BSCS FINAL PROJECT
Design and Test Specification

S1 & S2 Heart Sound Classification using Deep Learning

Project Advisor
Dr. Rabia Tahseen

Presented by:
Group ID: S25BS014

Student Reg#	Student Name
L1S22BSCS0070	**Ahmed Mustafa**
L1S22BSCS0212	**Ahsen Ahmed**
L1S22BSCS0188	**Abdullah Manzoor**
L1S22BSCS0220	**Hafiz Muhammad Haris**

Faculty of Information Technology & Computer Science

University of Central Punjab

Design and Test Specification

SDP Phase III

S1 and S2 Heart Sound Classification using Deep Learning

Advisor: Dr. Rabia Tahseen

Team S25BS014

Member Name	Primary Responsibility
Ahmed Mustafa	Documentation, Deep Learning Model Development & Training, PCG Signal Preprocessing
Ahsen Ahmed	Web Development, Database, Cloud Deployment
Abdullah Manzoor	Web Design Specification & Development
Hafiz Muhammad Haris	Web interface development and Database.

Table of Contents

Table of Contents i
Revision History ii
Previous Phases Feedback ii
Abstract 1
1. Introduction 1
1.1 Product 1
1.2 Background 1
1.3 Objective(s)/Aim(s)/Target(s) 2
1.4 Scope 2
1.5 Business Goals 2
1.6 Document Conventions 2
1.7 Miscellaneous 3
2. Technical Architecture 4
2.1 Application and Data Architecture 5
2.2 Component Interactions and Collaborations 15
2.3 Design Reuse and Design Patterns 15
2.4 Technology Architecture 20
2.5 Architecture Evaluation 20
3. Detailed/Component Design 24
3.1 Component-Component Interface 24
3.2 Component-External Entities Interface 27
3.3 Component-Human Interface 30
4. Screenshots/Prototype 31
4.1 Workflow 31
4.2 Screens 35
4.3 Additional Information 39
5. Other Design Details 39
6. Test Specification and Results 40
6.1 Test Case Specification 42
6.2 Summary of Test Results 44
7. Revised Project Plan 47
8. References 49
Appendix A: Glossary 50
Appendix B: IV & V Report 50

Revision History

Name	Date	Reason For Changes	Version
N/A	N/A	N/A	N/A
N/A	N/A	N/A	N/A

Previous Phases Feedback

Idea Defense Feedback (Screenshot)

S25 Idea Defence Feedback + Proposal Submission Deadline Summarize

PO ProjectOffice Reply Reply all Forward

To: L1S22BSCS0070 - AHMED MUSTAFA; L1S22BSCS0220 - HAFIZ MUHAMMAD HARIS; **+2 others** Tue 1/21/2025 4:20 PM

Cc: Dr Adnan Ghafoor; Dr Rabia Tehseen

Dear Students,

Following is the feedback regarding your Idea Defence:

Sr. #	**CS-69**
Project Title	**S1 & S2 HEART SIGNAL CLASSIFICATION USING DEEP LEARNING**

Good work.

Group must know whole project. Only one student knows.

Prepare your Proposal Document in the light of comments provided. Last date to submit the Proposal Document is **Thursday, January 30, 2025** not later than **03:00 PM**. The proposal document should be prepared using the template available at http://smrms.ucp.edu.pk/All_Phases_Templates.html. Submit the hard copy to Project Office in F-304 (Main Building).

When submitting the hard copy of the proposal document, use the duly signed Group Allocation Yellow Page (available at http://smrms.ucp.edu.pk/Forms.html) as the first page of the submission document. Also get the proposal tape bind before submission.

Phase 2 (SDS) Feedback (Screenshot)

S25- FYP: Phase 2 (SDS) Feedback Summarize

PO ProjectOffice Reply Reply all Forward

To: L1S22BSCS0070 - AHMED MUSTAFA; L1S22BSCS0220 - HAFIZ MUHAMMAD HARIS; **+2 others** Fri 10/17/2025 11:10 AM

Cc: Dr Rabia Tehseen

Dear Students,

Following is the feedback regarding your phase 2 (SDS):

S25BS014	**S1 & S2 HEART SIGNAL CLASSIFICATION USING DEEP LEARNING**

Some features marked as entity. Minor changes were suggested.

Convention of USD is incorrect. ERD needs revision. Minor changes were suggested from UML.

Your next milestone is phase 3 (i.e. DTS). Prepare your phase 3 (DTS) document in the light of feedback provided.

Abstract

This project aims to develop a comprehensive web-based system utilizing deep learning to accurately classify the first (S1) and second (S2) heart sounds from phonocardiogram (PCG) recordings. Heart sound classification plays a critical role in diagnosing cardiovascular issues, including valve defects, arrhythmias, and murmurs. By automating the analysis process, this system seeks to minimize human error and enhance diagnostic accuracy. The proposed tool will allow users to upload PCG recordings through an easy-to-use interface, generating detailed outputs such as visualizations, timestamps, and diagnostic interpretations. Advanced signal processing techniques and a hybrid deep learning model combining Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks will be employed to ensure robust and real-time performance.

1. Introduction

Heart sounds are essential diagnostic indicators that provide critical information about cardiac health. The two main sounds, S1 and S2, represent the closing of the heart's valves during the cardiac cycle. Traditional methods for analyzing these sounds rely on manual interpretation by medical professionals, which can be time-consuming and prone to human error. Moreover, such methods often struggle with noisy data or subtle abnormalities, limiting their diagnostic utility. This project addresses these challenges by introducing a deep learning-based solution for the automated classification of heart sounds. By integrating state of-the-art signal processing and machine learning techniques, the proposed system aims to provide an accurate and efficient diagnostic tool for doctors and researchers.

Recent advancements in deep learning have revolutionized signal processing. CNNs effectively extract spatial features from visual representations of sound, such as spectrograms, while LSTMs specialize in temporal sequence analysis. Combining these models offers unparalleled accuracy and robustness in analyzing complex signals. This project harnesses these capabilities to develop a user-friendly, scalable system that integrates a web-based interface for real-time analysis and visualization of heart sounds.

1.1 Product

The system aims to address a significant gap in the current healthcare and research landscape. The manual interpretation of heart sounds is labor-intensive, error-prone, and often inconsistent, particularly when dealing with noisy or complex data. Existing tools for heart sound analysis are limited in their ability to provide real-time results and user-friendly interfaces, hindering their widespread adoption in clinical and research settings. There is a need for an automated system that can reliably classify S1 and S2 heart sounds while offering accessible and interactive outputs. Moreover, these methods struggle to handle variability in real-world data and lack scalability, limiting their applicability for large-scale medical or research usage.

Additionally, accessible tools for analyzing PCG recordings in real-time are significantly lacking. Existing systems typically do not provide intuitive user interfaces, hindering their adoption by Doctors and Researchers. This lack of usability reduces the overall medical and research utility of heart sound analysis tools, delaying diagnosis and limiting their impact in the healthcare sector

1.2 Background

Heart sounds are a primary indicator of cardiac health and are used to detect various conditions including valve defects, murmurs, and arrhythmias. The two primary sounds, S1 and S2, represent valve closures during different phases of the cardiac cycle. Traditional auscultation requires trained cardiologists and offers limited reproducibility. With advancements in machine learning and signal processing, particularly deep learning architectures like CNN which are great at capturing spatial features and LSTM which are ideal for sequence data, it is now feasible to automate this classification. This project takes advantage of these advancements by combining CNN and LSTM architectures to automatically classify S1 and S2 heart sounds from PCG recordings, thereby enhancing accuracy and enabling scalable, real-time analysis through a web-based platform.

1.3 Objective(s)/Aim(s)/Target(s)

The primary objectives of this project are:

- To design and implement a hybrid CNN-LSTM model for classifying S1 and S2 heart sounds with high accuracy.
- To develop a web-based application that allows users to upload PCG recordings and view classification results interactively.
- To ensure the system performs well in noisy environments by employing advanced preprocessing techniques.
- To provide Doctors and researchers with meaningful insights through diagnostic visualizations and summaries

1.4 Scope

This project aims to provide a user-friendly, intelligent web platform that supports doctors, researchers, and patients in analyzing heart sound recordings such as PCG files. The core functionality revolves around enabling users to upload their PCG recordings, which the system then processes using advanced preprocessing techniques such as, filtering, MFCC extraction, and spectrogram generation to clean and structure the data. Once preprocessed, a hybrid CNN-LSTM deep learning model classifies the audio into S1 and S2 heart sound components.

Users can then view the classification results in the form of visual waveforms, time-based markers, and diagnostic summaries. The platform is designed to be highly interactive and accessible, with separate user roles such as admin, doctors, researchers, and patients that allow different levels of access and control. The project includes not only model development and training but also the integration of frontend and backend systems, deployment via Docker, and hosting on a cloud environment such as AWS or Azure.

While the system does not intend to replace a trained cardiologist's judgment, it serves as a powerful diagnostic support tool by providing quick, repeatable, and objective analysis of heart sounds, especially in scenarios where immediate professional diagnosis is not feasible.

1.5 Business Goals

The following are the business goals that can we implemented based on the completion of the project:

- Enhance the accuracy and speed of cardiac diagnostics in clinical and remote settings.
- Provide researchers with a platform to test and compare new models or datasets.
- Facilitate early detection of cardiovascular anomalies for patients in under-resourced areas.
- Improve the diagnostic workflow in hospitals and research labs.

1.6 Document Conventions

- **Font size:** 12
- **Font:** Times New Roman
- **Main Heading size:** 22
- **Sub-heading size:** 14
- **Text:** Justified

Note: Important points are **bold**

1.7 Miscellaneous

To ensure the smooth execution and management of the project, the following practices and measures will be adopted:

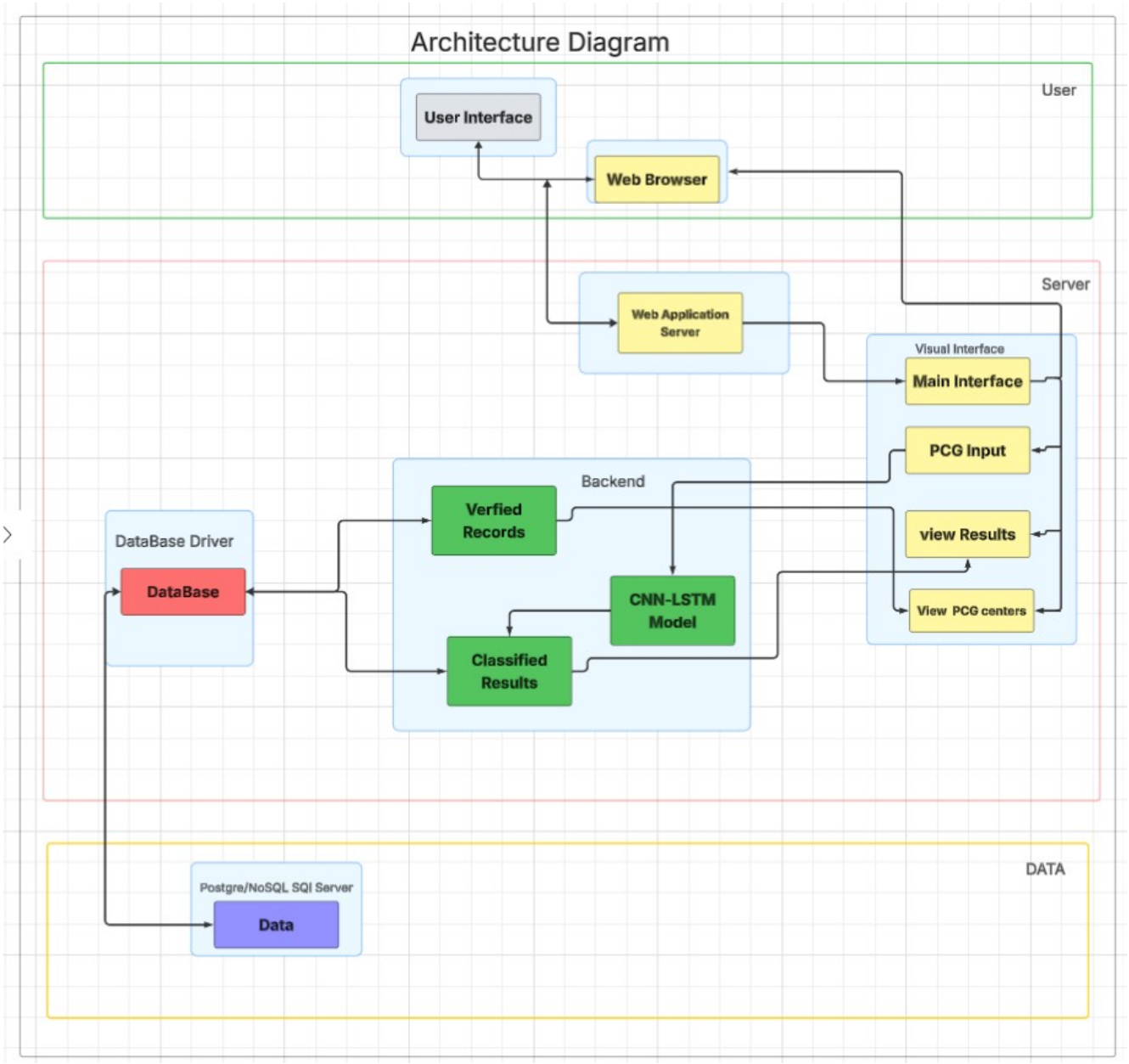
- **Regular Team Meetings:** Weekly progress meetings will be conducted to review the current status, set short-term goals, and address any blockers faced by team members. This promotes better coordination and accountability.
- **Progress Tracking:** Tools like Trello or GitHub Projects will be used to manage tasks, track progress, and assign responsibilities to ensure transparency throughout the development lifecycle.
- **Version Control:** Git will be used to manage code repositories, enabling collaborative development, rollback capability, and change history tracking.
- **Advisor Feedback:** Continuous feedback from the project advisor and relevant stakeholders (e.g., doctors, researchers) will be incorporated to improve model accuracy and system usability.
- **Documentation Maintenance:** Detailed documentation of APIs, model structure, and deployment steps will be maintained in a shared repository for future reference and maintainability.
- **Risk Mitigation:** Potential risks such as data availability issues, model performance bottlenecks, or integration delays will be identified early. Contingency plans, such as fallback datasets and modular architecture, will help handle unforeseen challenges.
- **Ethical Considerations:** The project will adhere to data privacy and security best practices. Sensitive user data and PCG recordings will be anonymized and securely handled in compliance with ethical guidelines.
- **Future Enhancements:** The system will be designed with scalability in mind, allowing for future improvements like additional heart sound classifications (e.g., murmurs), mobile app integration, or multilingual support.

2. Technical Architecture

The current system is a **custom-built, web-based diagnostic support application** that classifies S1 and S2 heart sounds from PCG recordings using deep learning. It is based on a **3-tier architecture** comprising:

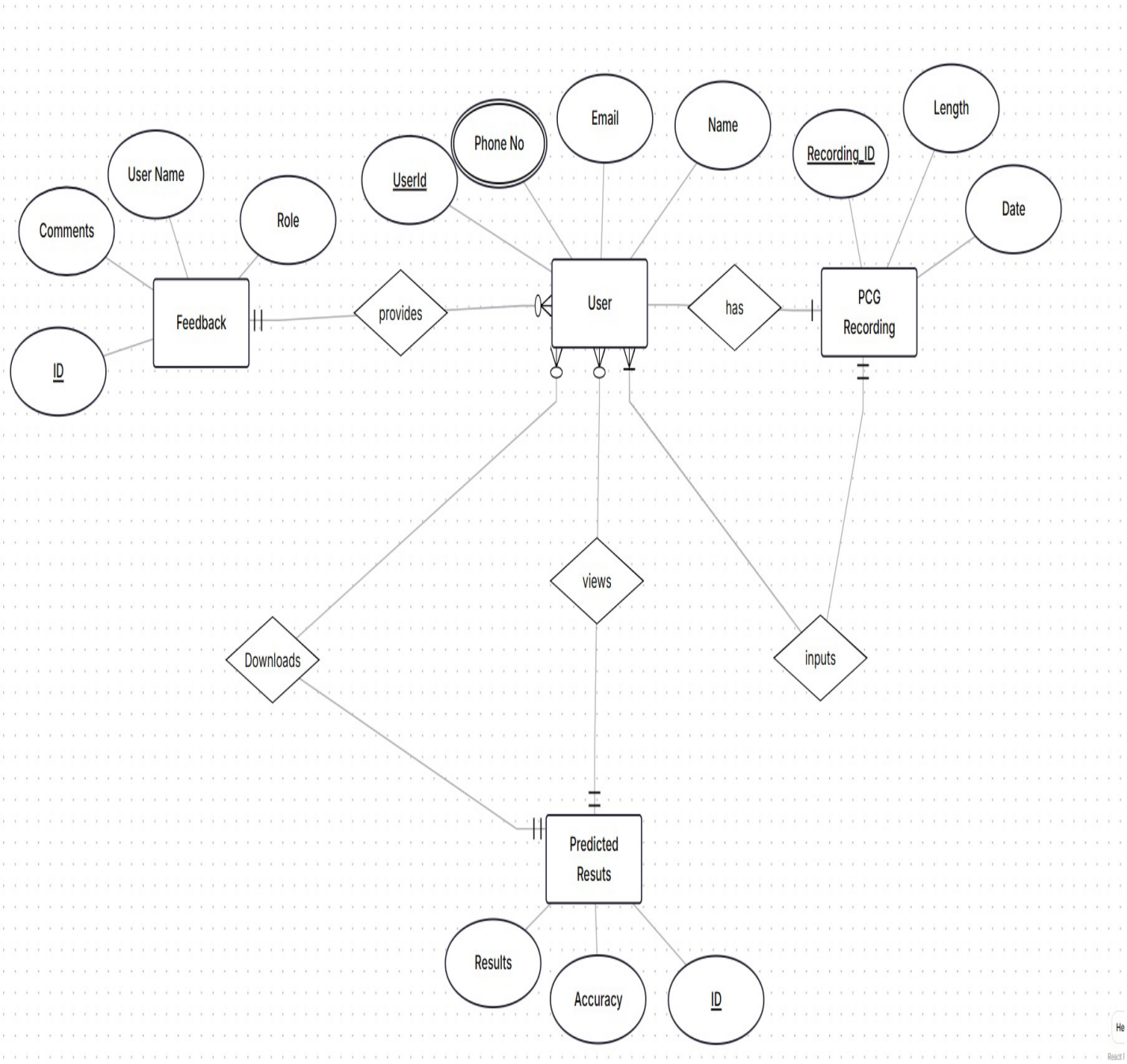
1. **Presentation Tier (User Tier)**
2. **Application Tier (Server Tier)**
3. **Data Tier (Database + ML Storage)**

This layered approach ensures modularity, separation of concerns, and scalability for real-time medical applications

