Software Design Document (SDD)

for

SmartHealthVault

Version 1.0

Author: Jennica B

Table of Contents

[Table of Contents 2](#_Toc205366397)

[1. Introduction 3](#_Toc205366398)

[2. System Architecture (High-Level Design) 5](#_Toc205366399)

[3. Data Design 9](#_Toc205366400)

[4. Component-Level Design (Detailed Design) 11](#_Toc205366401)

[5. Interface Design 13](#_Toc205366402)

[6. Security Design 14](#_Toc205366403)

[7. Deployment and Operations 14](#_Toc205366404)

[8. External Integrations 14](#_Toc205366405)

[9. Appendices 15](#_Toc205366406)

# Introduction

## Purpose of the Document

This document details the high-level and low-level design of the SmartHealthVault application. It will serve as the primary technical guide for the development, quality assurance, and deployment teams.

## Scope of the Design

This design encompasses the system architecture, data design, interface design, and security protocols for the mobile applications (iOS/Android), the web portal, and all backend services.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Term / Acronym** | **Full Form / Meaning** |
| **API** | Application Programming Interface |
| **JWT** | JSON Web Token |
| **OAuth2** | OAuth 2.0 Authorization Framework |
| **FHIR** | Fast Healthcare Interoperability Resources |
| **HIPAA** | Health Insurance Portability and Accountability Act |
| **GDPR** | General Data Protection Regulation |
| **RBAC** | Role-Based Access Control |
| **REST** | Representational State Transfer |
| **AI Engine** | Artificial Intelligence Engine |
| **OCR** | Optical Character Recognition |
| **NLP** | Natural Language Processing |
| **WebRTC** | Web Real-Time Communication |
| **TLS** | Transport Layer Security |
| **AES-256** | Advanced Encryption Standard (256-bit) |
| **CRUD** | Create, Read, Update, Delete |
| **CI/CD** | Continuous Integration / Continuous Deployment |
| **S3** | Amazon Simple Storage Service |
| **SDK** | Software Development Kit |
| **Redis** | Remote Dictionary Server |

## References

* Software Requirements Specification (SRS) for SmartHealthVault v2.0
* FastAPI Endpoint Specification Document
* Relevant regulatory guidelines (HIPAA Security Rule, GDPR, IRDAI Telemedicine Guidelines).

## Document Overview

This Software Design Document (SDD) provides a comprehensive architectural and technical description of the SmartHealthVault system. The document is structured into multiple sections, each addressing a critical aspect of the system’s design.

