Style Guidelines for Final Year Project ReportsCardioGraph Pro

Final Year Project – Pre Report

Session 2021-2025

A 4th Year Student

A project submitted in partial fulfilment of the

COMSATS University Degree

of

BSc. (Hons.)BS in Computer Science (CUI)



Department of Computer Science

COMSATS University Islamabad, Lahore Campus

06 November 2024

# Project Detail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type (Nature of project) | | | [ ] **D**evelopment [ ] **R**esearch [✓] **R**&**D** | | |
| Area of specialization | | | Machine Learning | | |
| **Project Group Members** | | | | | |
| Sr.# | Reg. # | Student Name | | Email ID | \*Signature |
| (i) | SP21-BCS-007 | Asad Ali | | sp21-bcs-007 @cuilahore.edu.pk |  |
| (ii) | SP21-BCS-003 | Asad ur Rehman | | sp21-bcs-003 @cuilahore.edu.pk |  |
| (iii) | SP21-BCS-017 | Muhammad Haroon Shahzad | | sp21-bcs-017 @cuilahore.edu.pk |  |

\*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

# Plagiarism Free Certificate

This is to certify that, I am Asad Ali S/D/o Ghaffar Ahmad, group leader of FYP under registration no CIIT/SP21-BCS-007/LHR at Computer Science Department, COMSATS University Islamabad, Lahore Campus. I declare that my FYP report is checked by my supervisor and the similarity index is 8% that is less than 20%, an acceptable limit by HEC. Report is attached herewith as Appendix A.

A black background with letters

Description automatically generatedDate: 11-03-2024 Name of Group Leader: Asad Ali Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Supervisor: Muhammad Aksam Iftikhar Co-Supervisor (if any): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Designation: Associate Professor Designation: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Abstract**

Heart disease is one of the major problems worldwide today. Effective early detection methods are needed for timely interventions. This project presents a comprehensive strategy and techniques to predict heart disease through the detailed analysis of electrocardiogram signals, and ECG image reports. Developing an advanced machine learning model by combining demographic information with ECG signal data, exploring ECG signal processing, and converting ECG image-based reports to signals. The project aims to make disease predictions more accurate and develop machine learning models to predict heart diseases accurately. The Project mainly focused on combining demographic information like age, gender, and medical history with their heart rhythm recordings to improve prediction. The project also works with image-based ECG signals and establishes a separate pipeline to convert ECG image-based reports into data that computers can understand. Once trained, these machine learning models can be deployed on the cloud service, and then seamlessly integrated into user friendly web interface. The web interface makes it easy to upload bulk ECG data and provides essential features like user authentication, keeping track of patient information, and seeing past predictions. The inclusion of a medical chatbot within web interface assists users to interpret results and offer valuable prescriptions, medical guidance, and diet plans. The mobile app also helps users to upload their image-based reports and see the predictions. In essence, This project represents a significant step forward in advancing heart disease prediction methodologies and offering scalable and user-friendly solutions.

**Table of Contents**

[Chapter 01 Introduction 10](#_Toc166700221)

[1 Introduction 11](#_Toc166700222)

[1.1 Introduction 11](#_Toc166700223)

[1.2 Objectives 11](#_Toc166700224)

[1.3 Problem Statement 12](#_Toc166700225)

[1.4 Assumptions and Constraints 12](#_Toc166700226)

[1.5 Project Scope 13](#_Toc166700227)

[Chapter 02 Requirements Analysis 15](#_Toc166700228)

[2 Requirements Analysis 16](#_Toc166700229)

[2.1 Literature Review 16](#_Toc166700230)

[2.2 Stakeholders List (Actors) 17](#_Toc166700231)

[2.3 Requirements Elicitation 18](#_Toc166700232)

[2.4 Use Case Descriptions 25](#_Toc166700233)

[2.5 Use Case Design 32](#_Toc166700234)

[2.6 Software Development Life Cycle Model 42](#_Toc166700235)

[Chapter 03: System Design 45](#_Toc166700236)

[3 System Design 46](#_Toc166700237)

[3.1 Work Breakdown Structure (WBS) 46](#_Toc166700238)

[3.2 Activity Diagram 47](#_Toc166700239)

[3.3 Sequence Diagram 67](#_Toc166700240)

[3.4 Software Architecture 84](#_Toc166700241)

[3.5 Class Diagram 85](#_Toc166700242)

[3.6 Database Diagram 86](#_Toc166700243)

[3.7 Network Diagram (Gantt Chart) 87](#_Toc166700244)

[Chapter 4 System Testing 88](#_Toc166700245)

[4 System Testing 89](#_Toc166700246)

[4.1 Test Cases 89](#_Toc166700247)

[4.2 Test Types 96](#_Toc166700248)

[Chapter 5 Conclusion 98](#_Toc166700249)

[5 Conclusion 99](#_Toc166700250)

[5.1 Problems Faced and Lessons Learned 99](#_Toc166700251)

[5.2 Project Summary 99](#_Toc166700252)

[5.3 Future Work 100](#_Toc166700253)

[Chapter 06 References 101](#_Toc166700254)

[6 References 102](#_Toc166700255)

**List of Tables**

[Table 1: FR-001 17](#_Toc166699365)

[Table 2: FR-002 17](#_Toc166699366)

[Table 3: FR-003 17](#_Toc166699367)

[Table 4: FR-004 17](#_Toc166699368)

[Table 5: FR-005 18](#_Toc166699369)

[Table 6: FR-006 18](#_Toc166699370)

[Table 7: FR-007 18](#_Toc166699371)

[Table 8: FR-008 18](#_Toc166699372)

[Table 9: FR-009 19](#_Toc166699373)

[Table 10: FR-010 19](#_Toc166699374)

[Table 11: FR-011 19](#_Toc166699375)

[Table 12: FR-012 19](#_Toc166699376)

[Table 13: FR-013 20](#_Toc166699377)

[Table 14: FR-014 20](#_Toc166699378)

[Table 15; FR-015 20](#_Toc166699379)

[Table 16: FR-016 21](#_Toc166699380)

[Table 17: FR-017 21](#_Toc166699381)

[Table 18: FR-018 21](#_Toc166699382)

[Table 19: FR-019 21](#_Toc166699383)

[Table 20: FR-020 22](#_Toc166699384)

[Table 21: Traceability Matric 23](#_Toc166699385)

[Table 22: UC-001 24](#_Toc166699386)