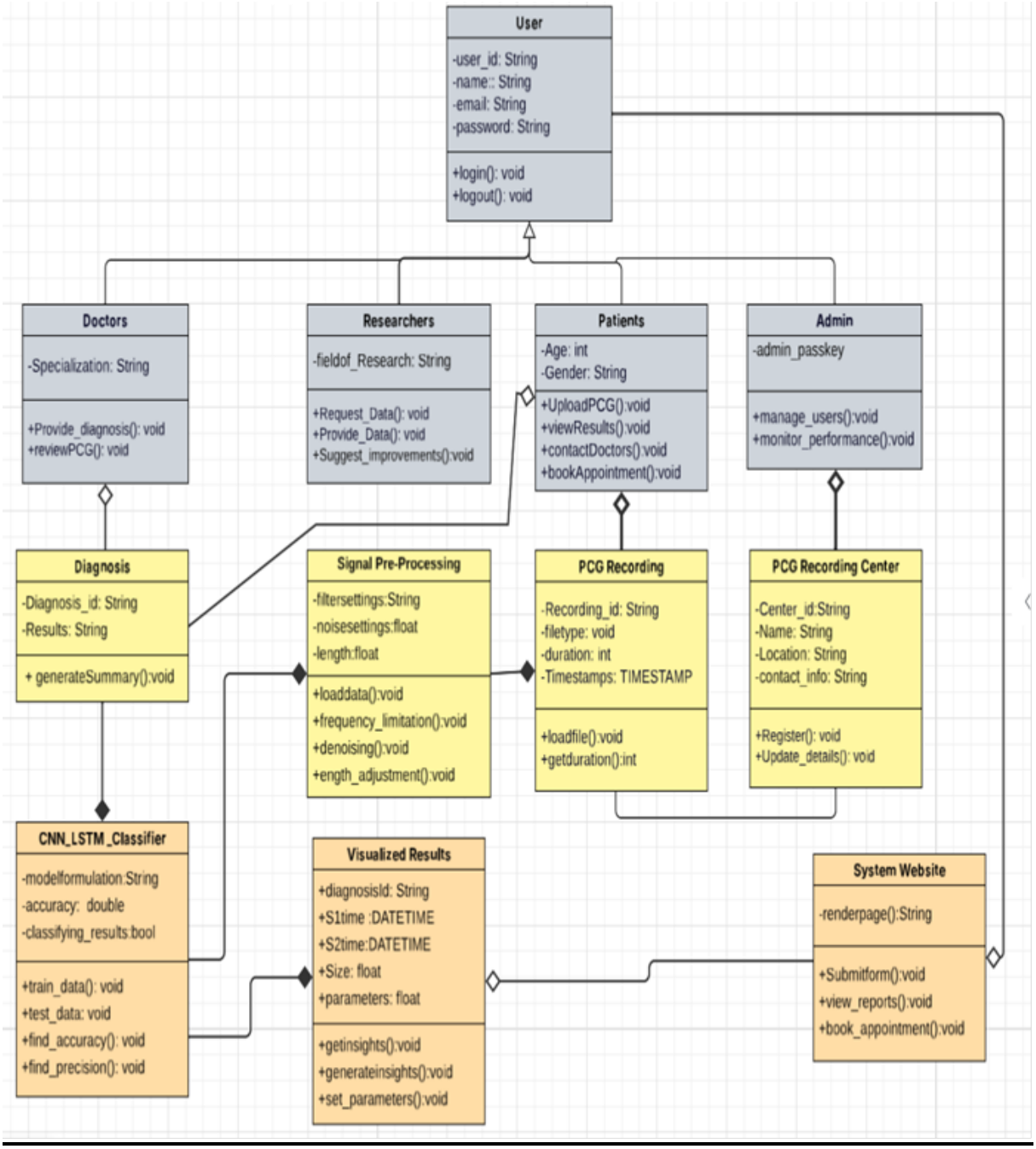


2.1 Application and Data Architecture

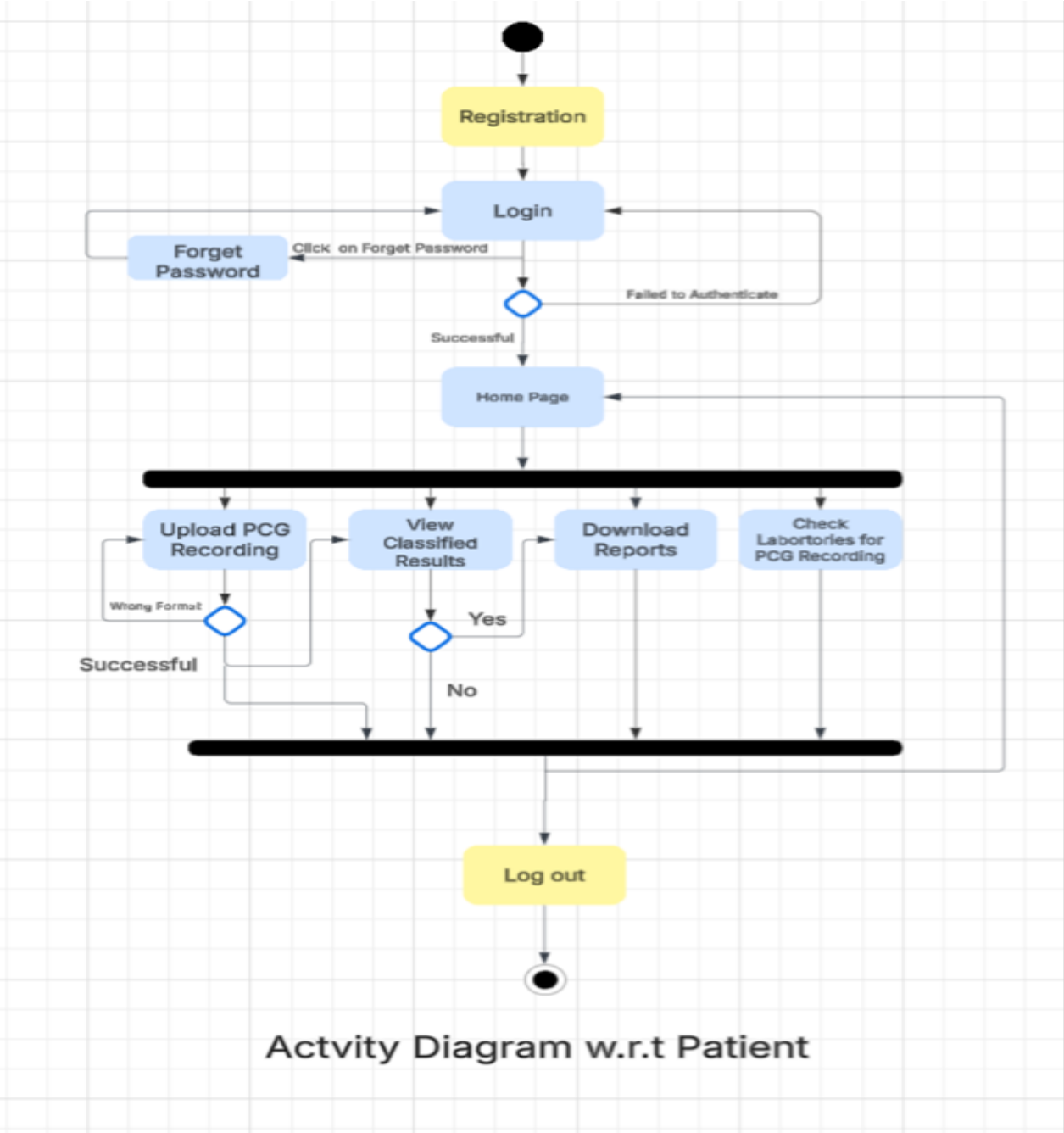
- **ERD Diagram:**



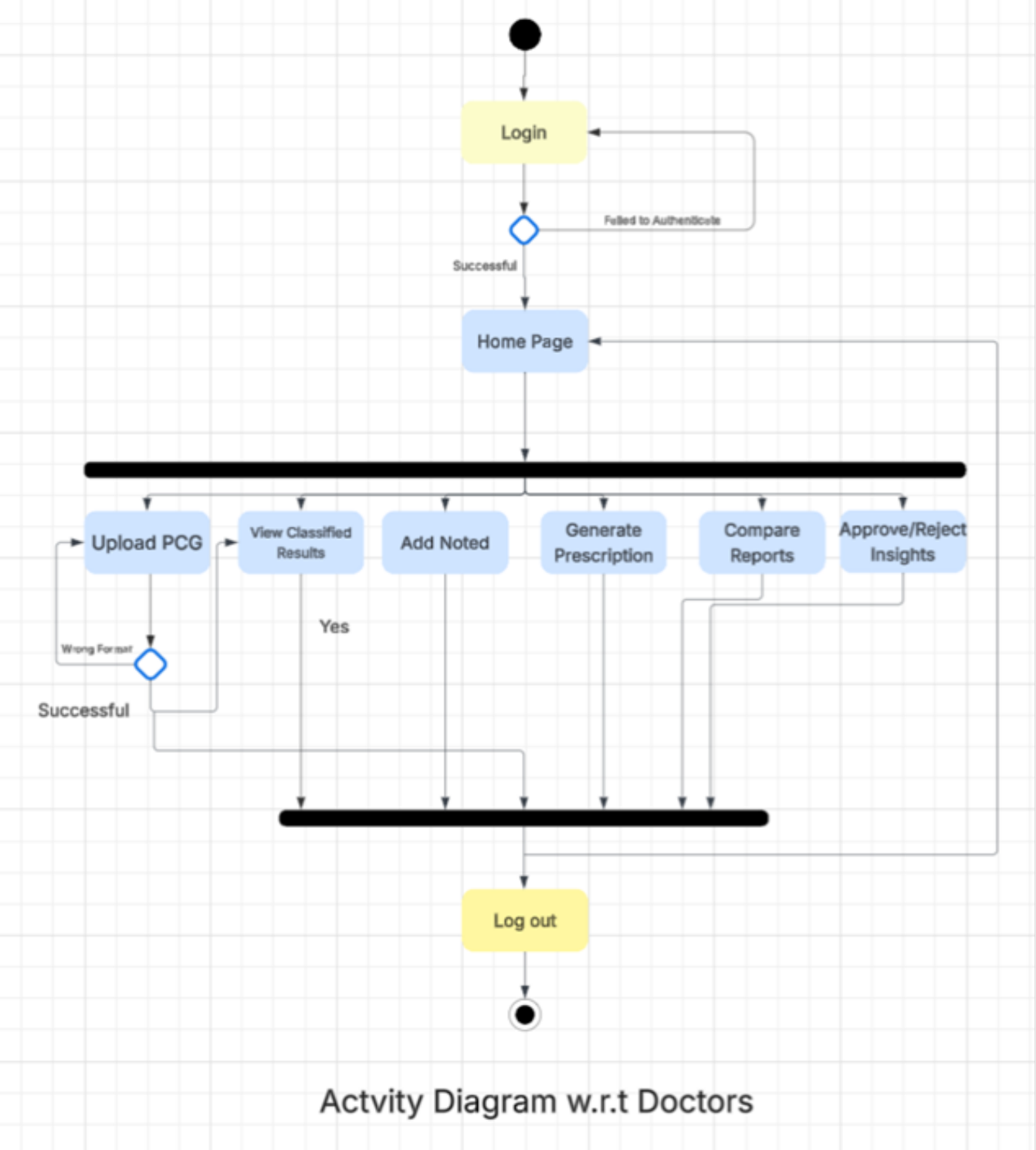
- **Class Diagram:**



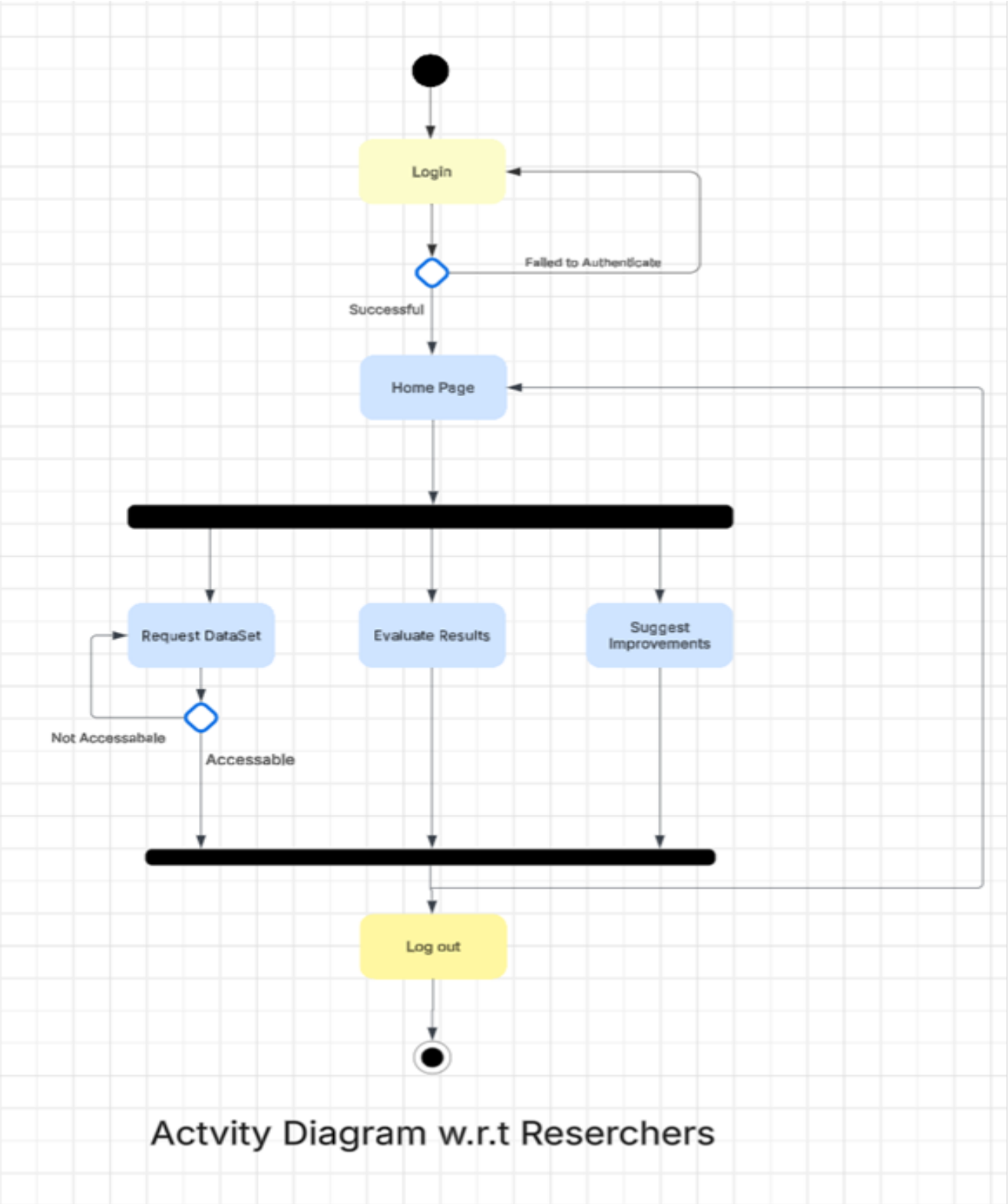
- **Activity Diagram**
  - **Activity Diagram w.r.t Patient:**



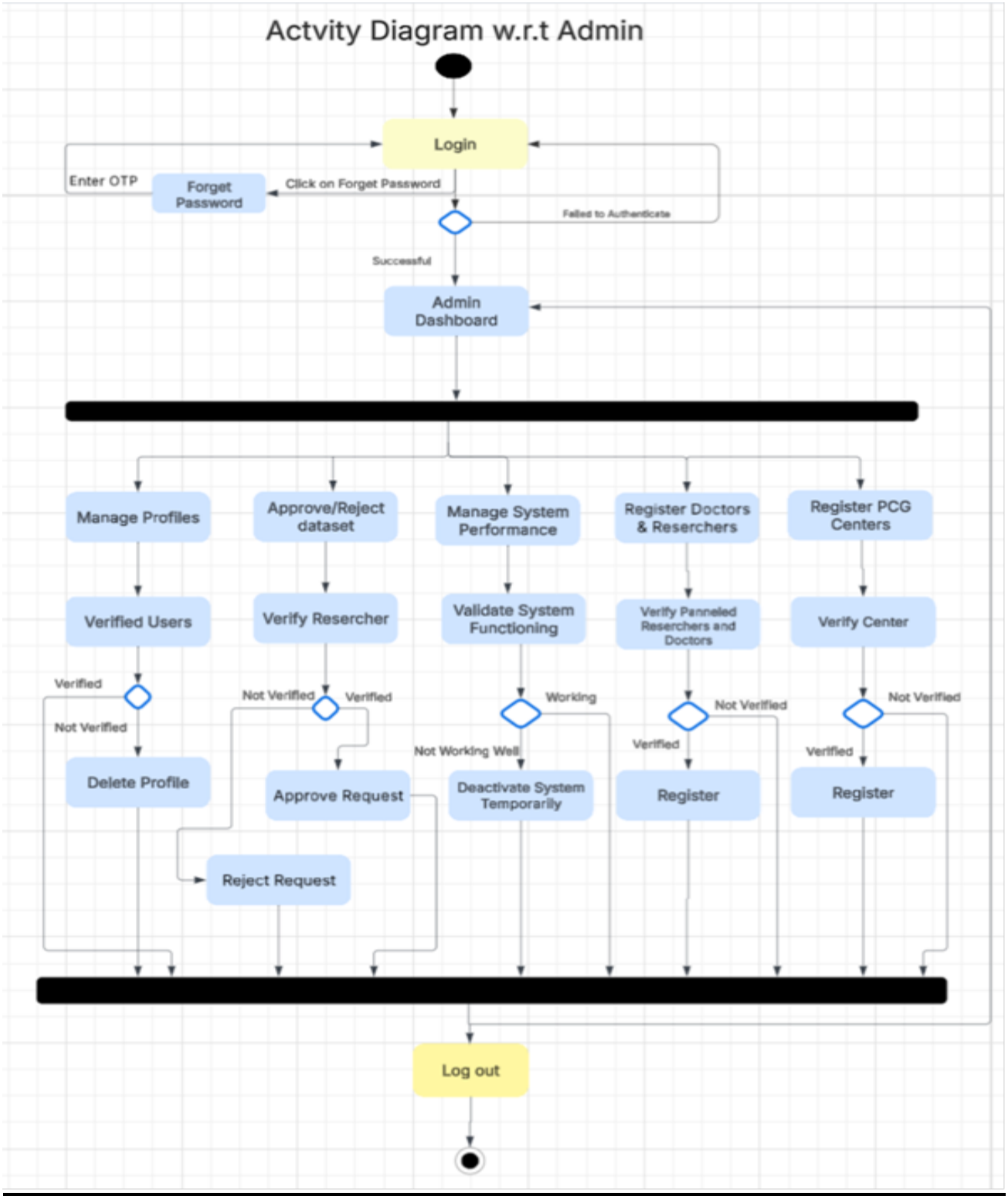
- **Activity Diagram w.r.t Doctors:**



- **Activity Diagram w.r.t Researchers:**



- **Activity Diagram w.r.t Admin:**



- **Decision Tables:**

Decision table for login

Condition	**Rules**			
Username	**T**	**T**	**F**	**F**
Password	**T**	**F**	**T**	**F**
Action				
Login Successful	✓			
Login Failed		✗	✗	✗

Decision table for Signup

Condition	**Rules**															
Email	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**F**	**F**	**F**	**F**	**F**	**F**	**F**	**F**
Age	**T**	**T**	**T**	**T**	**F**	**F**	**F**	**F**	**T**	**T**	**T**	**T**	**F**	**F**	**F**	**F**
Phone	**T**	**T**	**F**	**F**	**T**	**T**	**F**	**F**	**T**	**T**	**F**	**F**	**T**	**T**	**F**	**F**
Password	**T**	**F**	**T**	**F**	**T**	**F**	**T**	**F**	**T**	**F**	**T**	**F**	**T**	**F**	**T**	**F**
Action																
Sign-up Successful	✓															
Sign-up Failed		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

Decision table for forget password.