# System Architecture (High-Level Design)

## Module Interaction Diagram

## Architectural Style

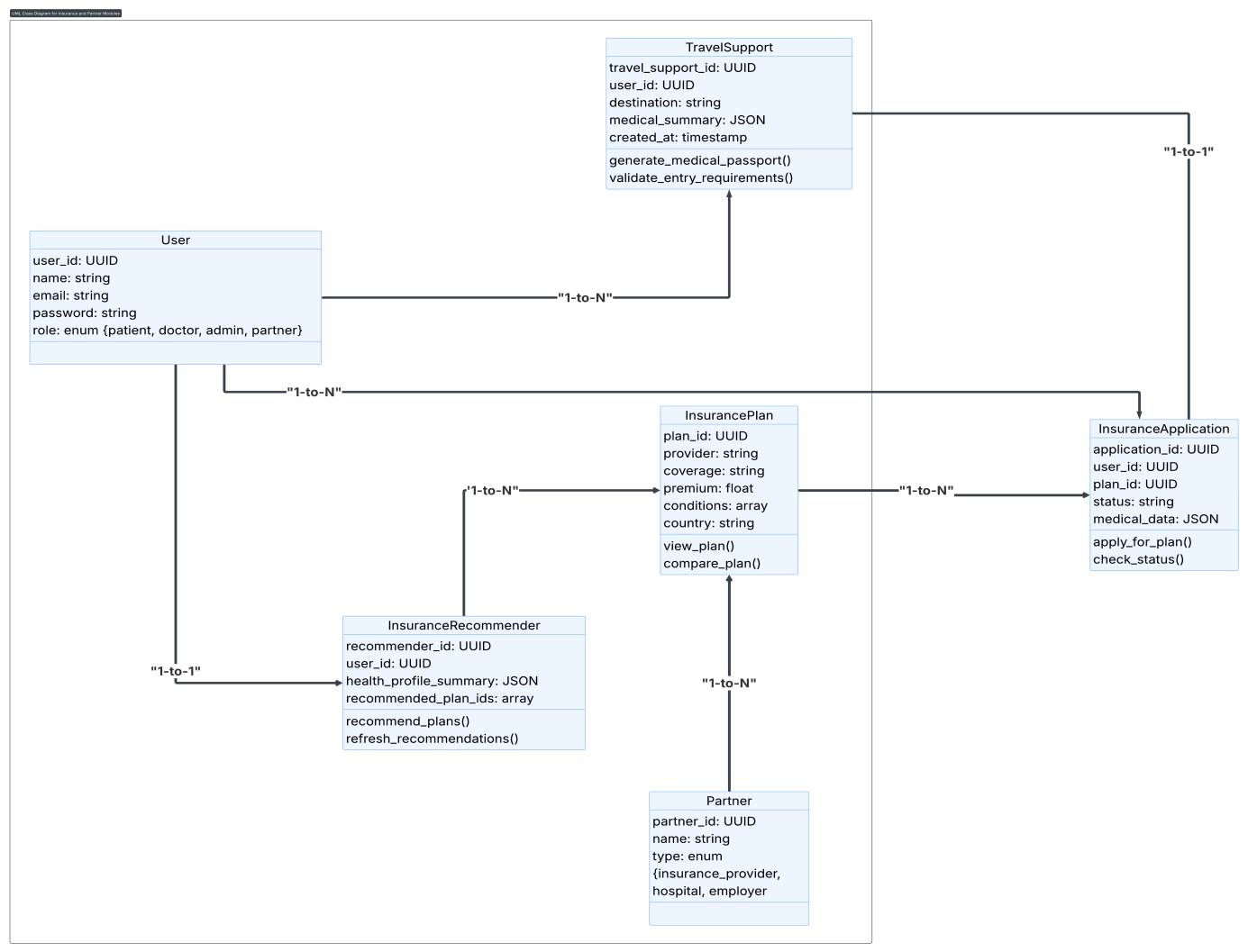
The system will be built on a **Microservices Architecture**. This choice supports scalability, independent deployment of services, and technology stack flexibility for different components (e.g., Python for AI, Node.js for real-time communication).