[Table 23: UC-002 24](#_Toc166699387)

[Table 24: UC-003 24](#_Toc166699388)

[Table 25: UC-004 25](#_Toc166699389)

[Table 26: UC-005 25](#_Toc166699390)

[Table 27: UC-006 25](#_Toc166699391)

[Table 28: UC-007 26](#_Toc166699392)

[Table 29: UC-008 26](#_Toc166699393)

[Table 30: UC-009 26](#_Toc166699394)

[Table 31: UC-010 27](#_Toc166699395)

[Table 32: UC-011 27](#_Toc166699396)

[Table 33: UC-012 27](#_Toc166699397)

[Table 34: UC-013 28](#_Toc166699398)

[Table 35: UC-014 28](#_Toc166699399)

[Table 36: UC-015 28](#_Toc166699400)

[Table 37: UC-016 29](#_Toc166699401)

[Table 38: UC-017 29](#_Toc166699402)

[Table 39: UC-018 29](#_Toc166699403)

[Table 40: UC-019 30](#_Toc166699404)

[Table 41: UC-020 30](#_Toc166699405)

[Table 42: Network Diagram (Gantt Chart) 86](#_Toc166699406)

[Table 43: TC-001 88](#_Toc166699407)

[Table 44: TC-002 88](#_Toc166699408)

[Table 45: TC-003 89](#_Toc166699409)

[Table 46: TC-004 89](#_Toc166699410)

[Table 47: TC-005 89](#_Toc166699411)

[Table 48: TC-006 90](#_Toc166699412)

[Table 49: TC-007 90](#_Toc166699413)

[Table 50: TC-008 90](#_Toc166699414)

[Table 51: TC-009 91](#_Toc166699415)

[Table 52: TC-010 91](#_Toc166699416)

[Table 53: TC-011 91](#_Toc166699417)

[Table 54: TC-012 92](#_Toc166699418)

[Table 55: TC-013 92](#_Toc166699419)

[Table 56: TC-014 93](#_Toc166699420)

[Table 57: TC-015 93](#_Toc166699421)

[Table 58: TC-016 93](#_Toc166699422)

[Table 59: TC-017 94](#_Toc166699423)

[Table 60: TC-018 94](#_Toc166699424)

[Table 61: TC-019 94](#_Toc166699425)

[Table 62: TC-020 95](#_Toc166699426)

**List of Figures**

[Figure 1: UC-001 32](#_Toc166699428)

[Figure 2: UC-002 32](#_Toc166699429)

[Figure 3: UC-003 33](#_Toc166699430)

[Figure 4: UC-004 33](#_Toc166699431)

[Figure 5: UC-005 34](#_Toc166699432)

[Figure 6: UC-006 34](#_Toc166699433)

[Figure 7: UC-007 35](#_Toc166699434)

[Figure 8: UC-008 35](#_Toc166699435)

[Figure 9: UC-009 36](#_Toc166699436)

[Figure 10: UC-010 36](#_Toc166699437)

[Figure 11: UC-011 37](#_Toc166699438)

[Figure 12: UC-012 37](#_Toc166699439)

[Figure 13: UC-013 38](#_Toc166699440)

[Figure 14: UC-014 38](#_Toc166699441)

[Figure 15: UC-015 39](#_Toc166699442)

[Figure 16: UC-016 39](#_Toc166699443)

[Figure 17: UC-017 40](#_Toc166699444)

[Figure 18: UC-018 40](#_Toc166699445)

[Figure 19: UC-019 41](#_Toc166699446)

[Figure 20: UC-020 41](#_Toc166699447)

[Figure 21: The Waterfall Model 42](#_Toc166699448)

[Figure 22: Work Breakdown Structure (WBS) 46](#_Toc166699449)

[Figure 23: Activity - User Sign Up 47](#_Toc166699450)

[Figure 24: Activity - User Login 48](#_Toc166699451)

[Figure 25: Activity - User Recognition 49](#_Toc166699452)

[Figure 26: Activity - Initiate Prediction Process 50](#_Toc166699453)

[Figure 27: Activity - Upload ECG Signal Data 51](#_Toc166699454)

[Figure 28: Activity - ECG Signal Visualization 52](#_Toc166699455)

[Figure 29: Activity - ECG Disease Prediction 53](#_Toc166699456)

[Figure 30: Activity - ECG Image Upload 54](#_Toc166699457)

[Figure 31: Activity - ECG Image Disease Prediction 55](#_Toc166699458)

[Figure 32: Activity - Prediction Display 56](#_Toc166699459)

[Figure 33: Activity - Chat Bot Integration 57](#_Toc166699460)

[Figure 34: Activity - Medical History Tracking 58](#_Toc166699461)

[Figure 35: Activity - Doctor Browsing and Filtering 59](#_Toc166699462)

[Figure 36: Activity - Appointment Scheduling 60](#_Toc166699463)

[Figure 37: Activity - Doctor Profile Management 61](#_Toc166699464)

[Figure 38: Activity - Appointment Management 62](#_Toc166699465)

[Figure 39: Activity - Doctor Patient Interaction 63](#_Toc166699466)

[Figure 40: Activity - Review and Rating System 64](#_Toc166699467)

[Figure 41: Activity - Admin Panel 65](#_Toc166699468)

[Figure 42: Activity - E-commerce Integration 66](#_Toc166699469)

[Figure 43: Sequence - User Sign Up 67](#_Toc166699470)

[Figure 44: Sequence - User Login 68](#_Toc166699471)

[Figure 45: Sequence - User Recognition 69](#_Toc166699472)

[Figure 46: Sequence - Initiate Prediction Process 70](#_Toc166699473)

[Figure 47: Sequence - Upload ECG Signal Data 71](#_Toc166699474)

[Figure 48: Sequence - ECG Signal Visualization 72](#_Toc166699475)

[Figure 49: Sequence - ECG Disease Prediction 73](#_Toc166699476)

[Figure 50: Sequence - ECG Image Upload 73](#_Toc166699477)

[Figure 51: Sequence - ECG Image Disease Prediction 74](#_Toc166699478)

[Figure 52: Sequence - Prediction Display 74](#_Toc166699479)

[Figure 53: Sequence - Chat Bot Integration 75](#_Toc166699480)

[Figure 54: Sequence - Medical History Tracking 76](#_Toc166699481)

[Figure 55: Sequence - Doctor Browsing and Filtering 77](#_Toc166699482)