Condition	**Rules**			
Email	**T**	**T**	**F**	**F**
OTP	**T**	**F**	**T**	**F**
Actions				
Verified	✓			
Not Verified		✗	✗	✗

Decision table for Request Dataset

Condition	**Rules**			
Public	**T**	**T**	**F**	**F**
Published	**T**	**F**	**T**	**F**
Actions				
Assessable	✓			
Not Assessable		✗	✗	✗

Decision table for Manage Profiles

Condition	**Rules**			
Phone No	**T**	**T**	**F**	**F**
Verified Email	**T**	**F**	**T**	**F**
Actions				
Verified	✓			
Not verified		✗	✗	✗

Decision table for Register PCG Centers

Condition	**Rules**			
PMC approved	**T**	**T**	**F**	**F**
Rating > 4.5	**T**	**F**	**T**	**F**
Actions				
Verified	✓			
Not Verified		✗	✗	✗

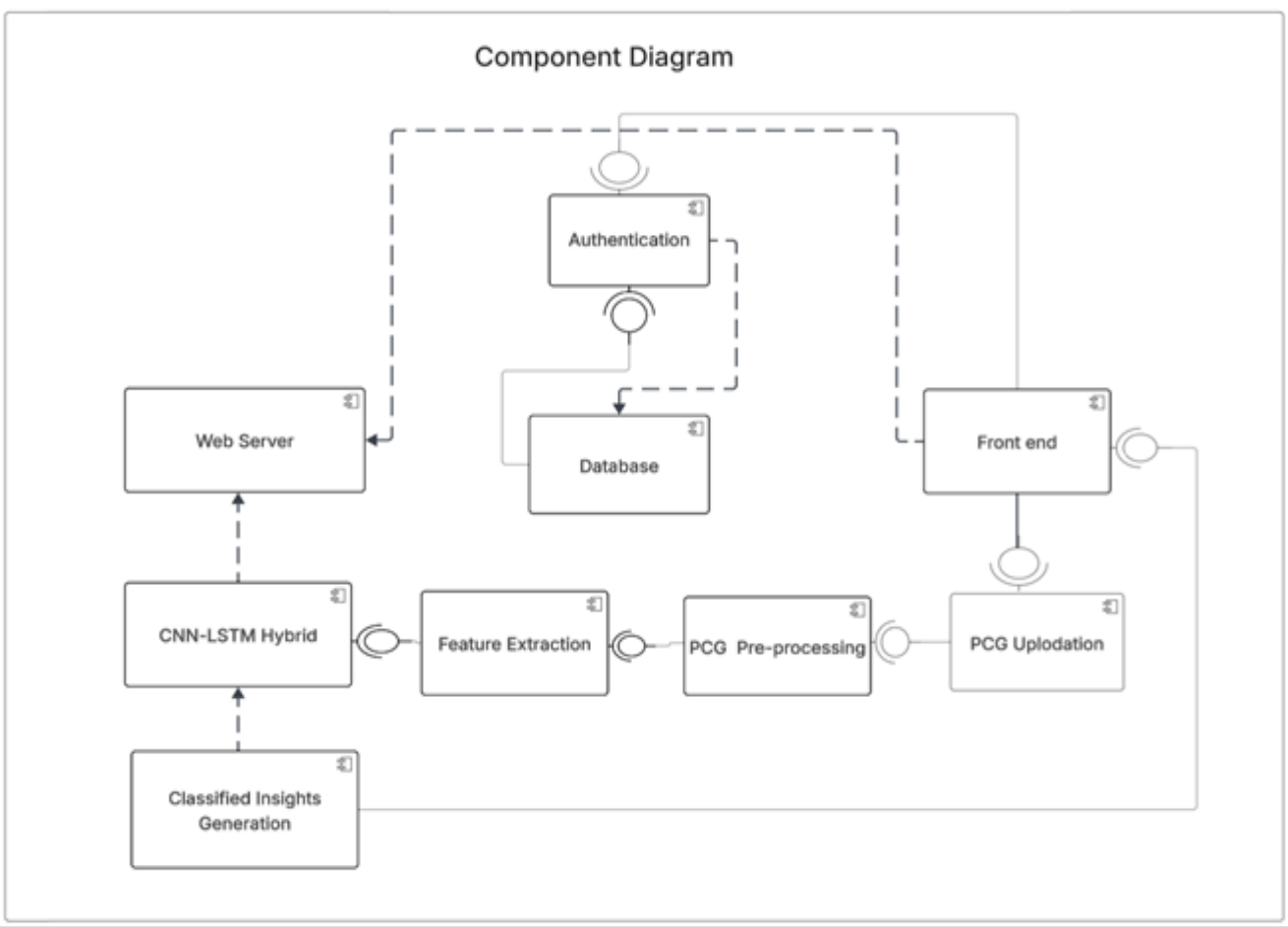
Decision table for PCG upload

Condition	**Rules**	
File upload	**T**	**T**
Actions		
Valid file	✓	
Invalid file		✗

Decision table for Register Doctors & Researchers

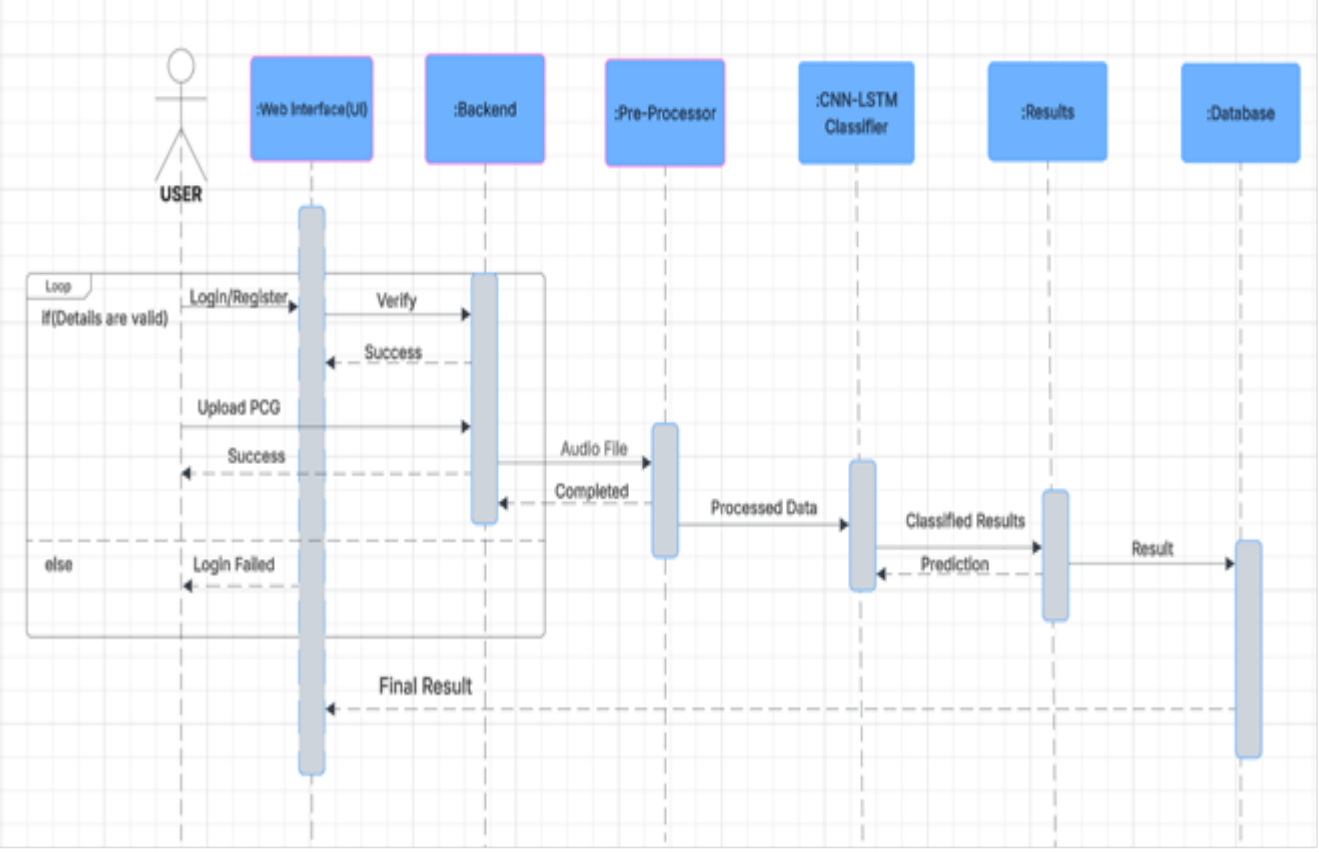
Condition	**Rules**			
PMC Accreditation	**T**	**T**	**F**	**F**
Verified Qualification	**T**	**F**	**T**	**F**
Action				
Verified	✓			
Not Verified		✗	✗	✗

- **Component Diagram**



2.2 Component Interactions and Collaborations

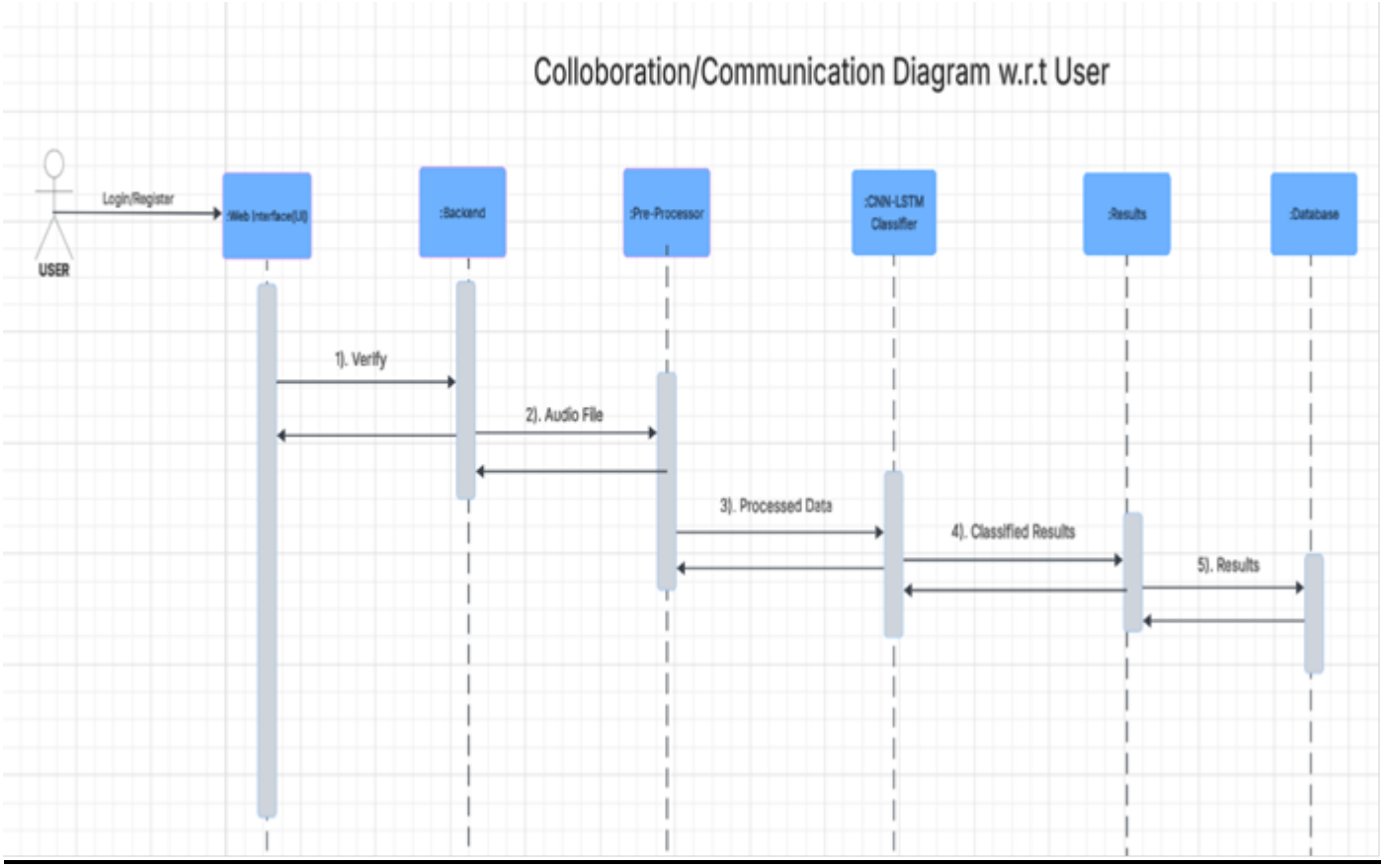
- **Sequence Diagram**



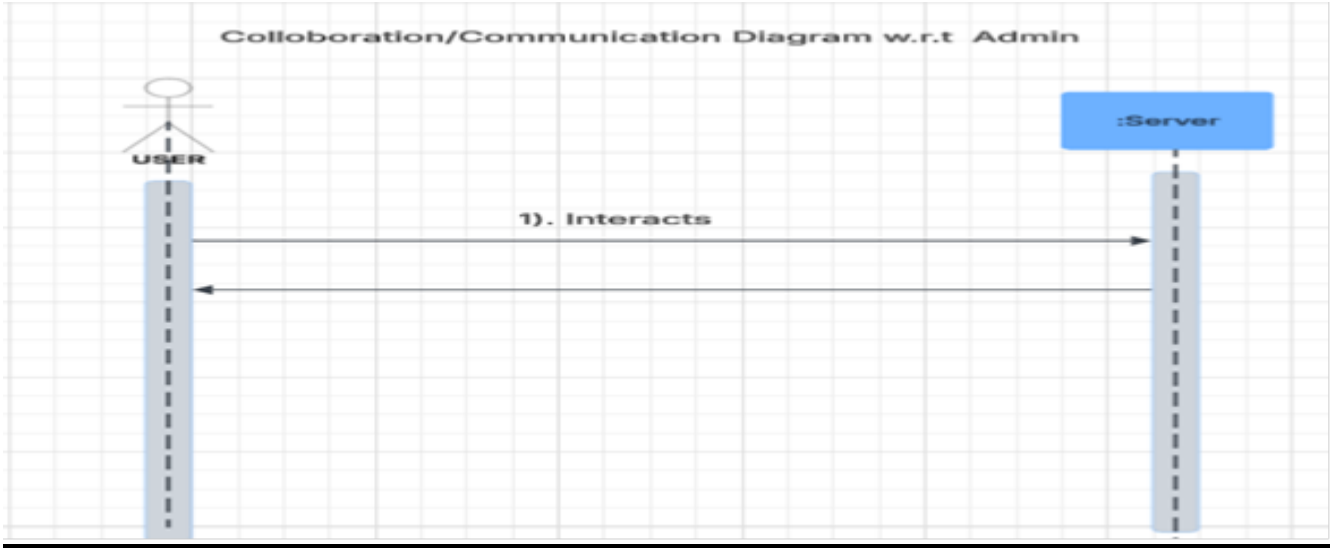
- **Sequence Diagram w.r.t Admin:**

- **Collaboration Diagram:**

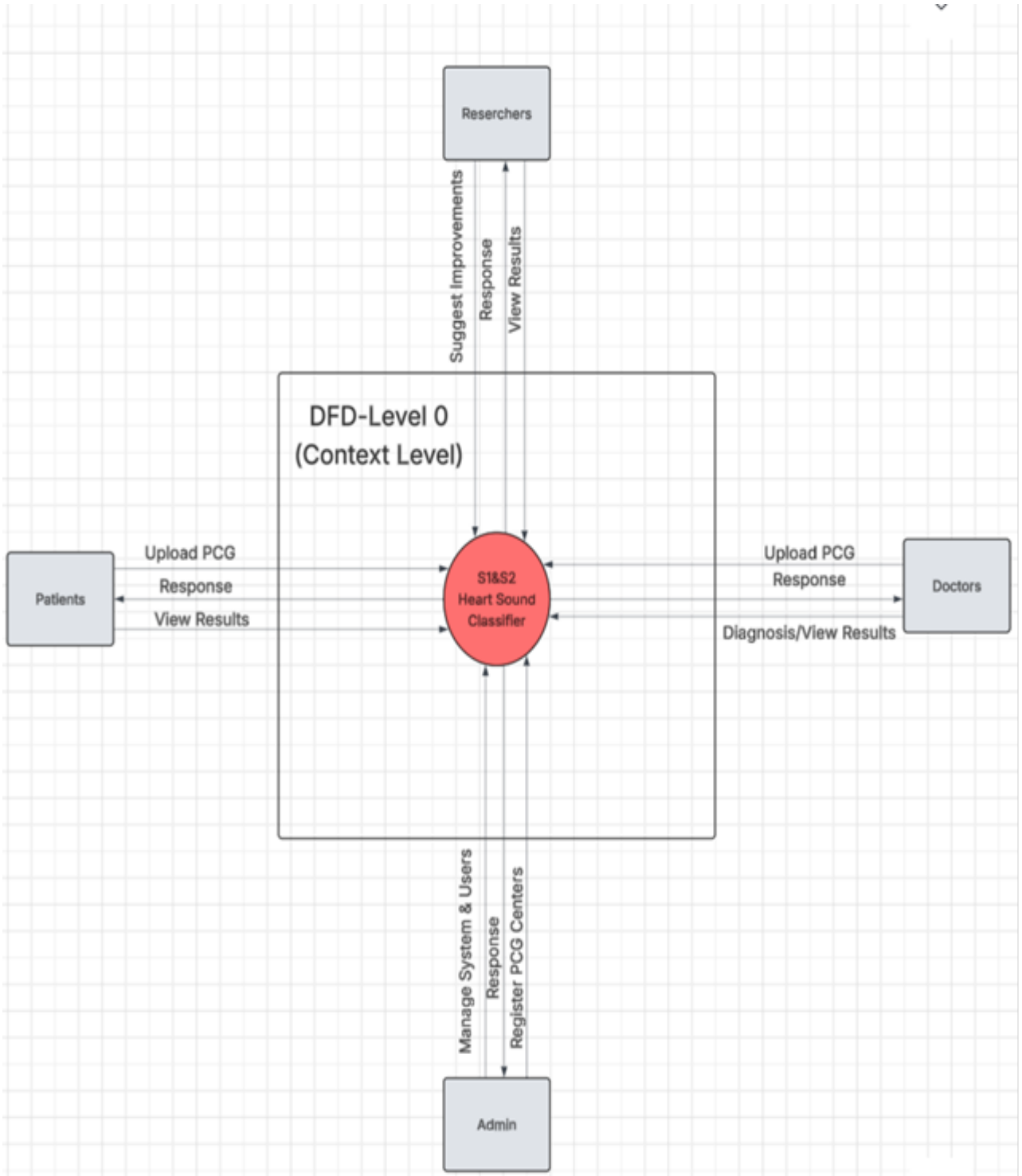
  - **Collaboration Diagram w.r.t User**



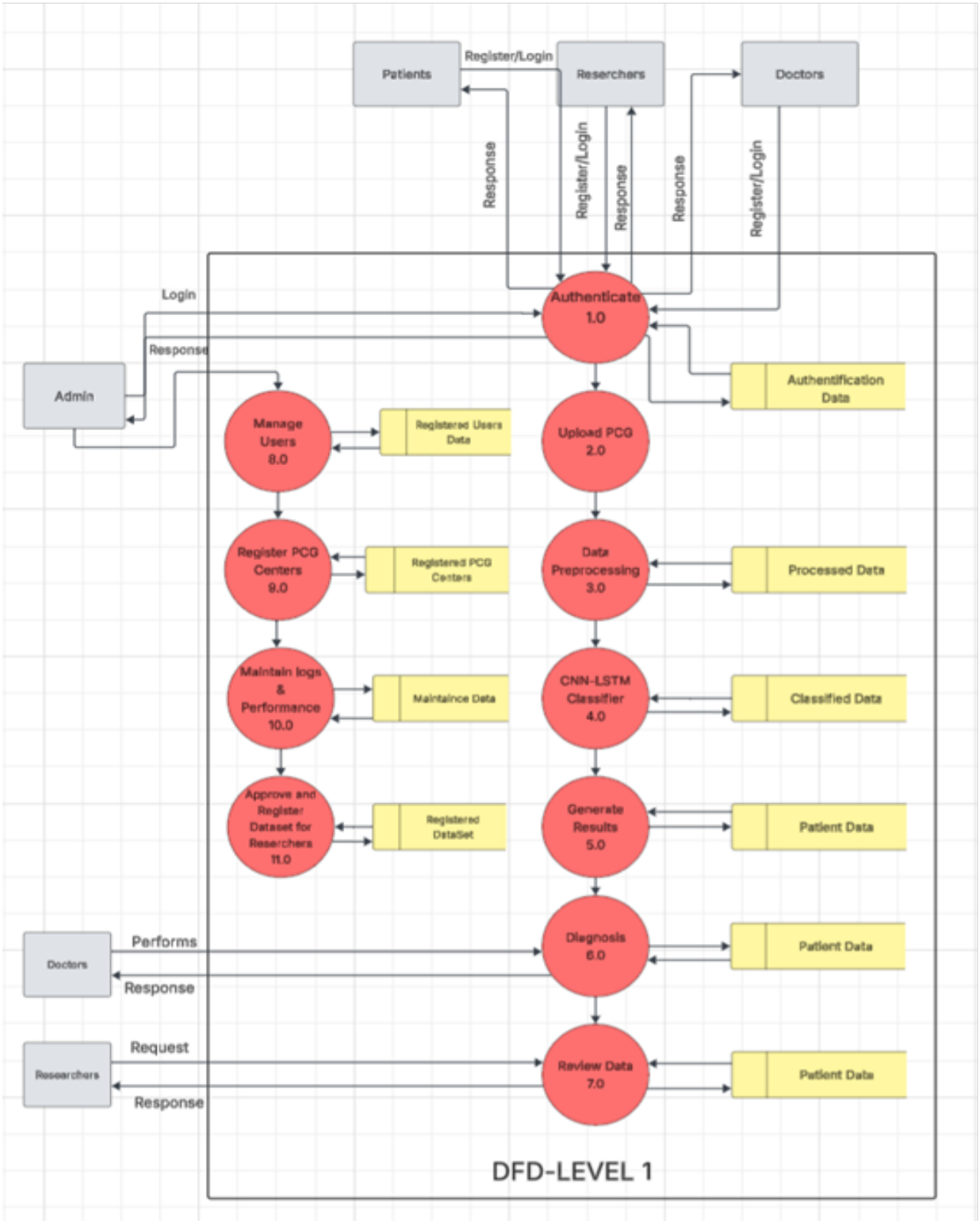
  - **Collaboration Diagram w.r.t Admin**