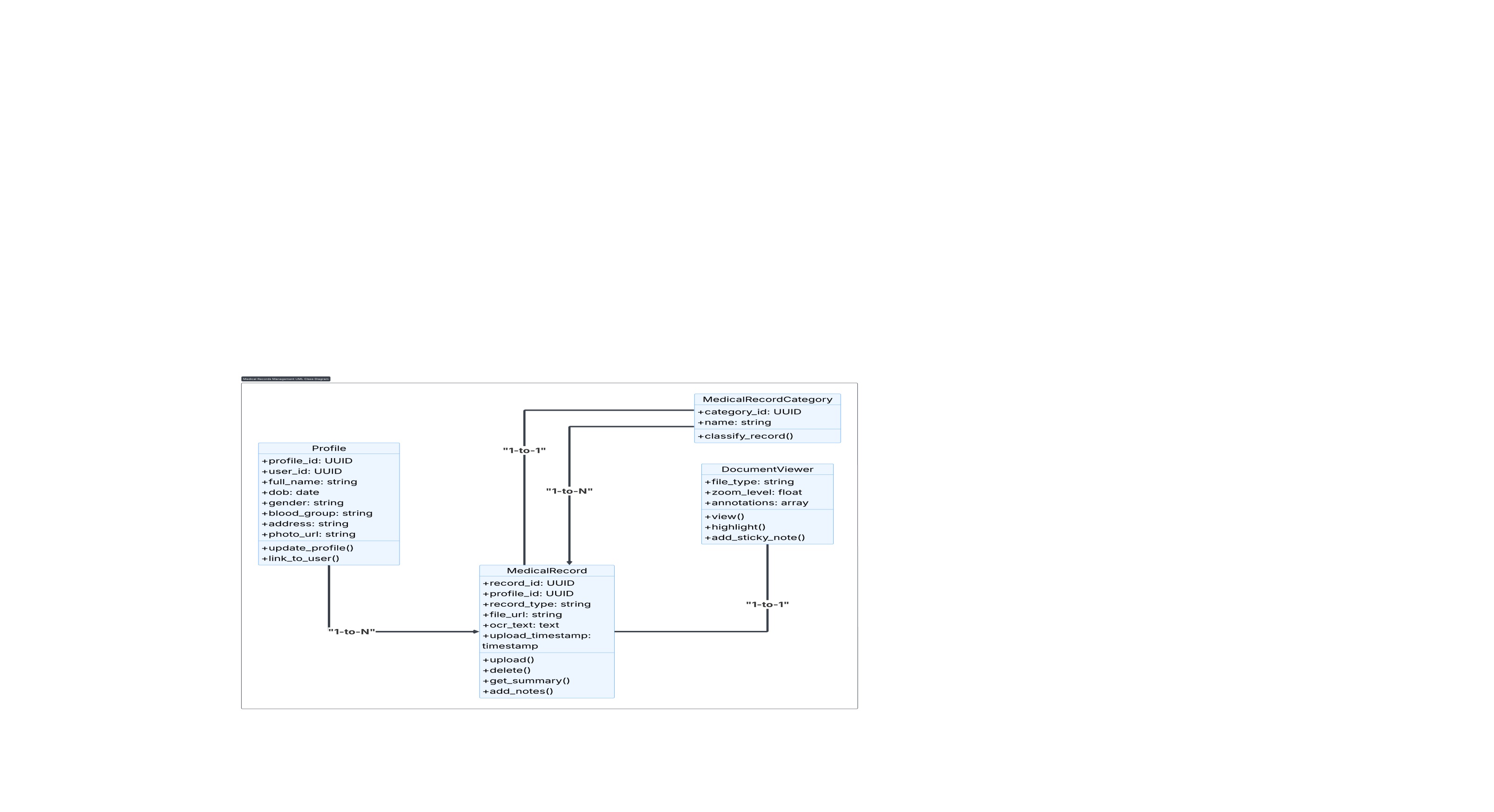
## System Components Diagram

A high-level diagram illustrating the major components:

* Clients: Mobile App (iOS/Android), Web Portal (for Doctors/Admins).
* API Gateway: A single entry point for all client requests, routing them to the appropriate microservice.
* Backend Microservices: User Service, Medical Records Service, AI/ML Service, Doctor & Telemedicine Service, Insurance Service, Notifications Service.
* Data Stores: Primary Database (e.g., PostgreSQL), Document/Blob Storage (e.g., AWS S3), Vector Database (for AI embeddings), Cache (e.g., Redis).
* Third-Party Integrations: Wearable APIs (Apple HealthKit, Google Fit), Insurance Provider APIs, Payment Gateways (Stripe, Razorpay).

Insurance & Partner Management



**P**rofiles and Medical Records

## 

## 

## 

## Technology Stack

* **Mobile App:** React Native or Flutter (for cross-platform compatibility).
* **Web Portal:** React or Angular.
* **Backend Microservices:** Python with FastAPI (chosen for performance and automatic API documentation).
* **AI/ML:** Python, TensorFlow/PyTorch, Scikit-learn, NLP libraries (spaCy, NLTK).
* **Databases:** AWS RDS (PostgreSQL) for structured data, AWS S3 for file storage.
* **Infrastructure:** Hosted on AWS/GCP/Azure, utilizing Docker for containerization and Kubernetes for orchestration.

## Design Goals and Constraints

* **Security & Compliance:** Design must be HIPAA and GDPR compliant. Zero-knowledge architecture will be an optional design consideration.
* **Scalability:** The architecture must handle over 1 million concurrent users and 50TB+ of data through auto-scaling.
* **Performance:** Adherence to strict performance metrics (e.g., <3s cold start, <1s on-device AI latency) is a primary design driver.
* **Availability:** Design for 99.9% uptime using multi-region deployment and robust disaster recovery plans.