[Figure 56: Sequence - Appointment Scheduling 78](#_Toc166699483)

[Figure 57: Sequence - Doctor Profile Management 79](#_Toc166699484)

[Figure 58: Sequence - Appointment Management 80](#_Toc166699485)

[Figure 59: Sequence - Doctor Patient Interaction 80](#_Toc166699486)

[Figure 60: Sequence - Review and Rating System 81](#_Toc166699487)

[Figure 61: Sequence - Admin Panel 82](#_Toc166699488)

[Figure 62: Sequence - E-commerce Integration 83](#_Toc166699489)

[Figure 63: Software Architecture 84](#_Toc166699490)

[Figure 64: Class Diagram 85](#_Toc166699491)

[Figure 65: Database Diagram 86](#_Toc166699492)

Chapter 01 Introduction

# Introduction

## Introduction

Early detection of heart disease is important and beneficial for heart patients to get help before any serious circumstances. ECG signal data provide useful insights into heart health. These insights can be used for the prediction of heart diseases in the early stages [1]. CardioGraph Pro is specially trained to predict diseases on such data using machine learning techniques to achieve better health treatment of heart patients [2].

This project mainly focusses to establish a robust data pipeline that have potential to integrate demographic data and Electrocardiogram signals together [3]. The system will take raw data that have both type of information and turn into that format which can be easy to use for analysis and prediction. The project will use the PhysioNet's Database (WFDB) for ECG signals data [4]. Then that data converts into CSV files for further processing.

After organising data and preprocessing it, the project will use machine learning techniques to give predictions about heart diseases based on ECG signals [5]. The model is tested thoroughly to make sure that it will predict accurately and efficiently [6]. The project will also explore image-based ECG signal processing, developing a separate pipeline to convert ECG images into analysable signals [7]. These image-based signals will undergo the same predictive modelling process as conventional ECG data, enriching the predictive capabilities of the models.

Upon successful training and validation, the trained machine learning model will be deployed on cloud service for scalability and accessibility. And it will relate to a website to use. This way the doctor and the regular person can upload their heart rhythm data to get predictions [8]. The website will even have a chatbot to help explain the result nicely This project also consists of a mobile app for those who can simply upload their ECG image to predict results quickly [9].

## Objectives

The primary aim of this project is to develop an advanced system for heart disease prediction that integrates ECG signals analysis and image ECG image-based report processing and provides a user-friendly interface to get accurate predictions about heart diseases.

Making it helpful for the doctors to get predicted disease labels along with ECG reports for several patients in a very sophisticated and rebuts way. It also aims to help a patient having any ECG report to get know about his disease, get prescriptions, and diet plans accordingly also integrates a chat bot to ask questions about the disease or answer any concern of the patient.

* Design and construct the data pipeline that combines ECG signals and demographic data from the PhysioNet WFDB database.
* Make the pipeline that converts ECG images into the data (signals) that the model can understand.
* Train test and optimize the machine learning model to accurately and precisely predict heart disease based on ECG signals and image-based reports.
* Training a model to give prescriptions based on the predicted disease and have diet plan recommendations.
* Deploy the optimized prediction model on a cloud platform to ensure scalability and accessibility.
* Build the website and app where the user can upload their heart data, safely log in manage their record, and see medical history.
* Include the medical chatbot (only for the website). that can explain the result answer the question and give them basic guidelines.

## Problem Statement

Through conducting this research and developing this solution, our main learning will be a depth understanding of how to train test and deploy machine learning models (multi-label classification) along with the integration process for web and mobile app. One of our main learnings will be understanding data preprocessing, data pipeline designing and working with the imaging data. Also, we will learn how to deploy and maintain a machine learning model to keep it working for the user on our website and mobile app.

## Assumptions and Constraints

The assumptions and constraints of the system are as follows:

### Assumptions:

* The ECG signals obtained from the PhysioNet WFDB database are accurate and reliable representations of heart activity.
* Demographic data associated with ECG signals are complete and correctly matched with the corresponding signals.
* The machine learning model trained on ECG signals will generalize well to unseen data and accurately predict heart diseases.
* Users will provide authentic and relevant information when uploading their heart data through the website or mobile app.
* The cloud platform chosen for deployment will offer sufficient scalability, reliability, and security for hosting the prediction model.
* Users accessing the website or mobile app have a basic understanding of navigating digital interfaces and interpreting medical information.
* The medical chatbot will provide accurate and helpful responses based on the predicted results and general medical knowledge.

### Constraints:

* Limited computational resources may restrict the size and complexity of the machine learning model that can be trained and deployed.
* The availability and accessibility of the PhysioNet WFDB database may be subject to restrictions or limitations imposed by the database administrators.
* Compliance with data privacy regulations and standards, such as GDPR or HIPAA, must be ensured when handling sensitive medical data.
* Development timelines and budgets may impose constraints on the scope and features of the website and mobile app.
* Compatibility issues may arise when integrating different technologies and platforms, requiring careful planning, and testing during implementation.
* User engagement and adoption may be influenced by factors such as ease of use, user interface design, and the perceived reliability of the prediction model.
* Maintenance and updates of the deployed machine learning model and software applications will require ongoing resources and support.

## Project Scope

The scope of the project encompasses the following key aspects:

### Data Pipeline Development:

* + Designing and constructing a robust data pipeline to integrate demographic data and Electrocardiogram (ECG) signals obtained from the PhysioNet WFDB database.
  + Developing additional pipeline functionality to convert ECG images into analyzable signals for enhanced predictive modeling.

### Machine Learning Model Training and Optimization:

* + Utilizing machine learning techniques to train, test, and optimize a predictive model capable of accurately diagnosing heart diseases based on ECG signals and image-based reports.
  + Incorporating multi-label classification to predict multiple heart conditions simultaneously, enabling comprehensive health assessments.

### Medical Guidance and Recommendations:

* + Enhancing the model's functionality to provide personalized prescriptions and diet plan recommendations based on predicted diseases, promoting proactive health management.

### Website and Mobile App Development:

* + Building user-friendly interfaces for the website and mobile app to enable users to securely log in, manage their records, and access their medical history.
  + Implementing features such as data visualization, interactive dashboards, and real-time notifications to enhance user experience and engagement.

### Medical Chatbot Integration:

* + Incorporating a medical chatbot within the website to provide users with explanations of predicted results, answer queries, and offer basic medical guidance.
  + Ensuring accuracy and reliability of the chatbot's responses through continuous training and validation against medical knowledge bases.