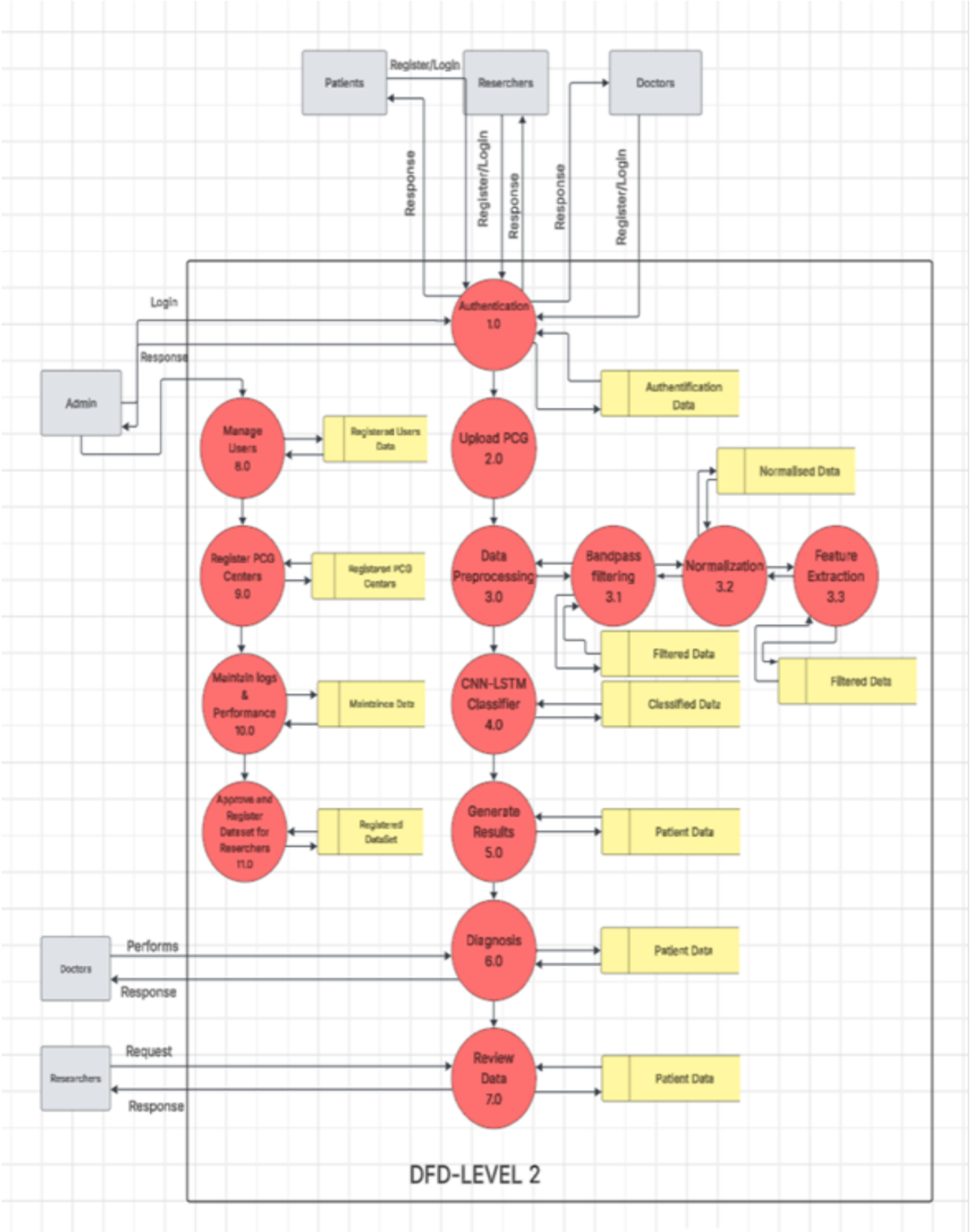
- **Data Flow Diagram**
  - **DFD Level-0 (Context Level):**



- **DFD Level-1**


Patients
Register/Login
Reserchers
Doctors
Response
Register/Login
Response
Response
Register/Login
Login
Authenticate 1.0
Response
Admin
Authentification Data
Manage Users 8.0
Registered Users Data
Upload PCG 2.0
Register PCG Centers 9.0
Registered PCG Centers
Data Preprocessing 3.0
Processed Data
Maintain logs & Performance 10.0
Maintaince Data
CNN-LSTM Classifier 4.0
Classified Data
Approve and Register Dataset for Reserchers 11.0
Registered DataSet
Generate Results 5.0
Patient Data
Doctors
Performs
Response
Diagnosis 6.0
Patient Data
Researchers
Request
Response
Review Data 7.0
Patient Data
DFD-LEVEL 1

- **DFD Level-2**


Patients
Register/Login
Reserchers
Doctors
Response
Register/Login
Response
Response
Register/Login
Login
Authentication 1.0
Response
Admin
Authentification Data
Manage Users 8.0
Registerstt Users Data
Upload PCG 2.0
Normalised Data
Register PCG Centers 9.0
Registaret PCG Centars
Data Preprocessing 3.0
Bandpass filtering 3.1
Normalization 3.2
Feature Extraction 3.3
Filtered Data
Filtered Data
Maintain logs & Performance 10.0
Maintaince Data
CNN-LSTM Classifier 4.0
Classified Data
Approve and Register Dataset for Reserchers 11.0
Registered DataSet
Generate Results 5.0
Patient Data
Doctors
Performs
Response
Diagnosis 6.0
Patient Data
Reserchers
Request
Response
Review Data 7.0
Patient Data
DFD-LEVEL 2

2.3 Design Reuse and Design Patterns

N/A

2.4 Technology Architecture

The anticipated technology architecture for the project is designed to ensure high availability, scalability, and performance to support real-time heart sound classification through a 3-tier web application. The solution leverages a modern cloud infrastructure, containerization, and modular components, ensuring flexibility in deployment and maintainability

- **Technology Stack (3-Tier)**
  - **Presentation Layer**:
    - Built using **ReactJS** or **HTML/CSS/JavaScript**
    - Runs on any **modern web browser**
  - **Application Layer**:
    - Developed using **Django (Python)** for server-side logic
    - Handles REST APIs, authentication, preprocessing
    - Integrates **Librosa** for audio features and **TensorFlow** for CNN-LSTM model
  - **Data Layer**:
    - Uses **PostgreSQL** or **MySQL** for relational data
    - PCG audio files stored in **AWS S3** or local file system

- **Platform & Hosting**
  - The entire system will be deployed on **cloud platforms** like **AWS** or **Azure**
  - All services (frontend, backend, ML model) will be **containerized using Docker**
  - The ML model (CNN-LSTM) is hosted as a **microservice** in a separate container
  - Ensures **high availability, fault tolerance, and scalability**

- **Connectivity Requirements**
  - Accessible via **Internet**, secured with **HTTPS**
  - Requires a **stable broadband or 4G connection** for file uploads/downloads
  - Backend services and ML model communicate through **internal APIs** on the cloud
  - Connectivity is secured via **virtual private networks** or container networks

- **Modes of Operation**
  - **Normal Mode**:
    - Real-time use by doctors/patients for classification and result visualization
  - **Maintenance Mode**:
    - Admin can temporarily disable access or push updates.

- **Batch Mode**:
  - For offline ML model retraining or bulk processing of datasets

- **Security & Access Control**
  - Implements **Role-Based Access Control (RBAC)** for different user types
  - Uses **HTTPS encryption** for all web traffic
  - Login sessions handled via **Django auth** with optional **Two-Factor Authentication (2FA)**
  - All research data is **anonymized** before access by researchers

- **Anticipated Infrastructure Components**
  - **Web Server**: Django backend container
  - **ML Inference Engine**: TensorFlow
  - **Database Server**: PostgreSQL or MySQL
  - **File Storage**: AWS S3 bucket or cloud file system
  - **Frontend Client**: Hosted as static files
  - **Deployment**: GitHub Actions, Docker Compose for build and deployment automation

2.5 Architecture Evaluation

Rationale Behind Technology Selection

The architectural choices for this project were made after carefully analyzing performance requirements, scalability needs, dataset characteristics, and deployment feasibility.

1. CNN–LSTM Hybrid Model

Reason for Selection:

- PCG signals contain both **spatial patterns** (frequency distribution) and **temporal patterns** (heartbeat sequence).
- CNN effectively extracts **local time–frequency features** from spectrograms.
- LSTM captures **long-term temporal dependencies**, enabling accurate identification of S1 and S2 cycles.
- Hybrid architectures have proven to outperform standalone CNN or RNN models in biomedical audio classification tasks.

Alternative Considerations & Limitations:

- **Pure CNN models:** Fast but fail to capture cyclic heartbeat dependencies.
- **Pure LSTM/GRU models:** Good for sequences but struggle with high-dimensional spectrogram inputs.
- **Transformer-based models:** More powerful but computationally expensive and not ideal for real-time web deployment.

The CNN-LSTM solution balances **accuracy, speed, and computational cost**, making it ideal for real-time classification.

2. Django Framework for Backend

Reason for Selection:

- Django provides a **robust, secure, and structured MVC architecture**.
- Built-in **authentication**, **ORM**, **security middleware**, and **session management** reduce development time.
- Its scalability supports future roadmap extensions (e.g., murmur detection, multi-class systems).
- Excellent fit for ML integration using Python-based libraries (TensorFlow, Librosa, NumPy).

Alternative Considerations & Limitations:

- **Flask:** Lightweight but requires manual implementation of security, form handling, and database layers.
- **FastAPI:** Very fast and modern but less aligned for full-fledged role-based web applications.

Django was chosen for its **enterprise-grade stability**, structured environment, and faster development cycle.

3. Docker-Based Deployment

Reason for Selection:

- Ensures **environment consistency** across development, testing, and production.
- Enables **microservices architecture**, isolating ML inference, backend, and frontend in separate containers.
- Ideal for cloud providers like AWS/Azure that support container orchestration.
- Simplifies CI/CD pipelines, versioning, and rollback procedures.

Alternative Considerations & Limitations:

- **Virtual Machines:** Heavyweight and slower to scale.
- **Bare-metal server deployment:** No isolation; difficult to maintain or update.

Docker offers the optimal combination of **portability, modularity, and low resource overhead**.

4. Cloud Hosting (AWS / Azure)

Reason for Selection:

- Offers **high availability**, auto-scaling, secure storage, and easy integration with S3 and managed databases.
- Supports GPU-based training/hosting if needed in the future.
- Reduces local infrastructure dependency.

Alternative Considerations & Limitations:

- **Local hosting:** Limited scalability and downtime risk.
- **Shared hosting:** Not suitable for ML workloads.

Cloud deployment ensures professional-level reliability and meets real-time medical system expectations.

5. PostgreSQL Database

Reason for Selection:

- Reliable relational integrity, essential for role-based access and medical records.
- Superior performance in handling structured, transactional data.
- Strong security features and support for JSON fields.

Alternative Options:

- **MySQL:** Similar performance but less efficient with complex queries.
- **NoSQL (MongoDB):** Not suitable for strict relational constraints.

PostgreSQL was chosen for its **balance of robustness, performance, and data consistency**.

Conclusion:

The selected architecture achieves the following:

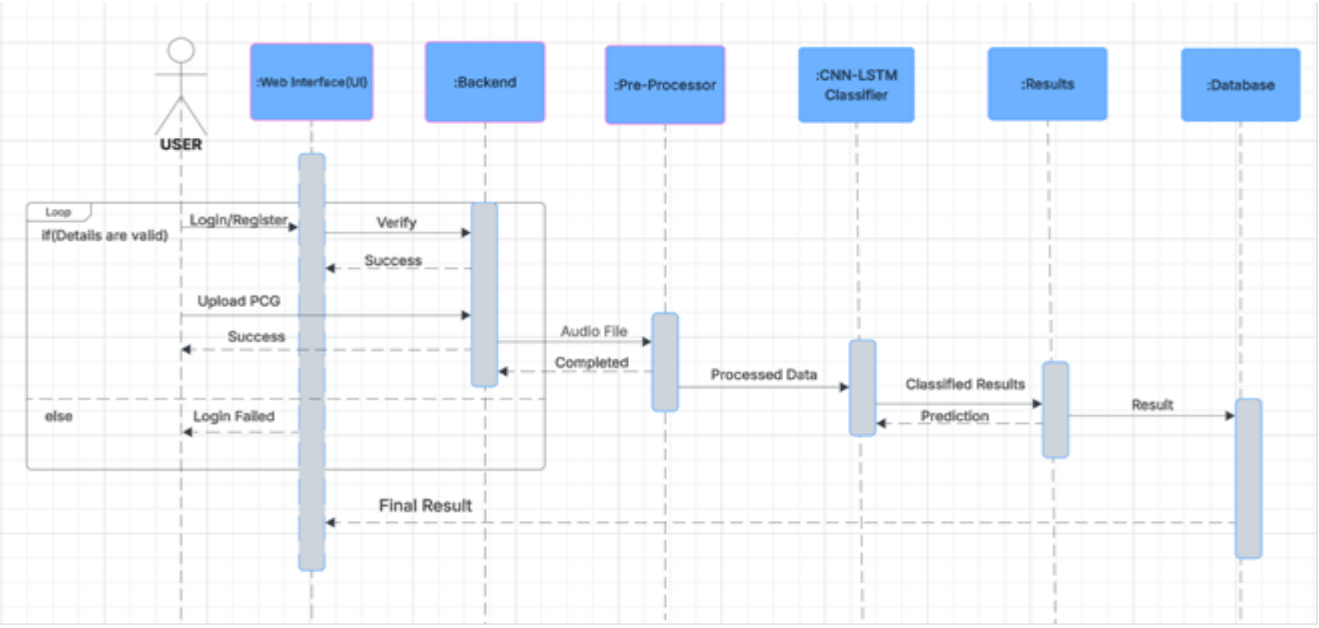
- **High accuracy** through efficient ML model selection
- **Scalability** through Docker and cloud deployment
- **Security** suitable for medical data handling
- **Maintainability** through modular design and MVC principles
- **Future extensibility** for additional heart sound classes or multi-disease detection

This combination ensures that the system is not only functional today but also **future-ready** for research advancements and real-world hospital deployment.

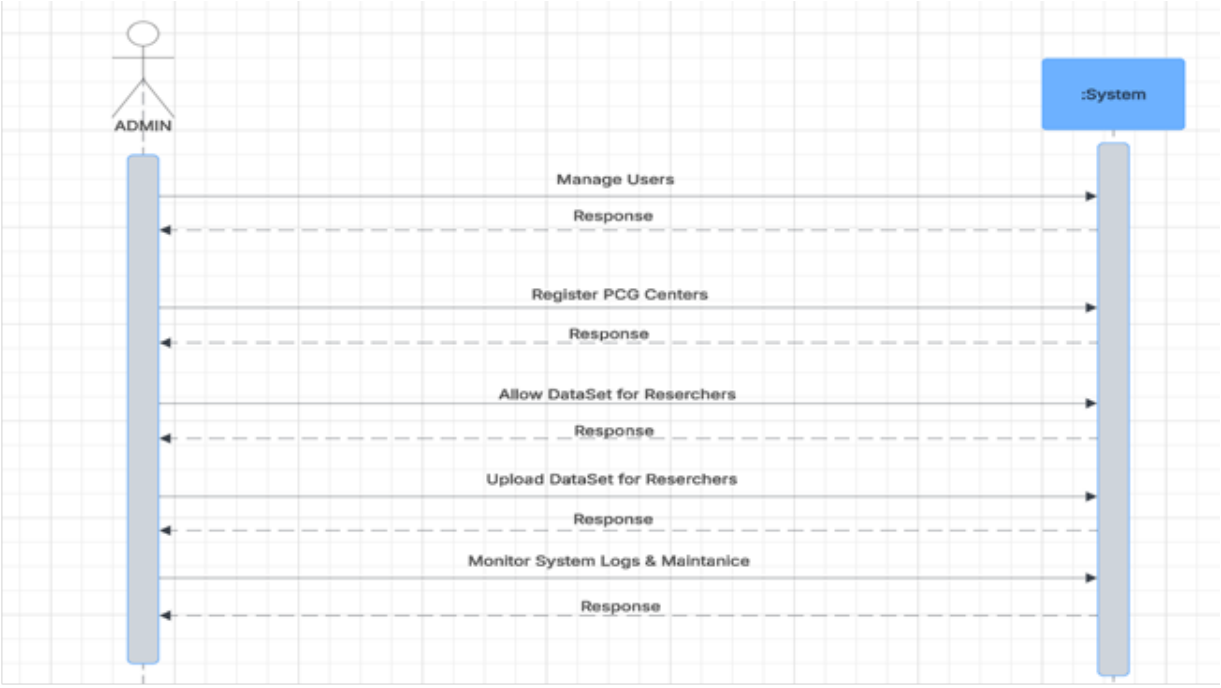
3. Detailed/Component Design

3.1 Component-Component Interface

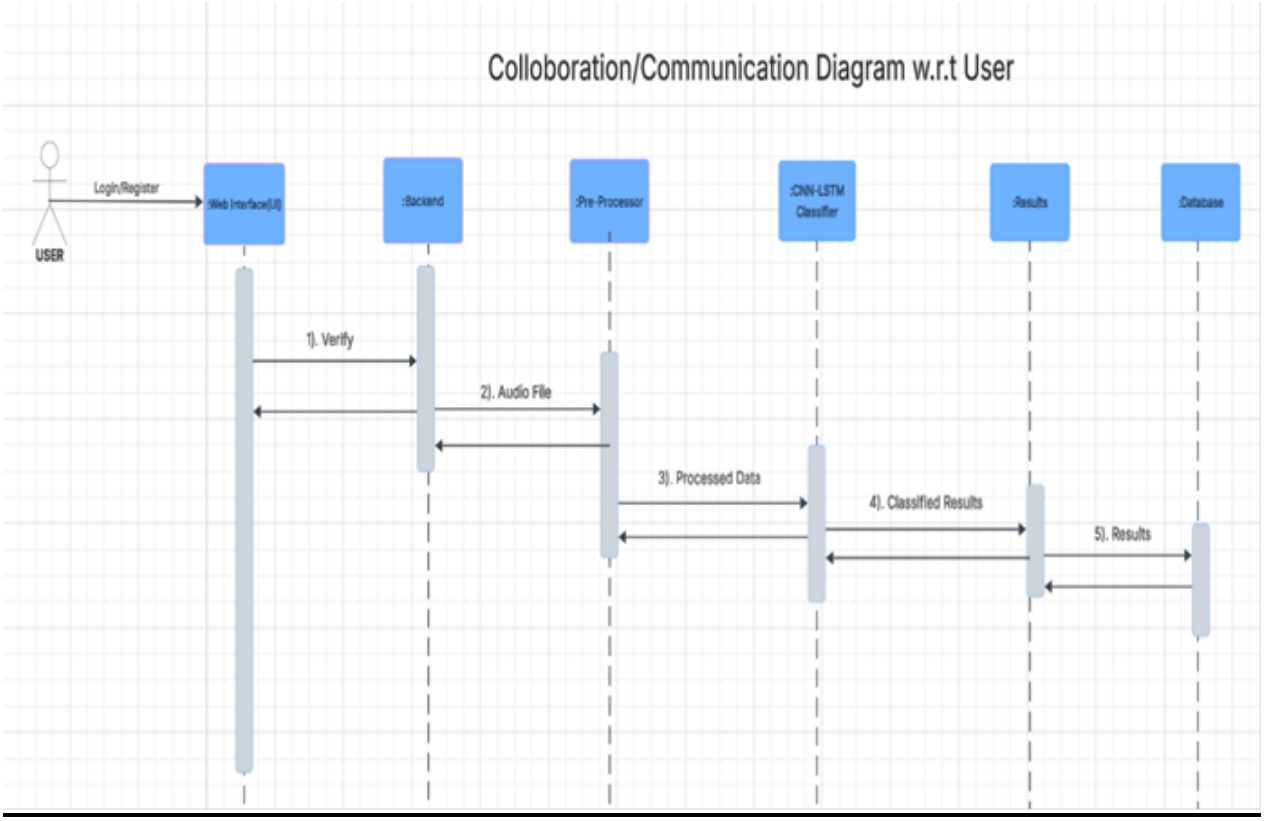
- **Sequence Diagram:**
  - **Sequence Diagram w.r.t User:**