# Data Design

## Key Tables:

* **Users Table**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| user\_id | UUID / INT |
| name | VARCHAR |
| email\_hash | VARCHAR |
| phone\_hash | VARCHAR |
| auth\_details | JSON / TEXT |
| role | ENUM |

* **Profiles Table**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| profile\_id | UUID / INT |
| user\_id | UUID / INT |
| full\_name | VARCHAR |
| dob | DATE |
| blood\_group | VARCHAR(3) |
| gender | VARCHAR |
| address | TEXT |
| photo\_url | VARCHAR |

* **Family\_Links Table**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| owner\_user\_id | UUID / INT |
| dependent\_profile\_id | UUID / INT |
| relationship | VARCHAR |

* **Medical\_Records Table**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| record\_id | UUID / INT |
| profile\_id | UUID / INT |
| record\_type | ENUM/VARCHAR |
| file\_url | VARCHAR |
| ocr\_text | TEXT |
| upload\_timestamp | TIMESTAMP |

* **AI\_Insights Table**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| insight\_id | UUID / INT |
| profile\_id | UUID / INT |
| risk\_type | VARCHAR |
| score | FLOAT |
| prediction\_date | DATE / TIMESTAMP |

* **Appointments Table**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| appointment\_id | UUID / INT |
| user\_id | UUID / INT |
| doctor\_id | UUID / INT |
| status | ENUM |
| time | TIMESTAMP |
| consultation\_type | ENUM |

## Data Dictionary

A comprehensive list of all data tables and their fields, including data type, constraints (e.g., NOT NULL), and description.

## Data Storage Design

* Medical Documents: All files (PDF, images) will be stored in a secure, encrypted cloud blob storage like AWS S3. Access will be managed through pre-signed URLs.
* Data Encryption: All data will be encrypted at rest using AES-256 and in transit using TLS 1.3.

## Backup and Disaster Recovery

* Automated daily backups of the database and file storage.
* Point-in-Time Recovery (PITR) will be enabled.
* Objective (RPO) of < 15 minutes.

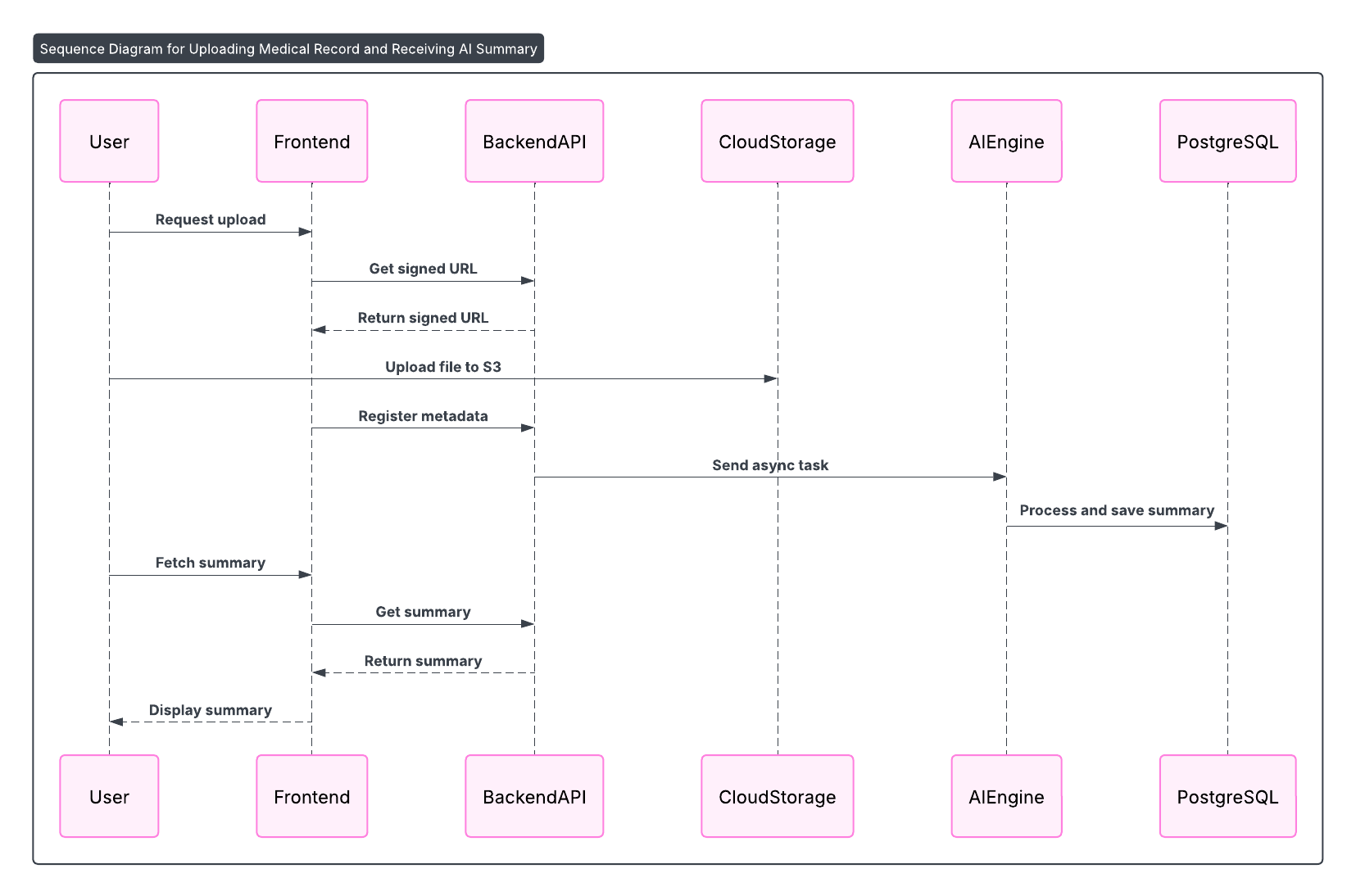
# Component-Level Design (Detailed Design)