### Testing, Validation, and Maintenance:

* + Conducting thorough testing and validation of the entire system to ensure reliability, accuracy, and compliance with regulatory standards such as GDPR and HIPAA.
  + Implementing mechanisms for ongoing maintenance, updates, and support to address emerging issues, enhance performance, and adapt to evolving user needs and technological advancements.

Chapter 02 Requirements Analysis

# Requirements Analysis

## Literature Review

[10] This study investigates the analysis of ECG signals for the purpose of detecting cardiovascular illness using a convolutional neural network (CNN). This research paper presents a Deep Neural Network model that was trained over the Two million labelled tests that were gathered as part of the Clinical Outcomes in Digital Electro cardiology study and assessed by the Telehealth Network of Minas Gerais.  With F1 scores is above 80% and specificity almost over 99% for the diagnosis of atrial fibrillation or normal rhythm.

[11] This study investigates the classification of Electrocardiogram signals using a Radial Basis Function (RBF) Neural Network for the detection of cardiac disease. The data used in this study report came from a PTB-XL database. The researcher proposed the three neural network architecture approaches the first one based on the convolution network the second on the SincNet and the third was mix of first type with additional entropy-based features. Comparably, the network built on top of SincNet recognized two classes with 85.8% ACC and five classes with 73.0% ACC. 89.82% ACC was attained by the network that based on the convolutional network with entropy features when two classes were recognized, and 76.5% ACC when five classes were recognized.

[12] This paper tried a hybrid deep learning model for heart disease prediction using ECG signals. It combines conventional ECG beat extraction with a Long Short-Term Memory (LSTM) network. This article's goal was to present a novel approach to categorizing heart disease from Electrocardiogram data. They developed a system that maintains critical heart wave data while enabling automation. CNN's automatic feature learning produces an accurate representation of heart function through adaptive heartbeat segmentation. It contributes to a decrease in incorrect classifications. The FSN approach provides a more accurate and dependable feature set for the classification of cardiac diseases since the features vector is created using Convolution neural network and QRS complex features. The experimental findings demonstrate that the proposed model outperforms the previous deep learning-based techniques. To determine how dependable the performance of the suggested model is, we advise examining further data sets in the future.

[13] In this study, they provide a novel approach that combines deep learning modelling, signal processing techniques, and cardiology to predict cardiac illnesses using ECG signals. To turn 1-D signals into 2-D images, they used wavelet transformation, which enables deep learning models.to examine the properties of the signal's various frequency bands concurrently. Their system is at the cutting edge and is quick and easy to use. Owing to their system's outstanding performance in the four ECG signal classification challenges, we anticipate that this approach will be applicable to increasingly challenging tasks. We'll test this with bigger datasets that have more detailed annotations. Furthermore, our method performs differently because different wavelets can extract different frequency characteristics utilizing the wavelet transform. We'll try to apply this model on other wavelets to process signals.

[14] This project aims to create algorithmic models for the analysis of ECG tracings to forecast cardiovascular illnesses. This project helps save lives and improve health care quality while losing cost. Improving medical care and saving lives are the immediate results of this study, which is particularly relevant as global health care and insurance prices rise. To optimize deep-learning parameters, we carried out several tests. Two different machine learning algorithms (VGG16 and MobileNetV2) tested, and both were performed very accurate to gives 95% result. When the model is put on Raspberry Pi, they are still accurate but slightly less do 94% for MobileNetV2 and 90 % for VGG16he main goal of this project is to improve how we can keep track of the health information in real time using smart devices.

### Existing Systems and Projects

Cardiograph Pro introduces a lightweight RNN architecture for analyzing heart disease from ECG images and ECG signals. The model achieves 84% f1 score for normal class while running efficiently on a single CPU, making it suitable for resource-limited devices and potential integration with Internet of Things (IoT) applications. Extracted features from the model also enhance traditional machine learning algorithms.

### Summary of Existing Studies and Systems

Existing studies and systems highlight the effectiveness of deep learning models, particularly CNNs, in analyzing ECG signals for cardiovascular disease detection. These approaches leverage large datasets and advanced techniques to achieve high accuracy rates. Additionally, lightweight architectures like the one proposed in Cardiograph Pro show promise for efficient and accurate heart disease diagnosis, paving the way for improved healthcare solutions.

## Stakeholders List (Actors)

**Project Supervisors:** The project supervisors oversee and guide the development process, providing direction, feedback, and support to ensure the project's success.

**Developers:** Responsible for the implementation of the project, developers design, code, test, and maintain the software according to the set requirements and guidelines.

**Users/Artists:** Primary stakeholders who interact with the platform to generate and share artworks using Neural Style Transfer. Their feedback shapes the platform's features and usability.

**Healthcare Professionals:** Provide domain expertise, requirements, and feedback if the project involves healthcare applications.

**Regulatory Bodies:** Set standards, regulations, and guidelines that the project must adhere to, particularly in fields such as healthcare or finance.

## Requirements Elicitation

### Functional Requirements

#### FR-001: User Sign Up

Table 1: FR-001

|  |  |
| --- | --- |
| Requirement ID | FR-001 |
| Description | The system shall provide a user sign-up mechanism to allow new users to register for an account. |
| Acceptance Criteria | New users must be able to register by providing a valid email address and creating a password. |
| Priority | High |
| Dependencies | Implement a user registration form with email verification functionality. |

#### FR-002: User Login

Table 2: FR-002

|  |  |
| --- | --- |
| Requirement ID | FR-002 |
| Description | The system shall provide a user login mechanism to allow registered users to access the system. |
| Acceptance Criteria | Registered users must be able to log in using their email address and password. Upon successful login, users should be directed to their dashboard. |
| Priority | High |
| Dependencies | Implement email and password authentication for user login. |

#### FR-003: User Recognition

Table 3: FR-003

|  |  |
| --- | --- |
| Requirement ID | FR-003 |
| Description | The system determines a user's role, either by extracting specific fields from the signup form, such as user type or role, or by querying the database for the user's role based on the provided credentials. |
| Acceptance Criteria | Upon login or signup, the system distinguishes between doctors and patients, granting them appropriate access rights. |
| Priority | High |
| Dependencies | User authentication mechanism (FR-001 & FR-002) |