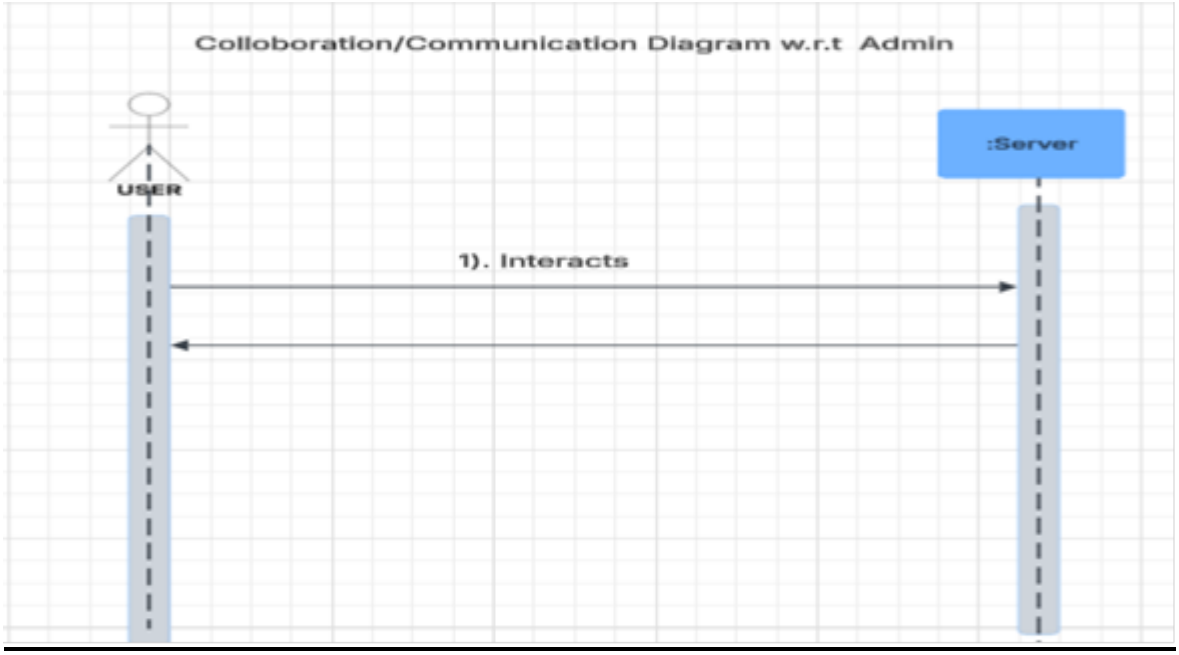
- **Sequence Diagram w.r.t Admin**



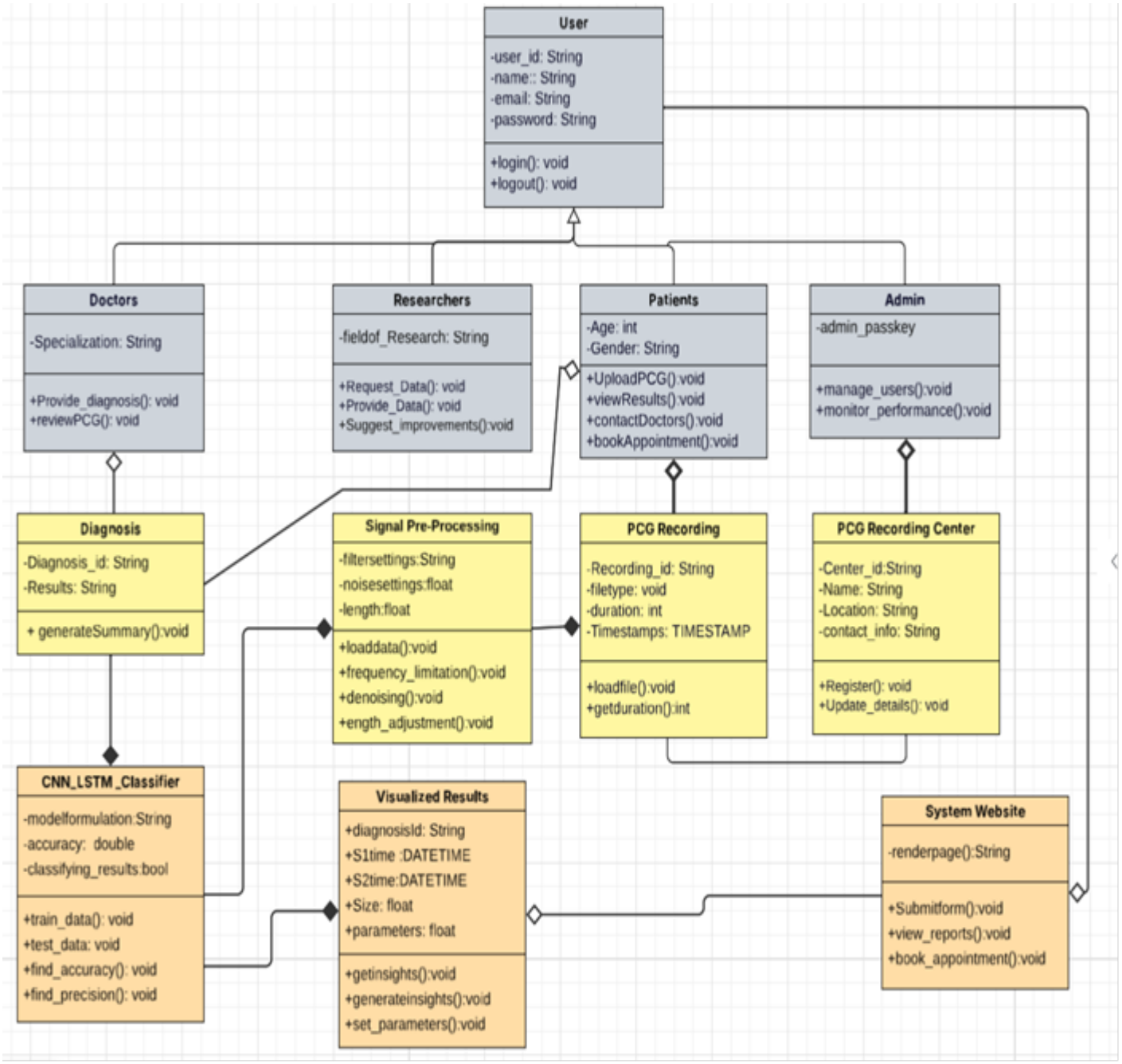
- **Collaboration Diagram:**
  - **Collaboration Diagram w.r.t User:**



  - **Collaboration Diagram w.r.t Admin:**



- **Class Diagram**



3.2 Component-External Entities Interface

The system uses a structured **3-Tier Architecture**, where each tier communicates with specific external entities. The interaction between tiers ensures a smooth flow of data from the user to the backend and finally to the data storage components. The following describes how each tier interacts with external entities as depicted in the architecture diagram.

1. User Tier

The **User Tier** represents all external users, such as doctors, patients, and researchers. Users interact with the system through a **User Interface** displayed in the **Web Browser**, making this tier an external entity to the server-side components.

External Interactions:

- Users upload PCG audio files through the web browser.
- The browser sends user requests (upload, view results, check PCG details) to the **Web Application Server** via HTTP/HTTPS.
- The User Interface acts purely as a presentation layer and does not perform processing.
- All rendered screens such as:
    - Main Interface
    - PCG Input Interface
    - View Results Interface are externally accessed from this tier.

Purpose of Interaction:

- Allow external users to initiate actions.
- Display results generated by backend processing.

2. Server Tier

The **Server Tier** is the core of the application and interacts with multiple external components. It includes the **Web Application Server**, backend logic modules (Verified Records, Classified Results), and the **CNN-LSTM Model**.

External Interactions:

a. User Tier → Server Tier

- Receives user uploads and requests from the browser.
- Sends visual responses back to the User Interface.

b. Backend → CNN-LSTM Model (External Processing Entity)

- The CNN-LSTM model acts as an external classification engine.
- Backend sends verified PCG recordings to the model for classification.
- Model returns S1/S2 results and confidence scores.

c. Backend → Visual Interface (External Presentation Modules)

- Backend passes processed results to various UI modules for display.
- These UI modules act as external entities connected to backend outputs.

Purpose of Interaction:

- Process user data.
- Communicate with DL model.
- Route results and outputs to the visual interfaces.

3. Data Tier

The **Data Tier** consists of independent data management components such as the **Database** and **PostgreSQL-Server**.
These systems act as external entities that the server tier relies on for persistent storage.

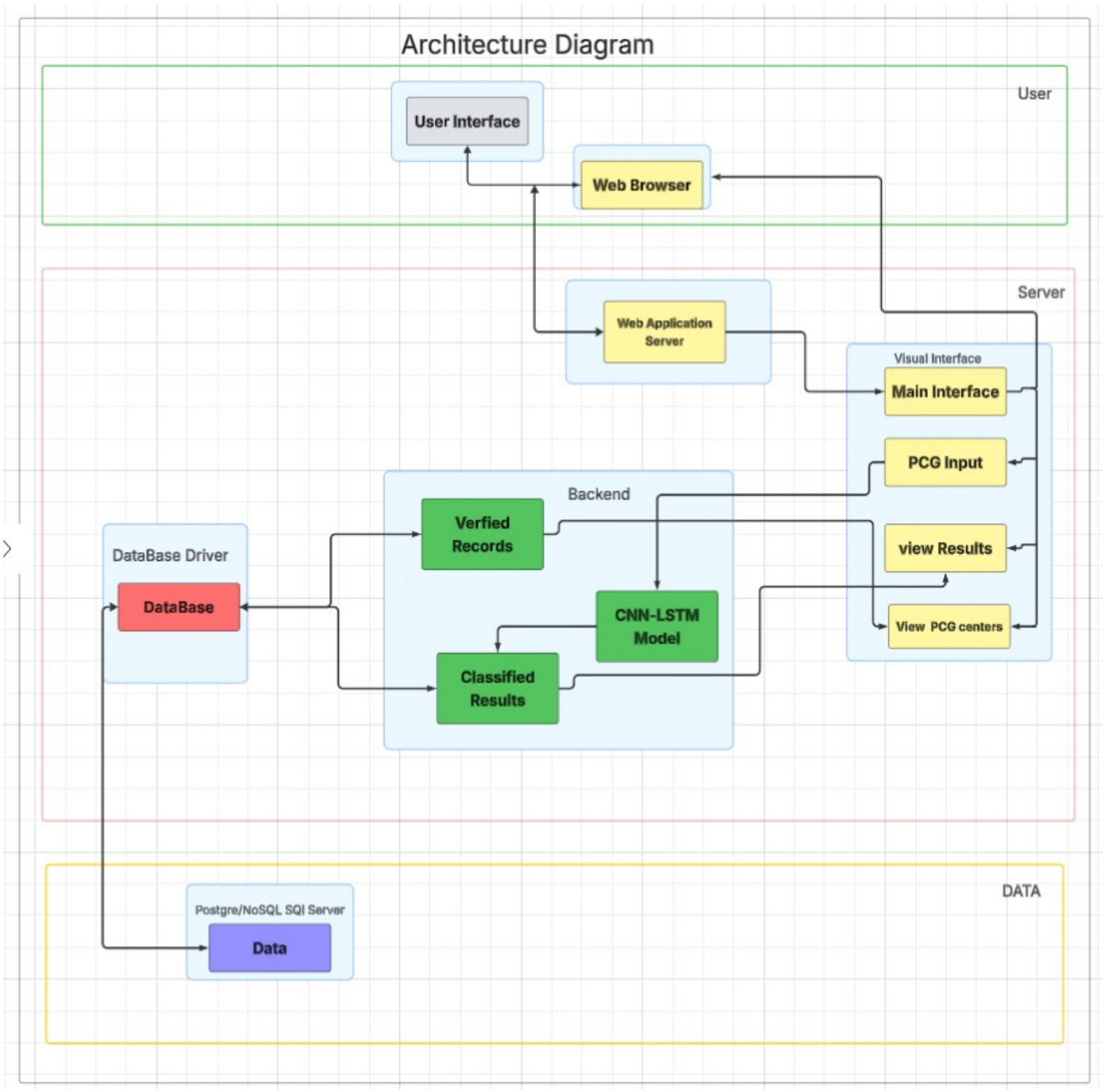
External Interactions:

- Backend communicates with the **Database Management System** to save:
    - Verified PCG records
    - Classified results
    - User activity logs
- Backend retrieves stored data when users request:
    - Previous PCG submissions
    - Past classification results
    - Record metadata
- PostgreSQL Server acts as an external storage engine for all structured data.

Purpose of Interaction:

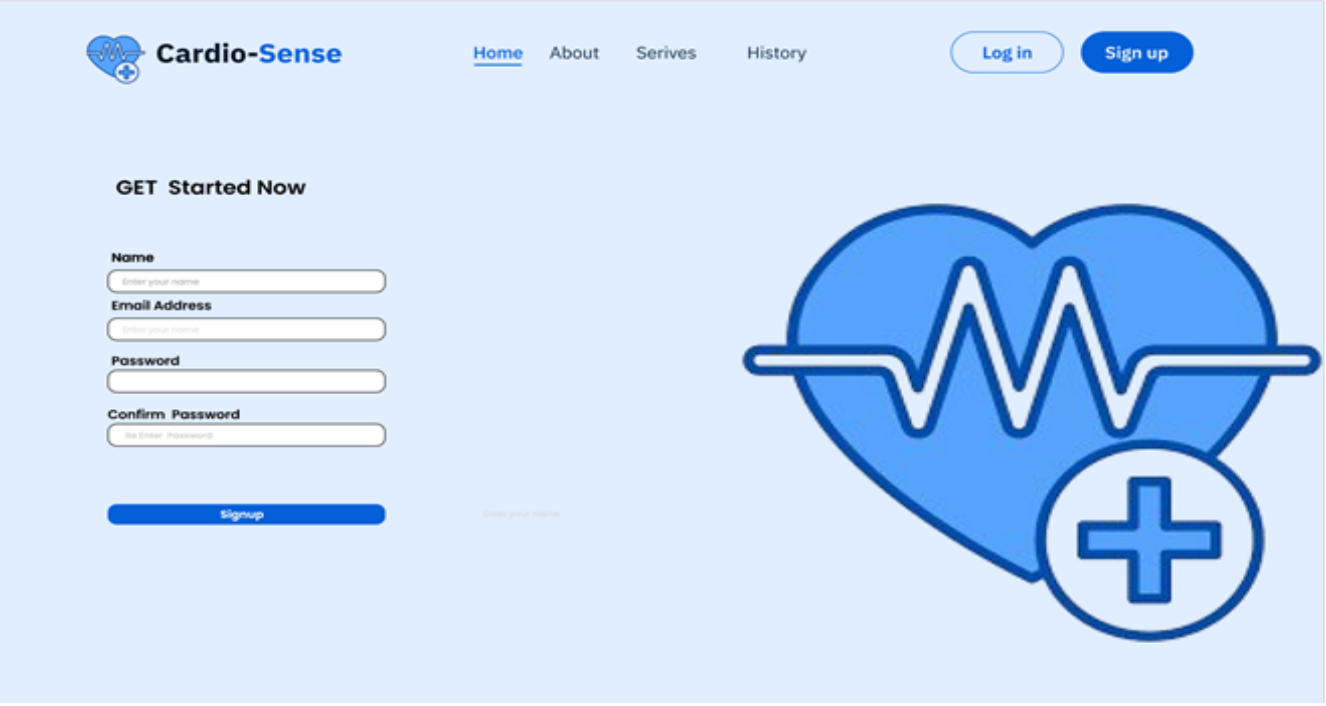
- Provide long-term, secure storage for PCG files and classification outputs.
- Ensure consistent retrieval of data for user requests.