## User Management & Authentication

* Sequence Diagram: A diagram illustrating the JWT-based login process, from credential submission to token generation and return.
* RBAC (Role-Based Access Control): A matrix defining permissions for each user role (Patient, Doctor, Admin, Guardian) against system actions.

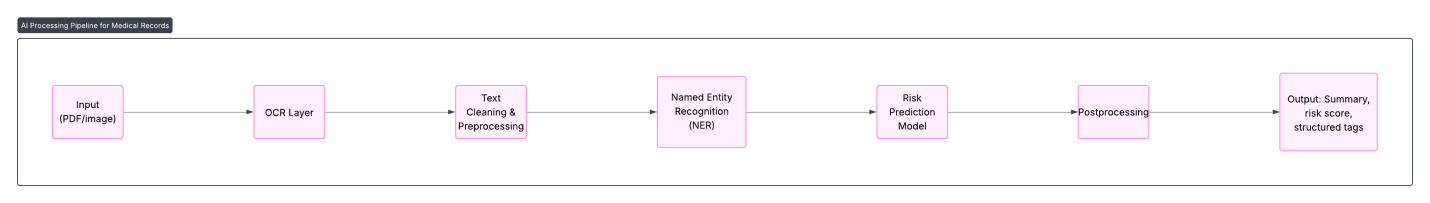
## Medical Record Upload & Processing

* **Sequence Diagram:** Illustrates the flow from a file upload request to OCR/NLP processing and metadata storage.
* **OCR & NLP Pipeline:**
  + File received at the /medical-record/upload endpoint.
  + A message is sent to a queue (e.g., RabbitMQ, AWS SQS) to trigger asynchronous processing.
  + A worker consumes the message, retrieves the file, performs OCR to extract text, and then uses NLP models to identify and categorize entities (doctor's name, medications, etc.).
  + Extracted data is saved to the database, linked to the original record.