#### FR-004: Initiate Prediction Process

Table 4: FR-004

|  |  |
| --- | --- |
| Requirement ID | FR-004 |
| Description | After logging in, the system shall allow users to click on a "Get Prediction" button to initiate the prediction process. |
| Acceptance Criteria | Users should be able to initiate the prediction process by clicking on the "Get Prediction" button. |
| Priority | High |
| Dependencies | Successful implementation of user login functionality and integration of the "Get Prediction" button in the user interface. |

#### FR-005: Upload ECG Signal Data

Table 5: FR-005

|  |  |
| --- | --- |
| Requirement ID | FR-005 |
| Description | After clicking the "Get Prediction" button, users should be prompted to upload their ECG signal data. |
| Acceptance Criteria | Upon clicking the "Get Prediction" button, a prompt should appear allowing users to upload their ECG signal data. After successful upload, the system should begin processing the prediction. |
| Priority | High |
| Dependencies | Implementation of file upload functionality for ECG signal data is required. |

#### FR-006: ECG Signal Visualization

Table 6: FR-006

|  |  |
| --- | --- |
| Requirement ID | FR-006 |
| Description | The system shall provide functionality to visually represent ECG signals in the form of an ECG graph. |
| Acceptance Criteria | Users should be able to view ECG signals graphically plotted on the system interface. The ECG graph should accurately depict the amplitude and frequency characteristics of the ECG signals. Users must be able to adjust the scale and parameters of the ECG graph for detailed analysis. |
| Priority | High |
| Dependencies | Availability of ECG signal data for visualization and integration with graph plotting libraries or tools for generating ECG graphs. |

#### FR-007: ECG Disease Prediction

Table 7: FR-007

|  |  |
| --- | --- |
| Requirement ID | FR-007 |
| Description | The system shall predict heart disease by executing a machine learning model on the provided ECG data. |
| Acceptance Criteria | Users should be able to upload ECG data, triggering the system to analyse the data using the machine learning model. The predictions generated by the model regarding potential cardiac abnormalities or conditions must be presented to the users clearly and accurately. |
| Priority | High |
| Dependencies | Integration of machine learning models for ECG data analysis and prediction is essential. |

#### FR-008: ECG Image Upload

Table 8: FR-008

|  |  |
| --- | --- |
| Requirement ID | FR-008 |
| Description | The system shall allow users to upload ECG image for prediction of heart disease. |
| Acceptance Criteria | Users must be able to select and upload ECG images from their device. The system should accept common image formats such as JPEG, PNG, and JPG. Upon successful upload, the system should display a confirmation message to the user. |
| Priority | High |
| Dependencies | Implementation of image processing functionality to extract relevant information from uploaded images. |

#### FR-009: ECG Image Disease Prediction

Table 9: FR-009

|  |  |
| --- | --- |
| Requirement ID | FR-009 |
| Description | The system shall analyse uploaded ECG images to predict heart disease using machine learning algorithms. |
| Acceptance Criteria | Uploaded ECG images should be processed by the system's machine learning model to generate predictions regarding potential cardiac abnormalities or conditions. |
| Priority | High |
| Dependencies | Integration of machine learning models capable of analysing ECG images for heart disease prediction. |

#### FR-010: Prediction Display

Table 10: FR-010

|  |  |
| --- | --- |
| Requirement ID | FR-010 |
| Description | The system shall present the predictions of heart disease to the user in a clear and understandable format. |
| Acceptance Criteria | After the ML model has generated predictions based on the provided data, the system should display the results to the user in an intuitive manner on the system interface. |
| Priority | High |
| Dependencies | Successful execution of the ML model for heart disease prediction and integration of prediction results with the user interface for display. |

#### FR-011: Chat Bot Integration

Table 11: FR-011

|  |  |
| --- | --- |
| Requirement ID | FR-012 |
| Description | The system shall integrate a chat bot feature capable of performing prediction analysis, addressing patient queries, providing prescriptions, and suggesting diet plans. |
| Acceptance Criteria | The chat bot should be able to interact with users in a conversational manner, understanding natural language queries related to prediction analysis, health concerns, medication inquiries, and dietary requirements. |
| Priority | High |
| Dependencies | Integration of prediction analysis module, medical knowledge databases, and nutritional information sources with the chat bot interface. |

#### FR-012: Medical History Tracking

Table 12: FR-012

|  |  |
| --- | --- |
| Requirement ID | FR-013 |
| Description | The system shall allow users to track their medical history. |
| Acceptance Criteria | The system must securely store and display this information to users upon request. |
| Priority | Medium |
| Dependencies | Implement a medical history form with appropriate fields for user input. |

#### FR-013: Doctor Browsing and Filtering

Table 13: FR-013

|  |  |
| --- | --- |
| Requirement ID | FR-014 |
| Description | The system shall allow users to browse healthcare professionals and apply filters based on timing and rating to find the perfect doctor. |
| Acceptance Criteria | Users should be able to access a directory of healthcare professionals, filter them based on available timings and ratings, and view detailed profiles of each professional, including their qualifications, specialties, clinic/hospital affiliations, and patient reviews. |
| Priority | High |
| Dependencies | Implementation of a comprehensive database of healthcare professionals with associated attributes such as availability schedules and ratings. Development of a user-friendly interface for browsing and filtering professionals based on specified criteria. |

#### FR-014: Appointment Scheduling

Table 14: FR-014

|  |  |
| --- | --- |
| Requirement ID | FR-015 |
| Description | The system shall provide functionality for users to schedule appointments with healthcare professionals. |
| Acceptance Criteria | Users should be able to view the availability of healthcare professionals, select preferred appointment slots, and confirm appointments. The system must notify both users and professionals about scheduled appointments. |
| Priority | High |
| Dependencies | Implement a scheduling system that integrates with user and professional calendars. |

#### FR-015: Doctor Profile Management

Table 15; FR-015

|  |  |
| --- | --- |
| Requirement ID | FR-016 |
| Description | Doctors can update personal info, qualifications, specialties, clinic affiliations, and availability. They can also respond to patient reviews. |
| Acceptance Criteria | Doctors should be able to create and manage their profiles on the platform. This includes adding personal information, qualifications, areas of specialization, availability schedule, and contact details. The profile management interface should be intuitive and allow doctors to update their information easily. |
| Priority | High |
| Dependencies | Implementation of user authentication system to ensure secure access for doctors. Development of a database system to store and manage doctor profiles securely. Integration with the platform's user interface to display doctor profiles to patients. |