Architecture Diagram:

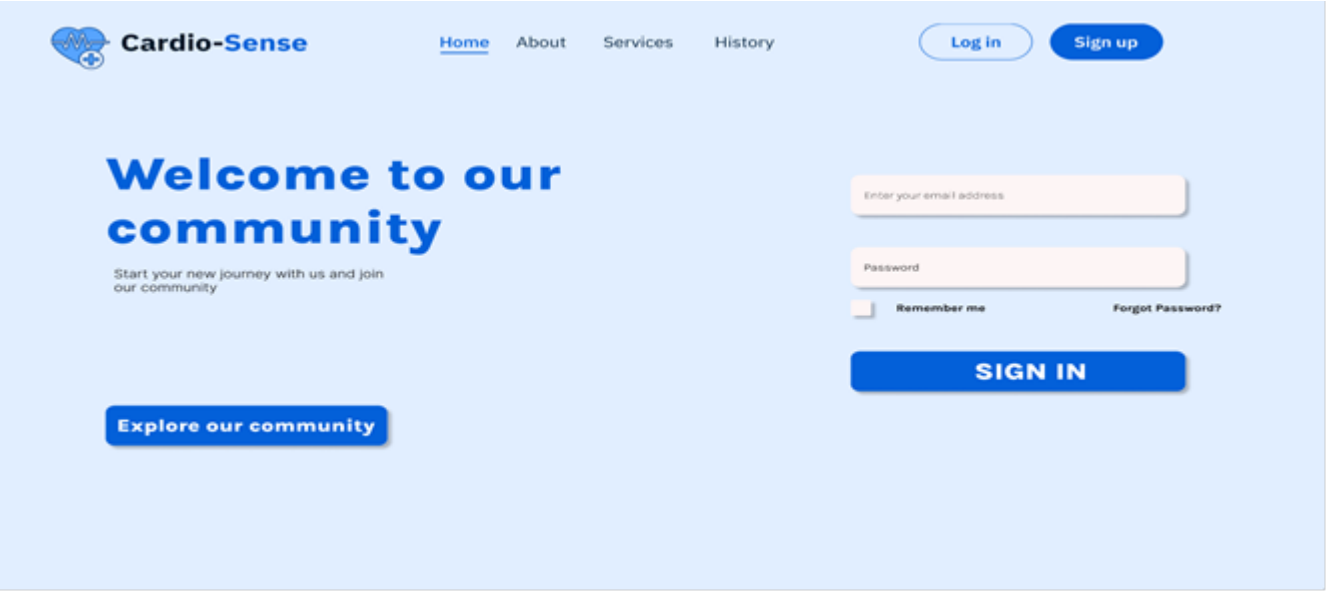


3.3 Component-Human Interface

Screen 1: Sign-up Register Page



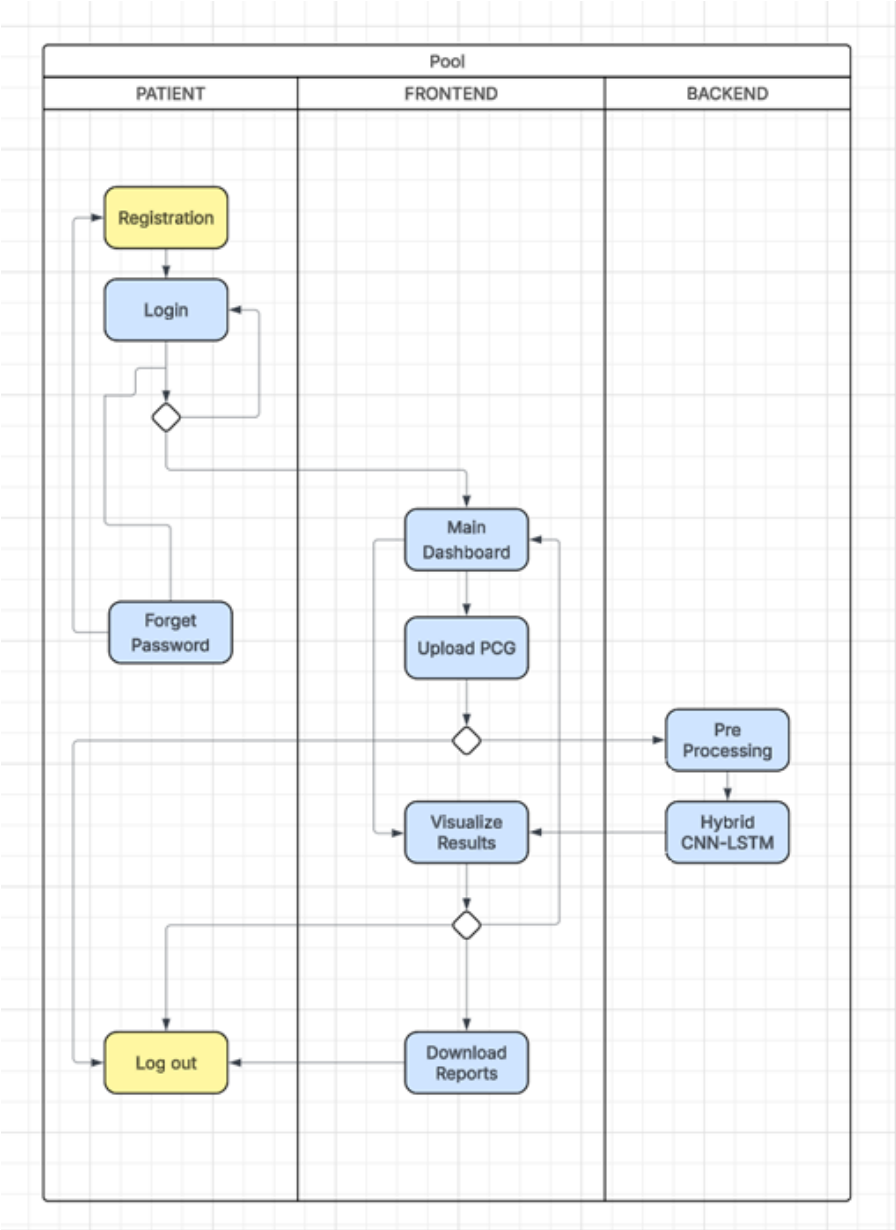
Screen 2: Register Page:



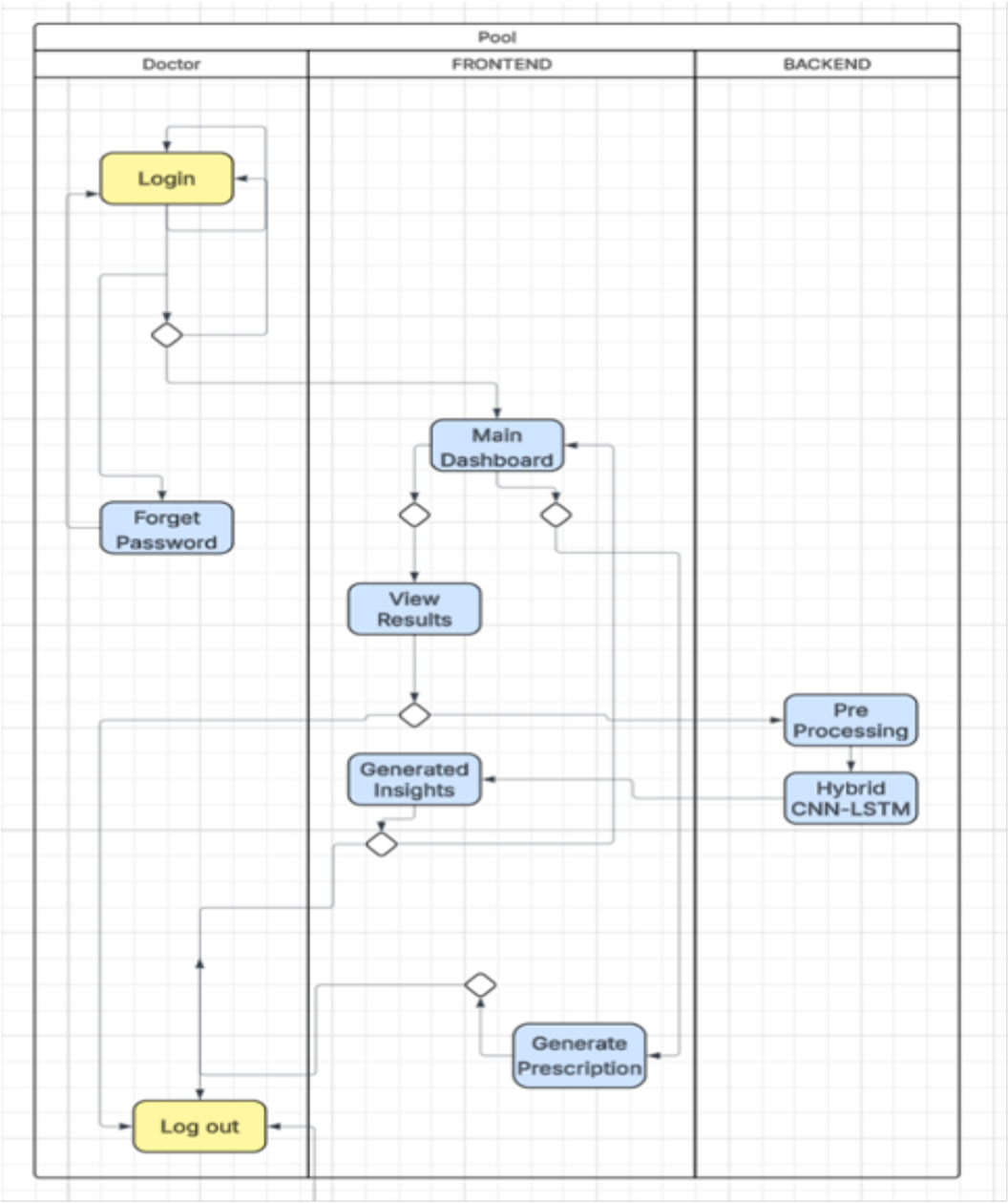
4. Screenshots/Prototype

4.1 Workflow

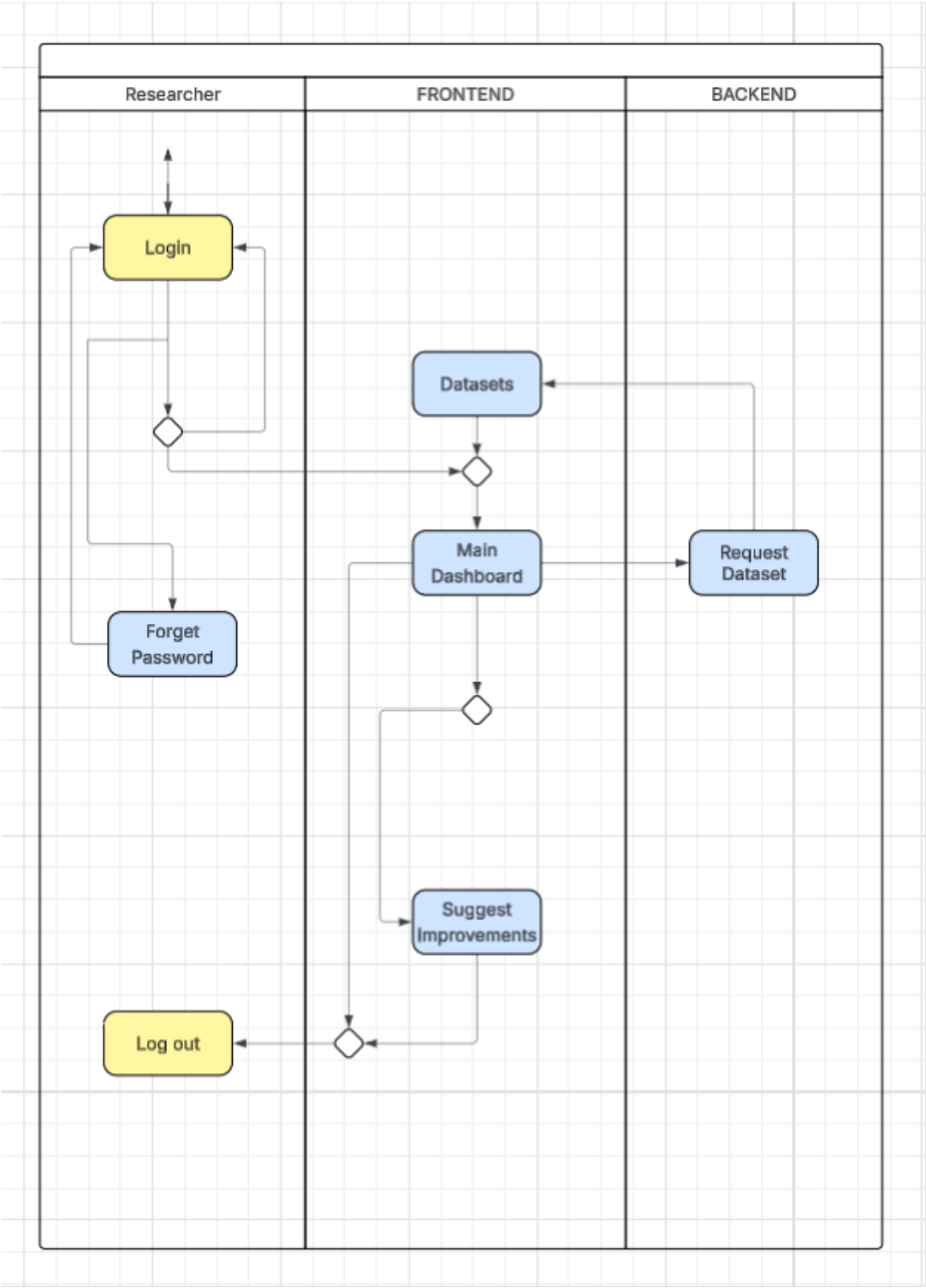
- **Swim Lane Diagram w.r.t Patients**



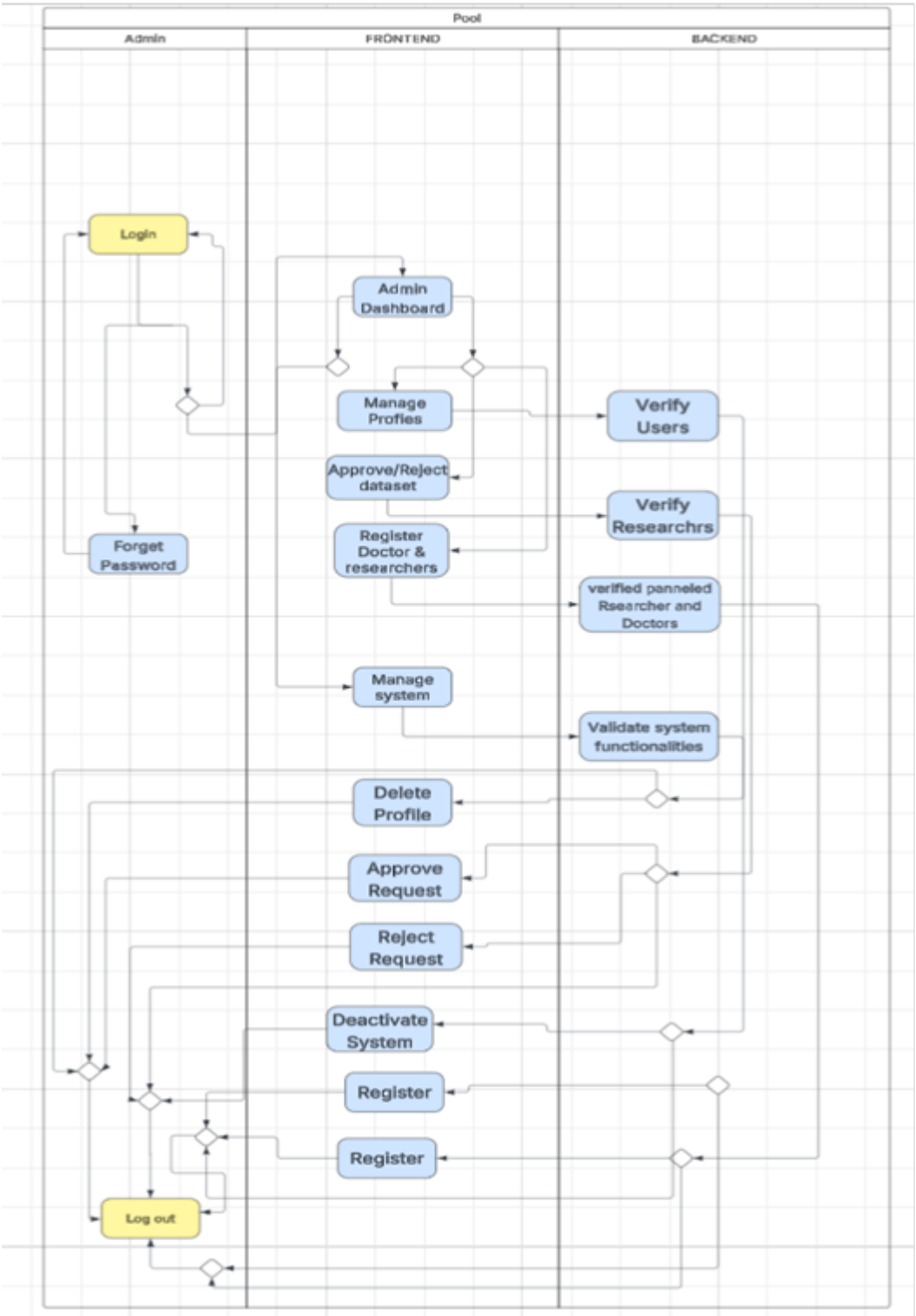
- **Swim Lane Diagram w.r.t Doctors**



- **Swim Lane Diagram w.r.t Researchers**

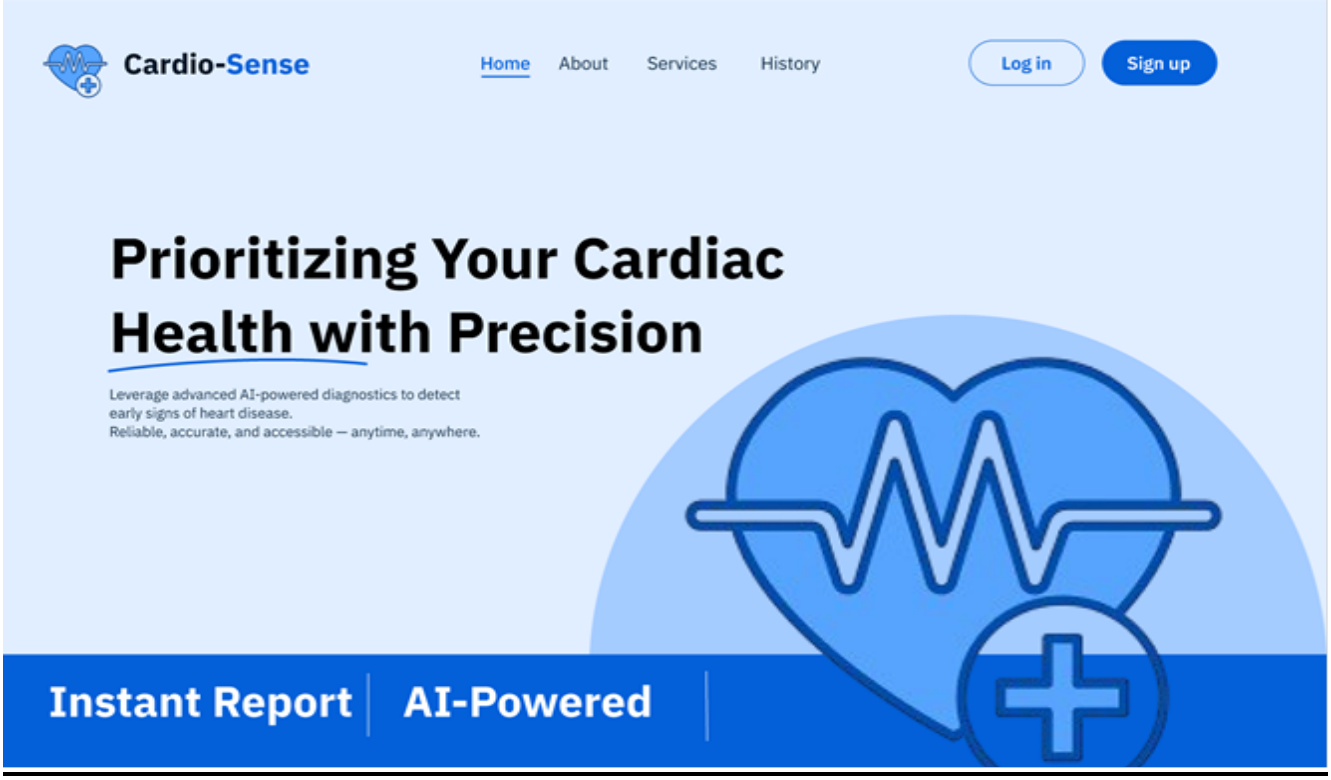


- **Swim Lane Diagram w.r.t Admin**

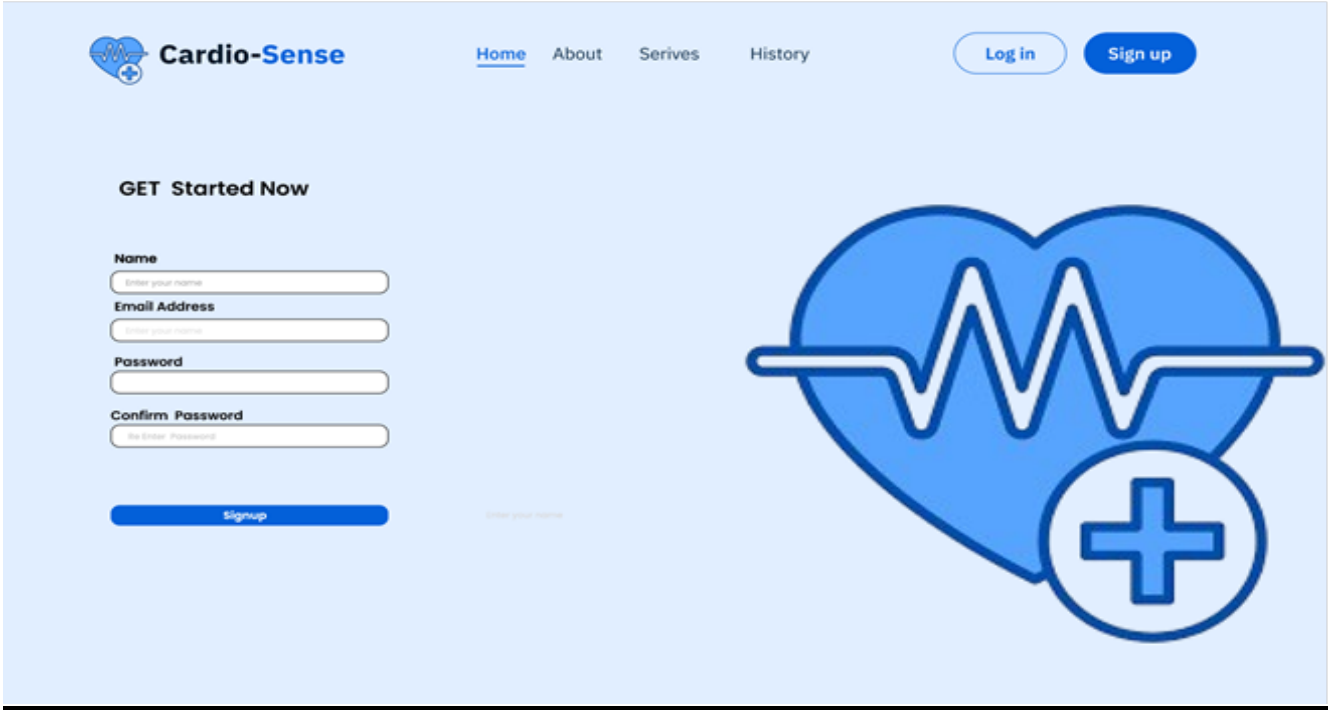


4.2 Screens

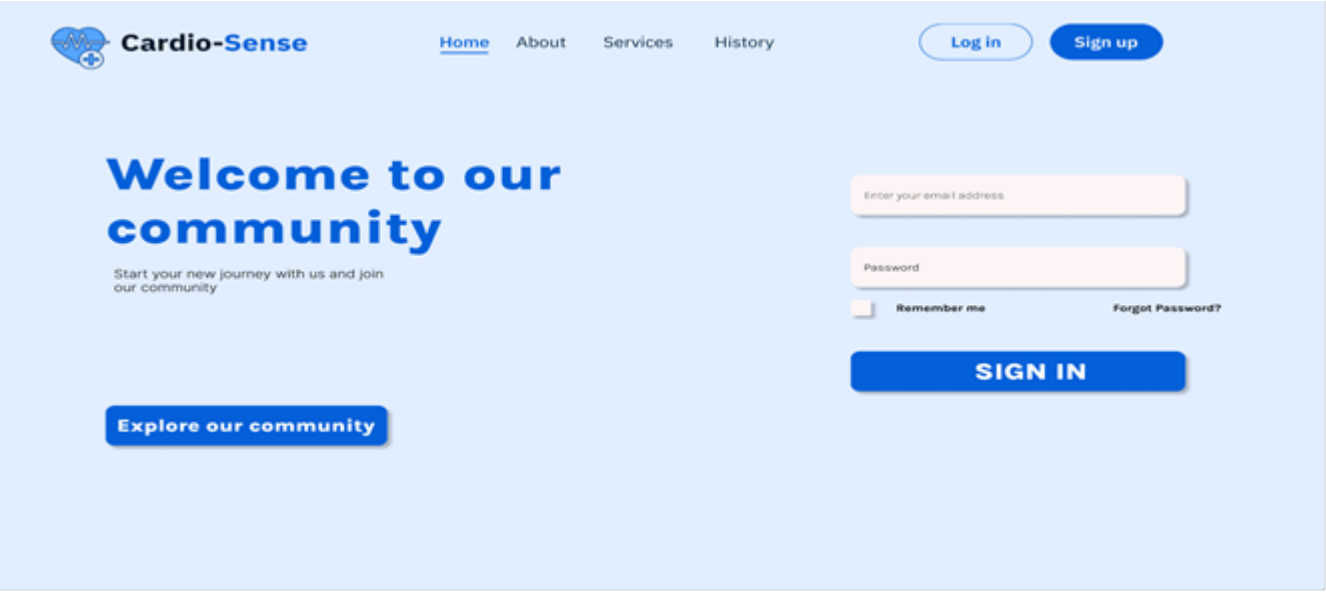
Screen 1:



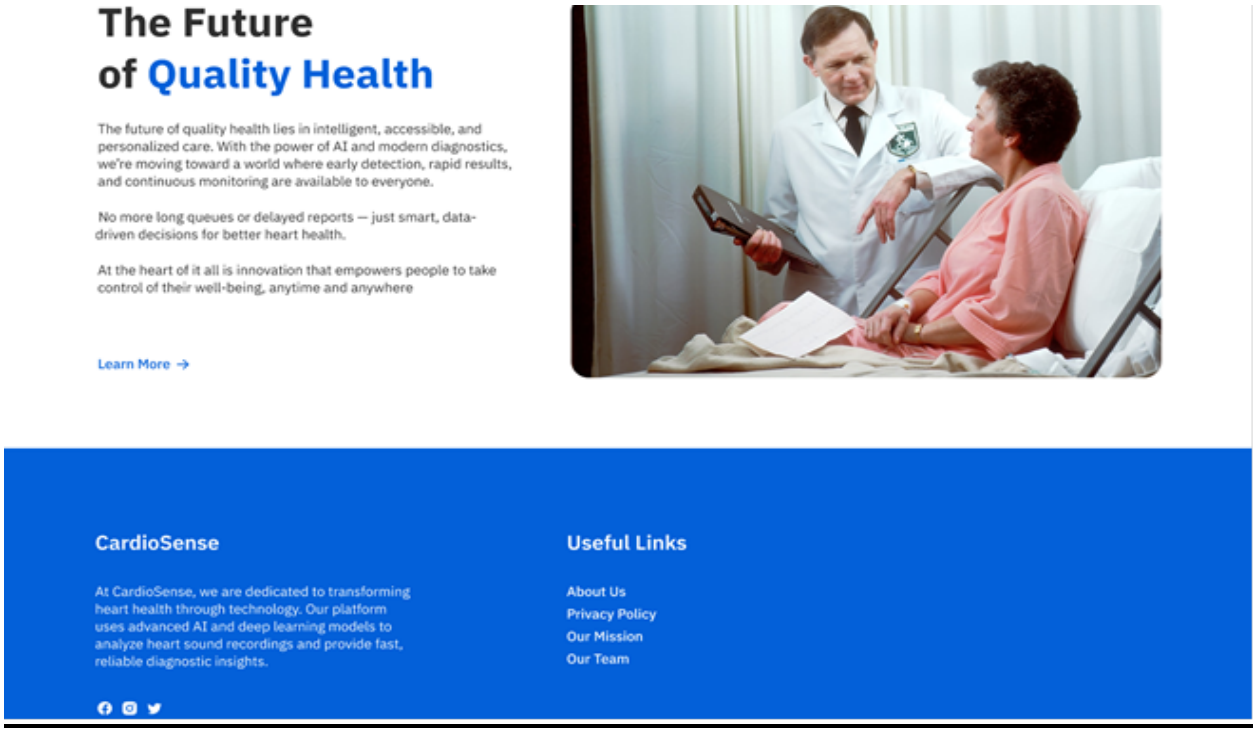
- **Screen 2**



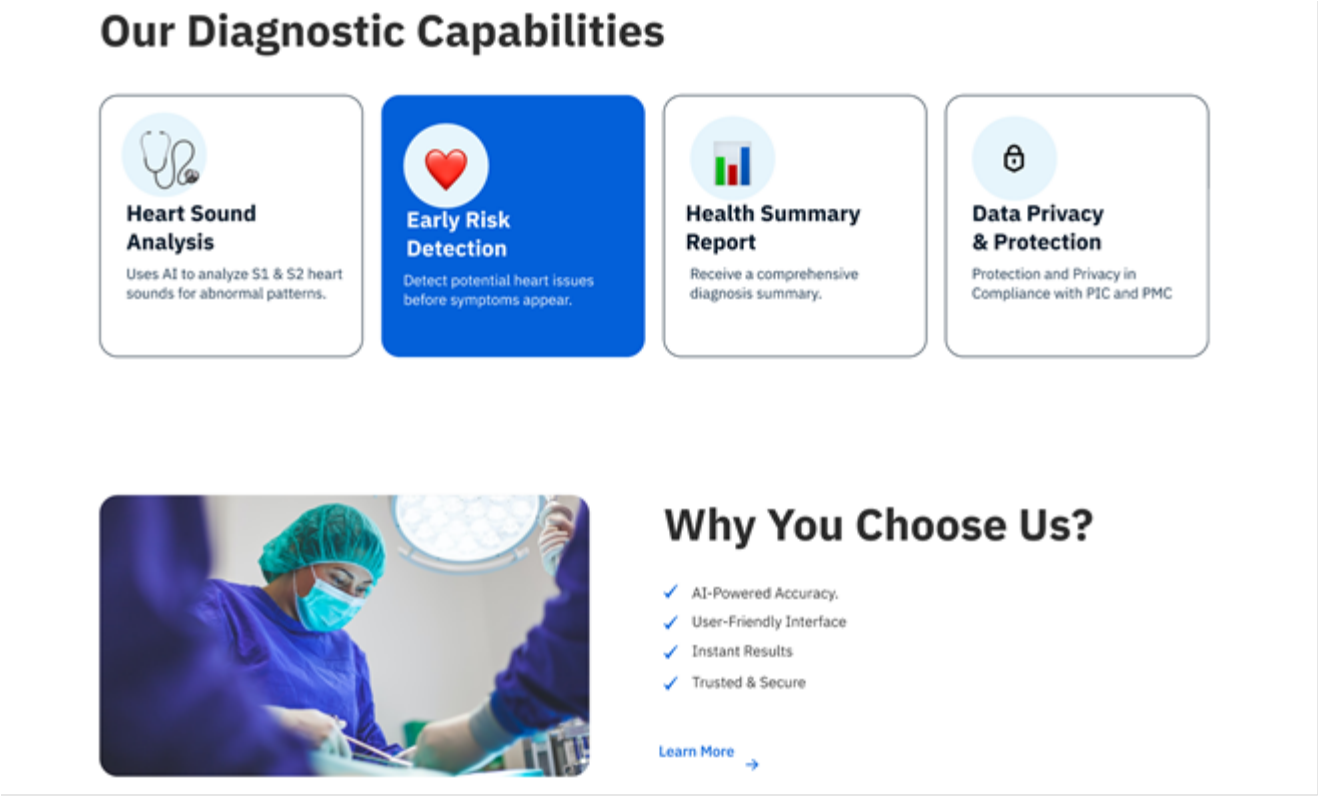
- **Screen 3:**



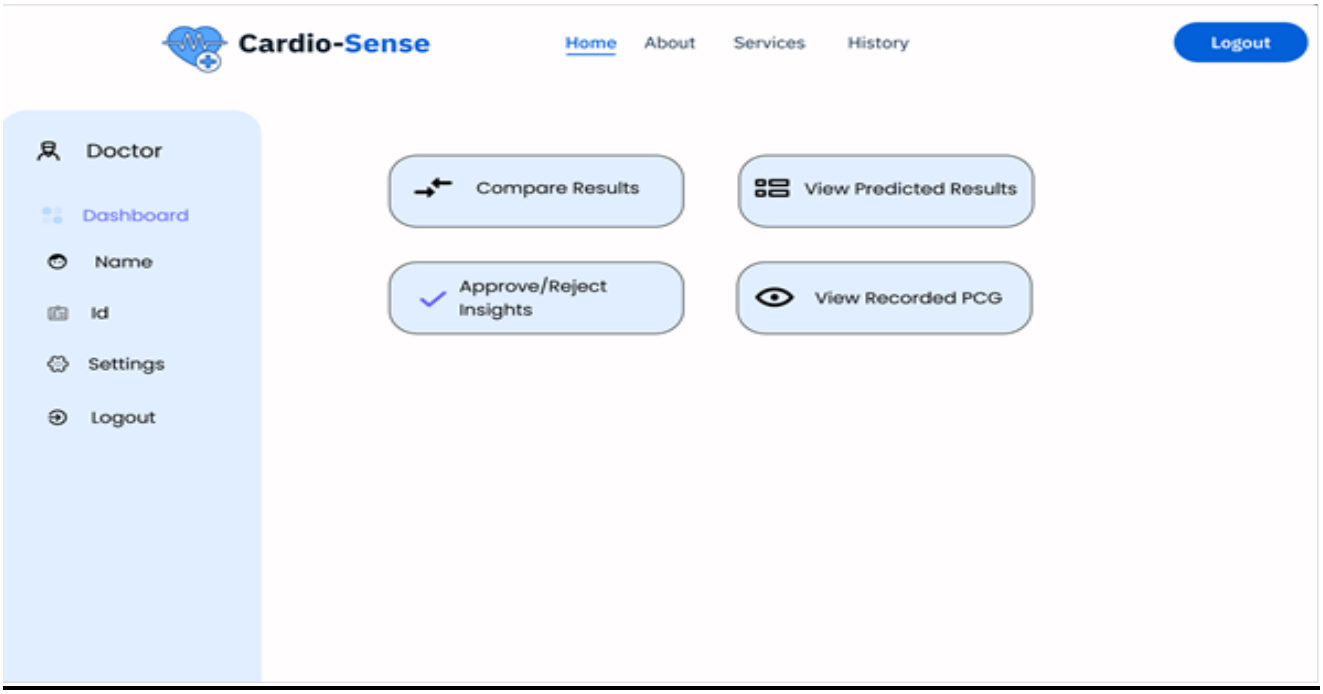
- **Screen 4:**



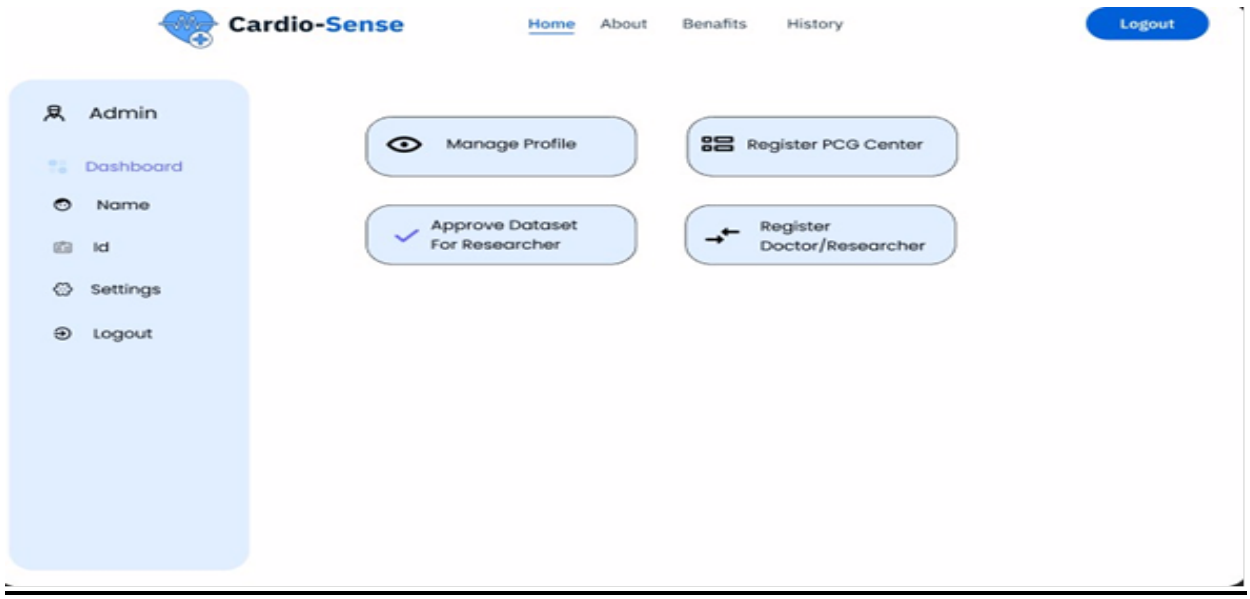
- **Screen 5:**



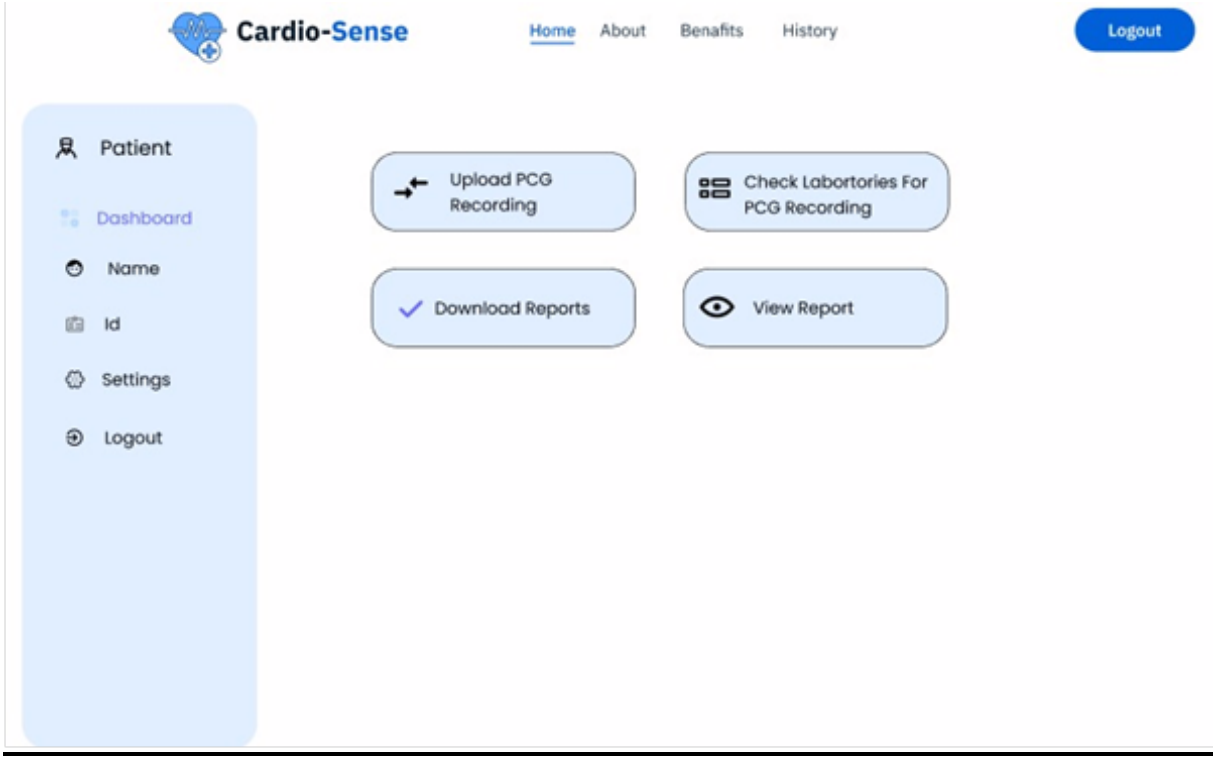
- **Screen 6:**



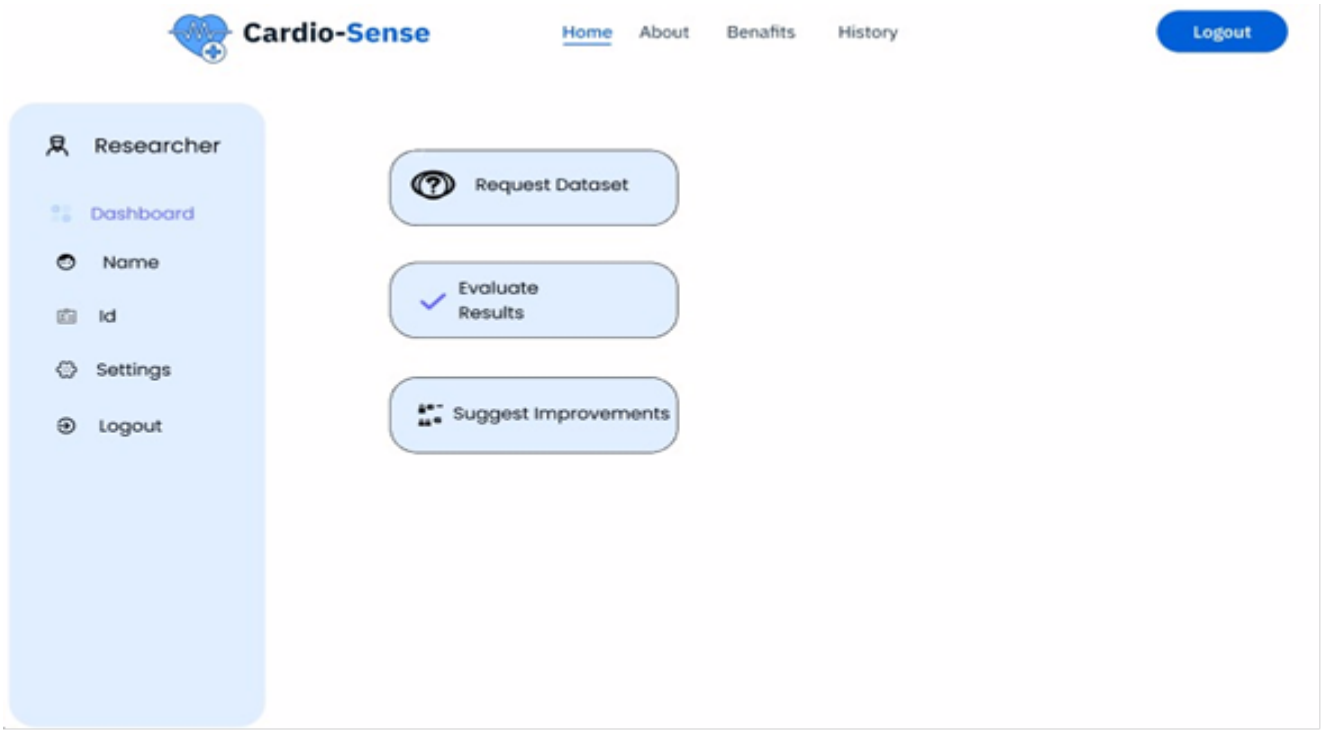
- **Screen 7:**



Screen 8:



- **Screen 9:**



4.3 Additional Information

- **Team Collaboration Tools**:
  Project coordination is managed using platforms like **GitHub** (for version control), **Google Docs** (for documentation), and **Trello** (for task tracking).