## AI Health Risk Prediction Engine

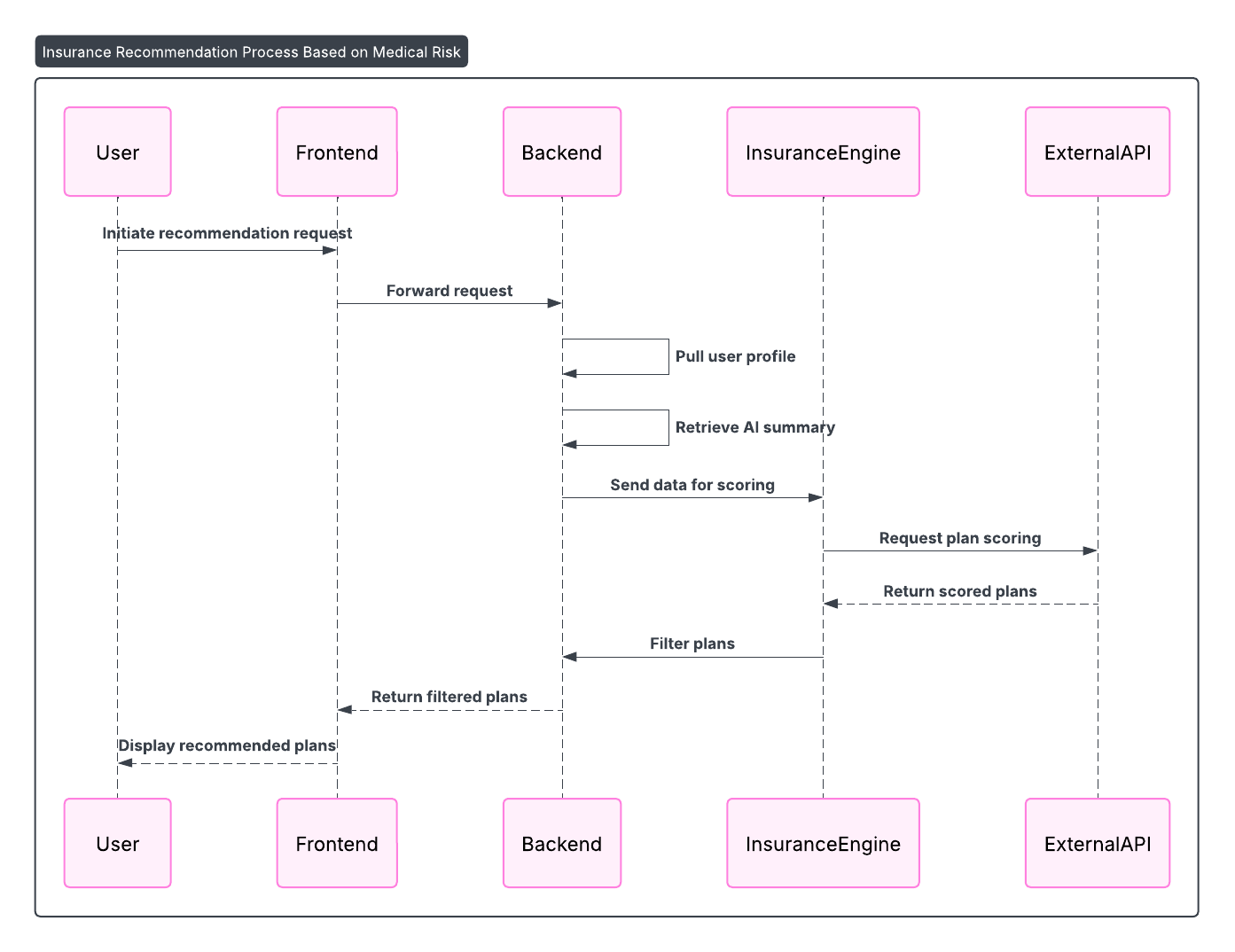
* **Model Architecture:** The design will specify the types of models used (e.g., Gradient Boosting Machines for risk prediction, BERT for NLP tasks).
* **Data Flow:** A diagram showing how anonymized user data (vitals, records, family history) is fed into the prediction models and how insights are generated and stored.
* **Explainable AI (XAI):** Design for generating explanations (e.g., using SHAP values) that identify the top factors (e.g., high cholesterol, age) contributing to a risk score. This will be exposed via the /ai/summary endpoint.



## Telemedicine and Video Consultation

* **SDK Integration:** The design will specify the use of a HIPAA-compliant third-party SDK (e.g., Twilio, Vonage) for implementing WebRTC-based video/audio calls to ensure security and reliability.
* **State Management:** The system will manage the state of the consultation (e.g., waiting, in-progress, completed, dropped) to handle events like billing and summary generation.