#### FR-016: Appointment Management

Table 16: FR-016

|  |  |
| --- | --- |
| Requirement ID | FR-017 |
| Description | Doctors set availability, patients book appointments accordingly. System handles conflicts and sends confirmation notifications. |
| Acceptance Criteria | Doctors should be able to view their appointment schedules and manage appointments efficiently. This includes the ability to add, edit, or cancel appointments as needed. |
| Priority | High |
| Dependencies | Integration with the doctor's availability schedule specified in their profile. |

#### FR-017: Doctor Patient Interaction

Table 17: FR-017

|  |  |
| --- | --- |
| Requirement ID | FR-018 |
| Description | Patients message doctors for inquiries, follow-ups, or updates. Doctors respond promptly. |
| Acceptance Criteria | Doctors should be able to communicate with patients through the messaging system. This includes responding to patient inquiries, providing medical advice, and sharing relevant information during appointment. |
| Priority | High |
| Dependencies | Implementation of secure messaging infrastructure to protect patient-doctor communication. Integration with the platform's user interface to facilitate seamless interaction between doctors and patients. Development of file-sharing functionality to enable the exchange of medical documents or images. |

#### FR-018: Review and Rating System

Table 18: FR-018

|  |  |
| --- | --- |
| Requirement ID | FR-019 |
| Description | The system shall incorporate a review and rating system to gather feedback from patients regarding their experiences with healthcare professionals. |
| Acceptance Criteria | After each appointment or consultation, patients should be prompted to provide feedback and ratings based on their experience with the healthcare professional. |
| Priority | Medium |
| Dependencies | Integration with the appointment scheduling system to link reviews and ratings with specific appointments or consultations. Implementation of data storage and retrieval mechanisms to store patient feedback securely. |

#### FR-019: Admin Panel

Table 19: FR-019

|  |  |
| --- | --- |
| Requirement ID | FR-020 |
| Description | The system will provide comprehensive Doctor Monitoring and Control feature within the admin panel, enabling administrators to supervise and manage doctor activities effectively. |
| Acceptance Criteria | The admin panel should provide comprehensive monitoring and control functionalities to oversee doctor activities and enforce platform policies. Administrators should be able to view detailed activity logs and performance metrics for individual doctors, including appointment acceptance rates, patient feedback, and attendance records. |
| Priority | Medium |
| Dependencies | Integration with the platform's database to retrieve and analyse doctor activity data, including appointment logs, feedback ratings, and other relevant metrics. |

### Non-Functional Requirements

#### Performance:

* + **Response Time**: The system shall respond to user actions within 2 seconds on average, ensuring a smooth user experience.
  + **Scalability**: The system should be able to handle an increasing number of users and data volume without significant degradation in performance. It should scale horizontally to accommodate additional computational resources as needed.

#### Reliability:

* + **Availability**: The system shall be available 99.9% of the time, allowing for scheduled maintenance windows.
  + **Fault Tolerance**: The system should gracefully handle errors and failures, ensuring that critical functionalities remain available even in the event of component failures.
  + **Data Integrity**: Data stored within the system should be accurate and consistent, with mechanisms in place to prevent data corruption or loss.

#### Security:

* + **Authentication and Authorization**: User authentication should be secure, utilizing industry-standard encryption protocols. Access to sensitive data should be restricted based on user roles and permissions.
  + **Data Privacy**: Personal health information (PHI) and other sensitive data should be encrypted both in transit and at rest, adhering to relevant data protection regulations such as GDPR or HIPAA.
  + **Protection against Attacks**: The system should implement measures such as input validation, SQL injection prevention, and cross-site scripting (XSS) protection to mitigate common security threats.

### Requirements Traceability Matric

Table 21: Traceability Matric

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Requirement No. | Description | Use Case ID | Activity  Diagram ID | Sequence Diagram ID | Test Case ID |
| FR-001 | User Sign Up | UC-001 | 3.2.1 | 3.3.1 | TC001 |
| FR-002 | User Login | UC-002 | 3.2.2 | 3.3.2 | TC002 |
| FR-003 | User Recognition | UC-003 | 3.2.3 | 3.3.3 | TC003 |
| FR-004 | Initiate Prediction Process | UC-004 | 3.2.4 | 3.3.4 | TC004 |
| FR-005 | Upload ECG Signal Data | UC-005 | 3.2.5 | 3.3.5 | TC005 |
| FR-006 | ECG Signal Visualization | UC-006 | 3.2.6 | 3.3.6 | TC006 |
| FR-007 | ECG Disease Prediction | UC-007 | 3.2.7 | 3.3.7 | TC007 |
| FR-008 | ECG Image Upload | UC-008 | 3.2.8 | 3.3.8 | TC008 |
| FR-009 | ECG Image Disease Prediction | UC-009 | 3.2.9 | 3.3.9 | TC009 |
| FR-010 | Prediction Display | UC-010 | 3.2.10 | 3.3.10 | TC010 |
| FR-011 | Chat Bot Integration | UC-011 | 3.2.11 | 3.3.11 | TC011 |
| FR-012 | Medical History Tracking | UC-012 | 3.2.12 | 3.3.12 | TC012 |
| FR-013 | Doctor Browsing and Filtering | UC-013 | 3.2.13 | 3.3.13 | TC013 |
| FR-014 | Appointment Scheduling | UC-014 | 3.2.14 | 3.314 | TC014 |
| FR-015 | Doctor Profile Management | UC-015 | 3.2.15 | 3.315 | TC015 |
| FR-016 | Appointment Management | UC-016 | 3.2.16 | 3.316 | TC016 |
| FR-017 | Doctor Patient Interaction | UC-017 | 3.2.17 | 3.317 | TC017 |
| FR-018 | Review and Rating System | UC-018 | 3.2.18 | 3.318 | TC018 |
| FR-019 | Admin Panel | UC-019 | 3.2.19 | 3.319 | TC019 |

## Use Case Descriptions

### UC-001: User Sign Up

Table 22: UC-001

|  |  |
| --- | --- |
| Use Case ID: | UC-001 |
| Name: | User Sign Up |
| Actors: | New Users |
| Description: | The system shall provide a user sign-up mechanism to allow new users to register for an account. |
| Pre-Conditions: | None |
| Post-Condition: | New users are successfully registered and have an account created. |
| Main Flow: | Users navigate to the registration page.  Users enter their email address and password.  Users submit the registration form.  The system verifies the email address and password.  The system creates a new account for the user. |
| Alternate Flow: | If the email address is already associated with an existing account, the system prompts the user to use a different email address or recover their password. |
| Exception: | N/A |