- **Model Training Considerations**:
  The deep learning model is trained offline using publicly available PCG datasets (e.g., PhysioNet). Training is performed on systems with appropriate GPU support to ensure efficient processing.

- **Data Privacy Compliance**:
  The system follows ethical and privacy standards, especially in handling sensitive health data. Uploaded PCG files are stored securely and anonymized where needed, following **PMC and PIC** recommendations.

- **Scalability Planning**:
  The architecture is designed to support additional modules in the future, such as murmur classification or mobile app integration, without overhauling the system.

- **User Support & Feedback Loop**:
  Feedback from users (especially doctors and researchers) will be collected periodically to improve system accuracy and usability. A suggestion/feedback form is available in the system.

- **System Limitations:**

- The system does not currently support mobile devices natively.
- Real-time processing speed may vary based on the size of the audio file and internet speed.
- Final clinical diagnosis still requires validation by a medical professional.

5. Other Design Details

- **Model Selection Rationale**:
  A hybrid **CNN-LSTM architecture** was chosen based on its effectiveness in handling time-series audio data. CNNs are used for extracting spatial features from spectrograms (converted from PCG signals), while LSTMs handle sequential dependencies in heart sound patterns.
- **Dataset Source and Preprocessing**:
  The project uses open-access PCG datasets from **PhysioNet 2016**. Each audio sample undergoes:
  - **Bandpass Filtering** to reduce background noise
  - **MFCC and Spectrogram Extraction** for feature representation
  - **Normalization and Segmentation** for uniform input lengths
- **Model Evaluation Metrics**:
  Performance of the classifier is assessed using:
  - **Accuracy**
  - **Precision**
  - **Recall**
  - **F1-Score**
  - **Confusion Matrix Visualization** for classification clarity between S1 and S2
- **Validation Strategy**:
  The model is trained using an 80-10-10 **train-validation-test split**, with **K-fold cross-validation** used during the tuning phase to ensure model robustness and avoid overfitting.

Design Flexibility for Future Enhancements

- **Multi-Class Expansion**:
  The current system supports binary classification (S1 vs S2). However, the architecture allows expansion to detect:
  - Murmurs (systolic/diastolic)
  - Extra heart sounds (S3, S4)
- **Modular ML Inference Engine**:
  The model is containerized separately, enabling independent updates or hot-swapping of improved versions without affecting the frontend/backend.

- **Feedback Loop for Model Refinement**:
  Doctors can **approve or reject** model-generated insights. This feedback is logged for future retraining, enabling a form of **active learning**.

Performance Considerations

- **Model Load Time**:
  TensorFlow models are preloaded into memory at server start to minimize classification latency.

- **Audio Size and Processing Time**:
  The average processing time for a 10-second PCG clip is under **10 seconds**, depending on server specs and traffic load.

- **System Throughput**:
  The system is tested to support **up to 50 concurrent users**, with load balancing handled by container orchestration tools (if scaled).

6. Test Specification and Results

6.1 Test Case Specification

Table 6.1: TC-1

Identifier	TC-1
Related requirements(s)	The system must allow registered users to log in using valid credentials and grant access to their respective dashboards.
Short description	Verify that a registered user can successfully log in.
Pre-condition(s)	User account exists; system is running.
Input data	Valid email & password
Detailed steps	1.Open login page. 2. Enter valid credentials. 3. Click "Login".
Expected result(s)	User is authenticated and redirected to dashboard.
Post-condition(s)	User session created.
Actual result(s)	User is Logged in and landed to the Dashboard page
Test Case Result	Pass

Table 6.2: TC-2

Identifier	TC-2
Related requirements(s)	The system shall allow users to upload PCG audio files in .wav format for further processing and classification
Short description	System accepts PCG audio files.
Pre-condition(s)	User logged in.
Input data	PCG file (.wav)
Detailed steps	1. Open PCG Input Interface. 2. Click "Upload". 3. Select .wav file. 4. Submit.
Expected result(s)	File successfully uploaded and stored.
Post-condition(s)	File path stored in DB.
Actual result(s)	File Uploaded and saved in Data base
Test Case Result	Pass

Table 6.1: TC-3

Identifier	TC-3
Related requirements(s)	The system must validate uploaded PCG files for format, sampling rate, size, and audio integrity.
Short description	File uploaded.
Pre-condition(s)	Valid or invalid .wav file
Input data	Valid email & password
Detailed steps	1. Upload PCG file. 2. System checks file properties. 3. Validation result generated
Expected result(s)	Valid files accepted; invalid files rejected.
Post-condition(s)	Validation logged.
Actual result(s)	File with Valid format is accepted.
Test Case Result	Pass

Table 6.4: TC-4

Identifier	TC-4
Related requirements(s)	The system must preprocess PCG recordings by applying filtering, normalization, and feature extraction (MFCC/Spectrogram)
Short description	Test preprocessing accuracy.
Pre-condition(s)	Valid PCG file uploaded.
Input data	PCG audio file
Detailed steps	1. Send file for preprocessing. 2. Apply bandpass filter. 3. Normalize signal. 4. Generate MFCC/Spectrogram.
Expected result(s)	Cleaned audio and extracted features produced.
Post-condition(s)	Processed data ready for model.
\Actual result(s)	Cleaned data is produced with extracted features.
Test Case Result	Pass

Table 6.1: TC-5

Identifier	TC-5
Related requirements(s)	The system shall classify heart sound segments into S1 and S2 using a CNN-LSTM deep learning mode
Short description	Verify DL model prediction.
Pre-condition(s)	Preprocessed features available.
Input data	MFCC/Spectrogram
Detailed steps	1. Send features to model. 2. Model runs classification. 3. Output returned.
Expected result(s)	Model outputs class label + probability.
Post-condition(s)	Result stored.
Actual result(s)	Model outputs class label + probability.
Test Case Result	Pass

Table 6.1: TC-6

Identifier	TC-6
Related requirements(s)	"The system must save classification results, including labels and confidence scores, into the database."
Short description	Store model output in DB.
Pre-condition(s)	Inference completed.
Input data	1. Receive prediction. 2. Save result to DB. 3. Link to user & PCG record.
Detailed steps	1.Open login page. 2. Enter valid credentials. 3. Click "Login".
Expected result(s)	Result successfully stored.
Post-condition(s)	Result available for display.
Actual result(s)	Result successfully stored.
Test Case Result	Pass

Table 6.1: TC-7

Identifier	TC-7
Related requirements(s)	The system shall display classification results, waveform plots, and spectrograms to the user
Short description	Ensure results display correctly.
Pre-condition(s)	Result exists in database.
Input data	User request
Detailed steps	1. Open "View Results". 2. System fetches data. 3. Display results.
Expected result(s)	Correct, complete result shown.
Post-condition(s)	Screen updated.
Actual result(s)	Correct, complete result shown.
Test Case Result	Pass

Table 6.1: TC-8

Identifier	TC-8
Related requirements(s)	The system shall allow users to view metadata of uploaded PCG recordings
Short description	Test PCG metadata display.
Pre-condition(s)	PCG uploaded.
Input data	File ID
Detailed steps	1. Open PCG details page. 2. Backend fetches metadata. 3. Display information.".
Expected result(s)	Metadata displayed accurately.
Post-condition(s)	N/A
Actual result(s)	Results Displayed accurately
Test Case Result	Pass

Table 6.1: TC-9

Identifier	TC-9
Related requirements(s)	The system must maintain stable database connectivity to ensure reliable storage and retrieval
Short description	Verify DB connection.
Pre-condition(s)	Backend running.
Input data	DB request
Detailed steps	1. Backend sends test query. 2. DB responds.
Expected result(s)	Connection successful.
Post-condition(s)	N/A
Actual result(s)	Connection successful.
Test Case Result	Pass

Table 6.1: TC-10

Identifier	TC-10
Related requirements(s)	The system must allow users to securely log out and terminate active sessions
Short description	Test logout.
Pre-condition(s)	User logged in.
Input data	Logout request
Detailed steps	1. Click logout. 2. Session destroyed.”.
Expected result(s)	Redirect to login page.
Post-condition(s)	No active session.
Actual result(s)	User is Redirected to login page.
Test Case Result	Pass

6.2 Summary of Test Results

Table 6.1: Summary of Test Results

Module Name	Test cases run	Number of defects found	Number of defects corrected so far	Number of defects still need to be corrected
User Authentication	TC1	0	0	0
PCG Upload & Validation	TC2	1	1	0
Preprocessing & Feature Extraction	TC4	1	1	0
DL Inference (CNN-LSTM)	TC5	2	1	1
Reporting & Visualization	TC6	3	2	1
Database & Persistence (PostgreSQL)	TC9	2	1	1
Complete System	6	9	6	3

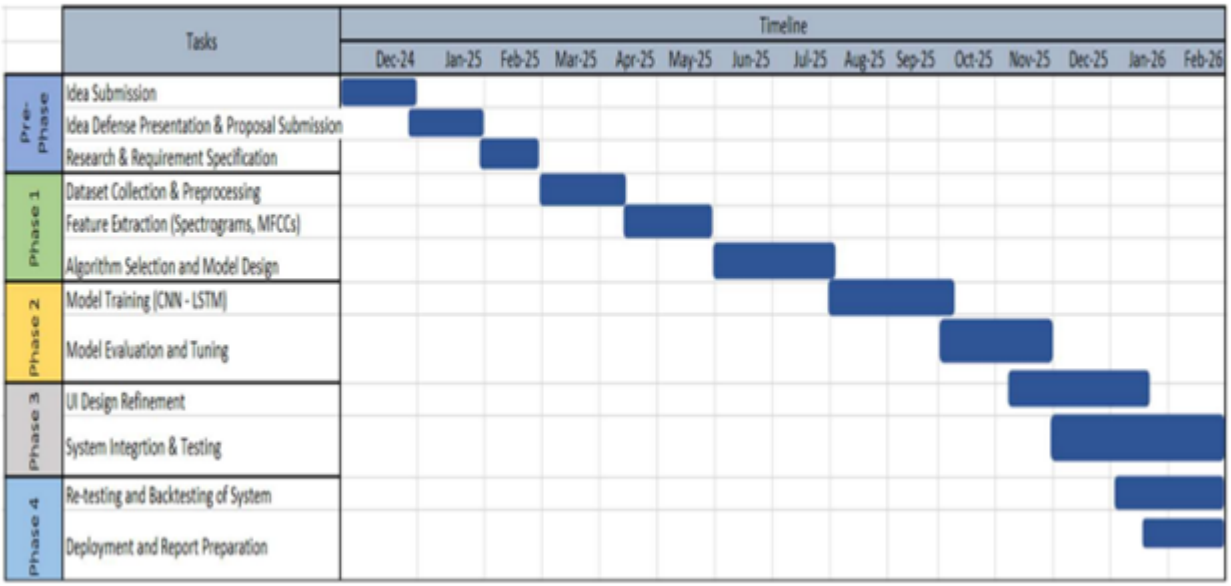
Table 6.2: Project Completion Status

Module Name	**Status** (Complete, Partially Implemented, Not Implemented)
User Authentication	Completed
PCG Upload & Validation	Completed
Preprocessing & Feature Extraction	Completed
DL Inference (CNN-LSTM)	Partially Implemented
Reporting & Visualization	Partially Implemented
Database & Persistence (PostgreSQL)	Partially Implemented
Complete System	Partially Implemented

7. Revised Project Plan

- **Ghantt Chart**

Gantt Chart:



8. References

- Minh Tuan, Nguyen & Lin, Wei & Huang, Jin. (2022). Heart Sound Classification Using Deep Learning Techniques Based on Log-mel Spectrogram. Circuits, Systems, and Signal Processing. 42. 10.1007/s00034-022-02124-1. https://www.researchgate.net/publication/362814433_Heart_Sound_Classification_Using_Deep_Learning_Techniques_Based_on_Log-mel_Spectrogram

- Shuvo SB, Ali SN, Swapnil SI et al (2021) CardioXNet: a novel lightweight deep learning framework for cardiovascular disease classification using heart sound recordings. IEEE Access 9:36955– 36967. https://www.researchgate.net/publication/349757121_CardioXNet_A_Novel_Lightweight_Deep_Learning_Framework_for_Cardiovascular_Disease_Classification_Using_Heart_Sound_Recordings

- Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., … & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215–e220. https://physionet.org/content/ltrsvp/1.0.0/

- Chen, W.; Zhou, Z.; Bao, J.; Wang, C.; Chen, H.; Xu, C.; Xie, G.; Shen, H.; Wu, H. Classifying Heart-Sound Signals Based on CNN Trained on MelSpectrum and Log-MelSpectrum Features. Bioengineering 2023, 10, 645. https://www.mdpi.com/2306-5354/10/6/645

- d'Arcy JL, Prendergast BD, Chambers JB, Ray SG, Bridgewater B. Valvular heart disease: the next cardiac epidemic. Heart. (2011) 97:91– 3. 10.1136/hrt.2010.205096. https://www.researchgate.net/publication/49675682_Valvular_heart_disease_The_next_cardiac_epidemic

- Heart sound recordings PhysioNet/CinC Challenge 2016: Training Sets. https://archive.physionet.org/pn3/challenge/2016/

- Review — Heart Sound Classification Using Deep Learning Techniques Based on Log-mel Spectrogram.https://sh-tsang.medium.com/review-heart-sound-classification-using-deeplearning-techniques-based-on-log-mel-spectrogram-9a3e5e49c8d5

- Heart Sound Classification using a Hybrid of CNN and GRU Deep Learning Models. https://www.sciencedirect.com/science/article/pii/S1877050924009724

- Audio Data Preparation and Augmentation.

  https://www.tensorflow.org/io/tutorials/audio

- https://www.researchgate.net/publication/349757121_CardioXNet_A_Novel_Lightweight_Deep_Learning_Framework_for_Cardiovascular_Disease_Classification_Using_Heart_Sound_Recordings

Appendix A: Glossary

- **Admin**

  The system administrator who manages user roles, approves dataset submissions, monitors system activity, and ensures smooth operation of the entire platform.

- **Aortic Stenosis**

  A heart condition where the aortic valve becomes narrowed, restricting blood flow from the heart. It may cause distinctive murmurs detectable in PCG recordings.

- **Classification**

  In the context of this project, it refers to the automated process of labeling heart sounds (like S1 and S2) as normal or abnormal using a trained AI model**.**

- **CNN (Convolutional Neural Network)**

  A deep learning architecture well-suited for recognizing visual and spatial patterns, used in this project for extracting features from heart sound representations like spectrograms.

- **Congestive Heart Failure (CHF)**

  A chronic condition where the heart cannot pump blood effectively, often associated with abnormal S3 or S4 sounds in PCG signals.

- **Deployment**

  The process of releasing the developed system into a live, cloud-hosted environment where users can access and interact with it in real time.

- **Doctor**

  A medical professional who uses the platform to upload, review, and interpret PCG recordings, and to generate prescriptions based on AI-supported findings.

- **Docker**

  A containerization platform used to package the application and its dependencies, enabling reliable and consistent deployment across environments.

- **Evaluation Metrics**

  Quantitative measurements such as accuracy, precision, recall, and F1 score, used to assess how well the AI model is performing in classifying heart sounds.

- **Hashed Passwords**

  A security practice where user passwords are converted into encrypted, unreadable strings before storage, ensuring data safety even in case of a breach.

- **Heart Murmur**

  An unusual sound heard during a heartbeat, usually indicative of valve issues or blood flow abnormalities, detectable using PCG recordings.

- **Hybrid Model**

  A combination of CNN and LSTM deep learning architectures. CNN extracts features, while LSTM captures temporal dependencies, together improving classification accuracy.

- **Interoperability**
  The system's ability to integrate with or communicate with other software systems, such as hospital EMRs or medical APIs.
- **LSTM (Long Short-Term Memory)**
  A type of recurrent neural network used to analyze time-based patterns in data, particularly useful for understanding sequential signals like heartbeats.
- **MFCC (Mel-Frequency Cepstral Coefficients)**
  A method of extracting meaningful features from audio signals, widely used in speech and heart sound recognition.
- **Mitral Valve Prolapse**
  A condition where the mitral valve doesn't close properly, often producing clicking or murmur sounds captured in PCG recordings.
- **Patient**
  A user who uploads their own PCG files to the system to receive classification results and downloadable health reports.
- **PCG (Phonocardiogram)**
  A digital recording of heart sounds captured via a microphone or stethoscope, used for diagnosing cardiac conditions.
- **Portability**
  The ability of the software to run in multiple environments without modification, ensured through the use of container technologies like Docker.
- **Preprocessing**
  The set of operations (such as filtering, normalization, and feature extraction) applied to raw PCG recordings before they are classified by the model.
- **RBAC (Role-Based Access Control)**
  A method of restricting access to system resources based on the user's role, ensuring that each user only accesses what they're permitted to.
- **Researcher**
  A user role with access to anonymized PCG datasets and performance results, intended to allow evaluation, feedback, and contributions to model improvement.
- **S1 & S2**
  The two primary heart sounds.
  • S1: Occurs at the beginning of systole (ventricular contraction).
  • S2: Occurs at the beginning of diastole (ventricular relaxation).

- **Security**
  The set of practices and mechanisms that protect the system from unauthorized access, data breaches, and misuse of sensitive health data.
- **Spectrogram**
  A visual representation of the frequency content of an audio signal over time, used to better understand and classify heart sound patterns
- **Testability**
  The ease with which the system can be tested for bugs and accuracy, often supported by automated unit tests and coverage metrics.
- **Two**-Factor Authentication (2FA)
  An additional layer of login security where users must enter a one-time code (e.g., from email or SMS) along with their password.
- **Usability**
  The overall friendliness and ease-of-use of the system's interface, particularly important for non-technical users such as patients

Appendix B: IV & V Report

(Independent verification & validation)
IV & V Resource

Name Signature

S#	Defect Description	Origin Stage	Status	Fix Time	
				Hours	Minutes
1					
2					
3					
…					

Table 1: List of non-trivial defects

This document has been adapted from the following:

- Previous project templates at UCP
- High-level Technical Design, Centers for Medicare & Medicaid Services. (www.cms.gov)