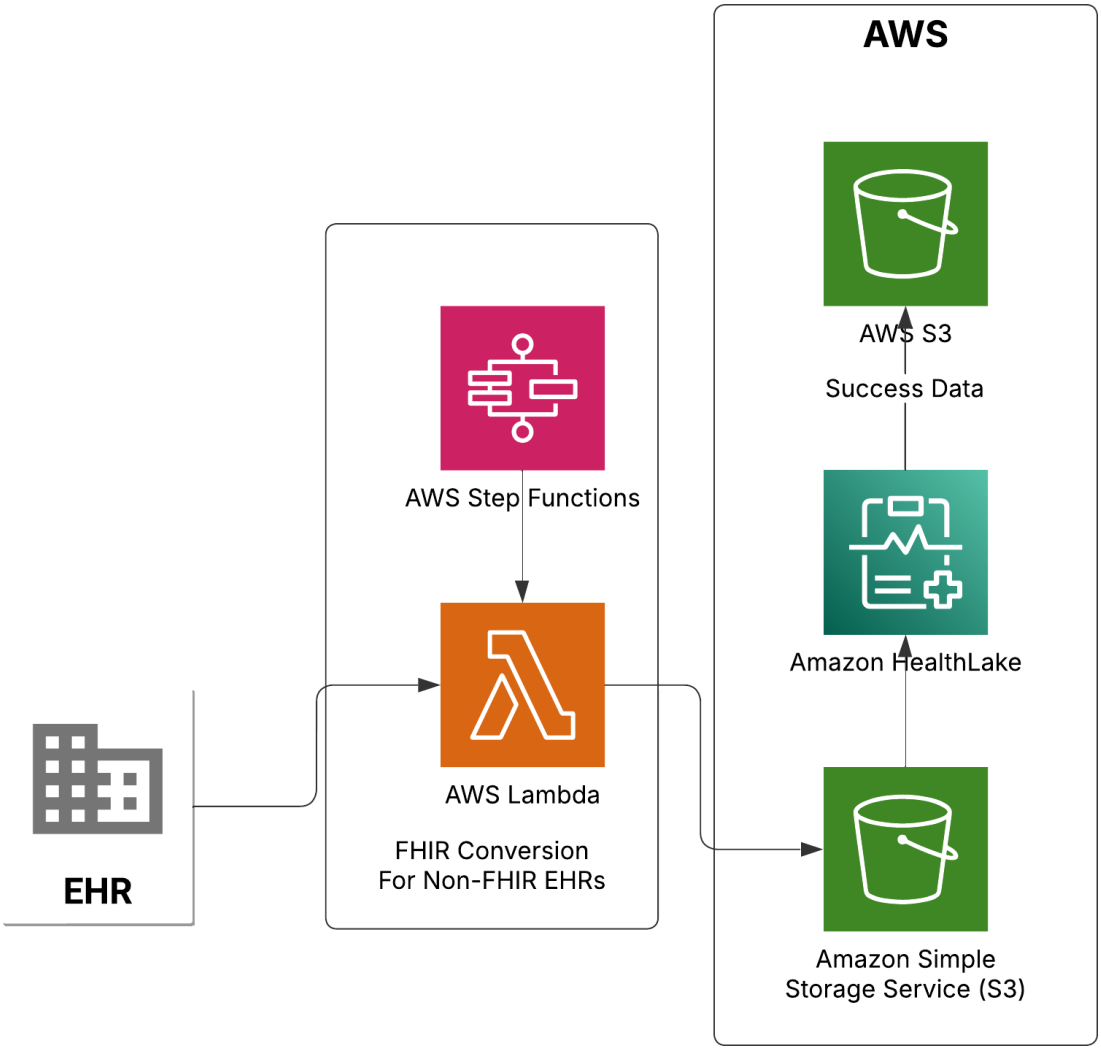

Figure 2: Overall flow Diagram

Explanation: The Overall Flow Diagram shares how the data fetched from EHR moves inside the System, As we can see, The AWS step functions call AWS lambda function which fetches data from EHR, this data is then Stored in AWS S3 bucket before being process by AWS healthlake, After processing, the data is again stored back in AWS S3 bucket.