### UC-002: User Login

Table 23: UC-002

|  |  |
| --- | --- |
| Use Case ID: | UC-002 |
| Name: | User Login |
| Actors: | Registered Users |
| Description: | The system shall provide a user login mechanism to allow registered users to access the system. |
| Pre-Conditions: | Users must be registered on the platform. |
| Post-Condition: | Registered users are logged in and directed to their dashboard. |
| Main Flow: | Users navigate to the login page.  Users enter their email address and password.  Users submit the login form.  The system verifies the email address and password.  The system logs in the user and redirects them to their dashboard. |
| Alternate Flow: | If the email address or password provided is incorrect, the system prompts the user to enter valid credentials or recover their password. |
| Exception: | N/A |

### UC-003: User Recognition

Table 24: UC-003

|  |  |
| --- | --- |
| Use Case ID: | UC-003 |
| Name: | User Recognition |
| Actors: | System, Users |
| Description: | The system determines a user's role, either by extracting specific fields from the signup form, such as user type or role, or by querying the database for the user's role based on the provided credentials. |
| Pre-Conditions: | User is logged in or signed up. |
| Post-Condition: | System identifies and assigns the appropriate role to the user. |
| Main Flow: | Upon login or signup, the system retrieves user credentials.  The system validates the credentials.  The system identifies the user's role based on the credentials. |
| Alternate Flow: | N/A |
| Exception: | N/A |

### UC-004: Initiate Prediction Process

Table 25: UC-004

|  |  |
| --- | --- |
| Use Case ID: | UC-004 |
| Name: | Initiate Prediction Process |
| Actors: | Users |
| Description: | After logging in, the system shall allow users to click on a 'Get Prediction' button to initiate the prediction process. |
| Pre-Conditions: | User is logged in. |
| Post-Condition: | Prediction process is initiated. |
| Main Flow: | User logs in to the system.  User clicks on the 'Get Prediction' button.  The system initiates the prediction process. |
| Alternate Flow: | N/A |
| Exception: | N/A |

### UC-005: Upload ECG Signal Data

Table 26: UC-005

|  |  |
| --- | --- |
| Use Case ID: | UC-005 |
| Name: | Upload ECG Signal Data |
| Actors: | Users |
| Description: | After clicking the 'Get Prediction' button, users should be prompted to upload their ECG signal data. |
| Pre-Conditions: | Prediction process is initiated. |
| Post-Condition: | ECG signal data is uploaded successfully. |
| Main Flow: | User clicks on the 'Get Prediction' button.  The system prompts the user to upload ECG signal data.  User uploads the ECG signal data. |
| Alternate Flow: | N/A |
| Exception: | N/A |

### UC-006: ECG Signal Visualization

Table 27: UC-006

|  |  |
| --- | --- |
| Use Case ID: | UC-006 |
| Name: | ECG Signal Visualization |
| Actors: | Users |
| Description: | The system shall provide functionality to visually represent ECG signals in the form of an ECG graph. |
| Pre-Conditions: | ECG signal data is available. |
| Post-Condition: | ECG signals are visualized on the system interface. |
| Main Flow: | User selects ECG data for visualization.  The system generates an ECG graph based on the selected data.  User views the ECG graph on the system interface. |
| Alternate Flow: | If the selected ECG data is invalid or incomplete, an error message is displayed, and the user is prompted to select valid data. |
| Exception: | N/A |

### UC-007: ECG Disease Prediction

Table 28: UC-007

|  |  |
| --- | --- |
| Use Case ID: | UC-007 |
| Name: | ECG Disease Prediction |
| Actors: | Users |
| Description: | The system shall predict heart disease by executing a machine learning model on the provided ECG data. |
| Pre-Conditions: | ECG signal data is uploaded. |
| Post-Condition: | Heart disease prediction results are displayed. |
| Main Flow: | User uploads ECG signal data.  The system analyses the data using a machine learning model.  Prediction results are presented to the user. |
| Alternate Flow: | N/A |
| Exception: | If the uploaded ECG data is corrupted or incomplete, the prediction process is halted, and an error message is displayed. |

### UC-008: ECG Image Upload

Table 29: UC-008

|  |  |
| --- | --- |
| Use Case ID: | UC-008 |
| Name: | ECG Image Upload |
| Actors: | Users |
| Description: | The system shall allow users to upload ECG images for the prediction of heart disease. |
| Pre-Conditions: | User is logged in. |
| Post-Condition: | ECG images are successfully uploaded. |
| Main Flow: | User navigates to the ECG image upload section.  User selects an ECG image for upload.  The system verifies and processes the uploaded image. |
| Alternate Flow: | If the uploaded image format is unsupported or the image is of poor quality, the system prompts the user to upload a valid image. |
| Exception: | N/A |

### UC-009: ECG Image Disease Prediction

Table 30: UC-009

|  |  |
| --- | --- |
| Use Case ID: | UC-009 |
| Name: | ECG Image Disease Prediction |
| Actors: | Users |
| Description: | The system shall analyse uploaded ECG images to predict heart disease using machine learning algorithms. |
| Pre-Conditions: | ECG images are uploaded. |
| Post-Condition: | Heart disease prediction results are displayed. |
| Main Flow: | User uploads an ECG image for analysis.  The system processes the image using machine learning algorithms.  Prediction results are presented to the user. |
| Alternate Flow: | N/A |
| Exception: | If the uploaded image does not contain sufficient information or is corrupted, the prediction process is terminated, and an error message is displayed. |

### UC-010: Prediction Display

Table 31: UC-010

|  |  |
| --- | --- |
| Use Case ID: | UC-010 |
| Name: | Prediction Display |
| Actors: | Users |
| Description: | The system shall present the predictions of heart disease to the user in a clear and understandable format. |
| Pre-Conditions: | Heart disease prediction results are available. |
| Post-Condition: | Prediction results are displayed to the user. |
| Main Flow: | User views the prediction results on the system interface. |
| Alternate Flow: | N/A |
| Exception: | If there are no prediction results available or the results cannot be displayed due to technical issues, an error message is displayed. |