## Sequence Diagram – Insurance



# Interface Design

## User Interface (UI) Design

* **Guiding Principles:** The design will adhere to mobile-first, elder-friendly, and accessibility (WCAG 2.1) principles.
* **Component Library:** A common set of UI components (buttons, forms, cards) will be designed for consistent user experience across the app.
* **Wireframes:** Reference to the wireframe descriptions in Section 5 of the SRS.

## API Design

* **RESTful Principles:** All APIs will be designed to be stateless and follow REST conventions.
* **Authentication:** All protected endpoints will require a Bearer token in the Authorization header, as detailed in the FastAPI specification.
* **API Documentation:** The use of FastAPI will auto-generate interactive API documentation (Swagger UI/OpenAPI).
* **Versioning:** APIs will be versioned (e.g., /api/v1/...) to ensure backward compatibility.

# Security Design

## Authentication and Authorization

* Implementation of OAuth 2.0 with JWT for stateless session management.
* Two-Factor Authentication (2FA) will be enforced for sensitive operations.

## Data Privacy Controls

* **Consent Management:** A dedicated module will log and manage user consent for all data processing activities, particularly for AI predictions and data sharing.
* **Data Anonymization:** A process for de-identifying and anonymizing data for AI model training and analytics will be designed.

## Compliance Strategy

* **HIPAA:** Encrypt all PHI, implement strict access controls, maintain audit logs of data access, and sign Business Associate Agreements (BAAs) with all third-party services (like AWS).
* **GDPR:** Ensure the "Right to Erasure" by designing a service to permanently delete user data upon request. Provide clear visibility into data usage and obtain explicit consent.

## Audit Logging

* A dedicated logging service will record all significant events, including user logins, data access, record modifications, and changes in permissions.

# Deployment and Operations

## CI/CD Pipeline

A diagram illustrating the automated pipeline using tools like Jenkins or GitHub Actions for continuous integration, testing, and deployment.

## Infrastructure as Code (IaC)

Terraform or AWS CloudFormation will be used to define and manage the cloud infrastructure.

## Monitoring and Alerting

A monitoring stack (e.g., Prometheus/Grafana or AWS CloudWatch) will be used to track system health, performance metrics, and resource utilization. Alerts will be configured for critical failures and performance degradation.

# External Integrations

## Telemedicine SDK Integration (Video/Audio Calls)

* Connects with Fitbit, Apple Health, etc.
* Syncs metrics like heart rate, steps, and sleep.
* OAuth2-based token access; data stored in HealthData table.

## Insurance Provider API Integration

* Uses Google Translate or AWS Translate.
* Enables multilingual support for records, insights, and UI text.
* Translations triggered by user language preference or data upload.

## Wearable Device Integration

* Connects with Fitbit, Apple Health, etc.
* Syncs metrics like heart rate, steps, and sleep.
* OAuth2-based token access; data stored in HealthData table.

## Translation & Multilingual Support

* Uses Google Translate or AWS Translate.
* Enables multilingual support for records, insights, and UI text.
* Translations triggered by user language preference or data upload.

## Cloud Storage (Object Store for Medical Records)

* Stores user-uploaded records via pre-signed URLs.
* TLS in transit and AES-256 encryption at rest.
* Metadata indexed in the system database.

## Notification System

* Sends OTPs, alerts, appointment reminders.
* Supports SMS, email, and push notifications.
* Managed by backend notification controller.

# Appendices

## Glossary

|  |  |
| --- | --- |
| **Term** | **Description** |
| JWT | Token for user authentication |
| OCR | Text extraction from images |
| WebRTC | Real-time communication (video/audio) |
| RBAC | Role-based access control |
| SHAP | Explains AI model predictions |

## Sample Wireframes

Upload screen, teleconsultation interface, insights dashboard, and insurance page (Figma links or images if available).

## Common Error Scenarios

|  |  |
| --- | --- |
| **Case** | **Handling** |
| Upload fails | Retry prompt, log error |
| API timeout | Backoff retry, user notified |
| Consent revoked | Access denied, audit logged |

## Assumptions

* Users have access to valid contact info.
* Uploaded records < 20 MB and are PDF/JPEG.
* AI engine may take up to 30s per inference.

## References

* OpenAPI docs (internal)
* External APIs: Twilio, AWS, Google Translate
* Compliance: HIPAA, GDPR guidelines