### UC-011: Users, Chat Bot

Table 32: UC-011

|  |  |
| --- | --- |
| Use Case ID: | UC-012 |
| Name: | Chat Bot Integration |
| Actors: | Users, Chat Bot |
| Description: | The system shall integrate a chat bot feature capable of performing prediction analysis, addressing patient queries, providing prescriptions, and suggesting diet plans. |
| Pre-Conditions: | User is logged in. |
| Post-Condition: | Chat bot aids the user. |
| Main Flow: | User interacts with the chat bot through the messaging interface.  Chat bot analyses user queries and provides relevant responses. |
| Alternate Flow: | If the chat bot encounters a query it cannot address, it notifies the user and suggests contacting support. |
| Exception: | N/A |

### UC-012: Medical History Tracking

Table 33: UC-012

|  |  |
| --- | --- |
| Use Case ID: | UC-013 |
| Name: | Medical History Tracking |
| Actors: | Users |
| Description: | The system shall allow users to track their medical history. |
| Pre-Conditions: | User is logged in. |
| Post-Condition: | User medical history is recorded and accessible. |
| Main Flow: | User navigates to the medical history section.  User enters relevant medical information. |
| Alternate Flow: | N/A |
| Exception: | If there is an error in recording or retrieving medical history data, the system alerts the user and prompts them to try again later. |

### UC-013: Doctor Browsing and Filtering

Table 34: UC-013

|  |  |
| --- | --- |
| Use Case ID: | UC-014 |
| Name: | Doctor Browsing and Filtering |
| Actors: | Users |
| Description: | The system shall allow users to browse healthcare professionals and apply filters based on timing and rating to find the perfect doctor. |
| Pre-Conditions: | User is logged in. |
| Post-Condition: | User finds a suitable healthcare professional. |
| Main Flow: | User accesses the directory of healthcare professionals.  User applies filters based on timing and rating.  User views detailed profiles of filtered professionals. |
| Alternate Flow: | N/A |
| Exception: | If there are no healthcare professionals matching the user's filters or the directory is unavailable, the system notifies the user and suggests alternative actions. |

### UC-014: Appointment Scheduling

Table 35: UC-014

|  |  |
| --- | --- |
| Use Case ID: | UC-015 |
| Name: | Appointment Scheduling |
| Actors: | Users |
| Description: | The system shall provide functionality for users to schedule appointments with healthcare professionals. |
| Pre-Conditions: | User is logged in. |
| Post-Condition: | Appointment is scheduled and confirmed. |
| Main Flow: | User selects a healthcare professional.  User views the availability of the professional.  User selects preferred appointment slot.  User confirms the appointment. |
| Alternate Flow: | If there are scheduling conflicts or the selected professional is unavailable, the system prompts the user to choose an alternative time or professional. |
| Exception: | N/A |

### UC-015: Doctor Profile Management

Table 36: UC-015

|  |  |
| --- | --- |
| Use Case ID: | UC-016 |
| Name: | Doctor Profile Management |
| Actors: | Doctors |
| Description: | Doctors can update personal info, qualifications, specialties, clinic affiliations, and availability. They can also respond to patient reviews. |
| Pre-Conditions: | Doctor is logged in. |
| Post-Condition: | Doctor profile is updated successfully. |
| Main Flow: | Doctor accesses the profile management section.  Doctor updates personal information, qualifications, etc.  Doctor saves changes to the profile. |
| Alternate Flow: | N/A |
| Exception: | If there is an error in updating the profile or saving changes, the system alerts the doctor and prompts them to try again later. |

### UC-016: Appointment Management

Table 37: UC-016

|  |  |
| --- | --- |
| Use Case ID: | UC-017 |
| Name: | Appointment Management |
| Actors: | Doctors |
| Description: | Doctors set availability, patients book appointments accordingly. System handles conflicts and sends confirmation notifications. |
| Pre-Conditions: | Doctor is logged in. |
| Post-Condition: | Appointment schedule is updated. |
| Main Flow: | Doctor accesses the appointment management section.  Doctor sets availability or manages existing appointments.  System updates the appointment schedule. |
| Alternate Flow: | N/A |
| Exception: | If there are conflicts in the appointment schedule or errors in updating, the system alerts the doctor and prompts them to resolve the issues. |

### UC-017: Doctor Patient Interaction

Table 38: UC-017

|  |  |
| --- | --- |
| Use Case ID: | UC-018 |
| Name: | Doctor Patient Interaction |
| Actors: | Doctors, Patients |
| Description: | Patients message doctors for inquiries, follow-ups, or updates. Doctors respond promptly. |
| Pre-Conditions: | Doctor and patient are logged in. |
| Post-Condition: | Communication between doctor and patient is established. |
| Main Flow: | Patient sends a message to the doctor through the messaging interface.  Doctor receives the message and responds accordingly. |
| Alternate Flow: | N/A |
| Exception: | If there are delays in message delivery or technical issues in communication, the system notifies both the doctor and patient and advises them to try again later. |

### UC-018: Review and Rating System

Table 39: UC-018

|  |  |
| --- | --- |
| Use Case ID: | UC-019 |
| Name: | Review and Rating System |
| Actors: | Patients |
| Description: | The system shall incorporate a review and rating system to gather feedback from patients regarding their experiences with healthcare professionals. |
| Pre-Conditions: | Patient has completed an appointment. |
| Post-Condition: | Feedback and ratings are submitted successfully. |
| Main Flow: | Patient completes an appointment with a healthcare professional.  Patient provides feedback and ratings for the professional. |
| Alternate Flow: | N/A |
| Exception: | If there is an error in submitting feedback or ratings, the system alerts the patient and prompts them to try again later. |

### UC-019: Admin Panel

Table 40: UC-019

|  |  |
| --- | --- |
| Use Case ID: | UC-020 |
| Name: | Admin Panel |
| Actors: | Administrators |
| Description: | The system will provide comprehensive Doctor Monitoring and Control feature within the admin panel, enabling administrators to supervise and manage doctor activities effectively. |
| Pre-Conditions: | Administrator is logged in. |
| Post-Condition: | Administrator monitors and manages doctor activities successfully. |
| Main Flow: | Administrator accesses the admin panel.  Administrator views activity logs and performance metrics for doctors.  Administrator takes necessary actions based on the information. |
| Alternate Flow: | N/A |
| Exception: | If there are errors in retrieving activity logs or metrics, the system alerts the administrator and advises them to try again later. |

## Use Case Design

### User Sign Up

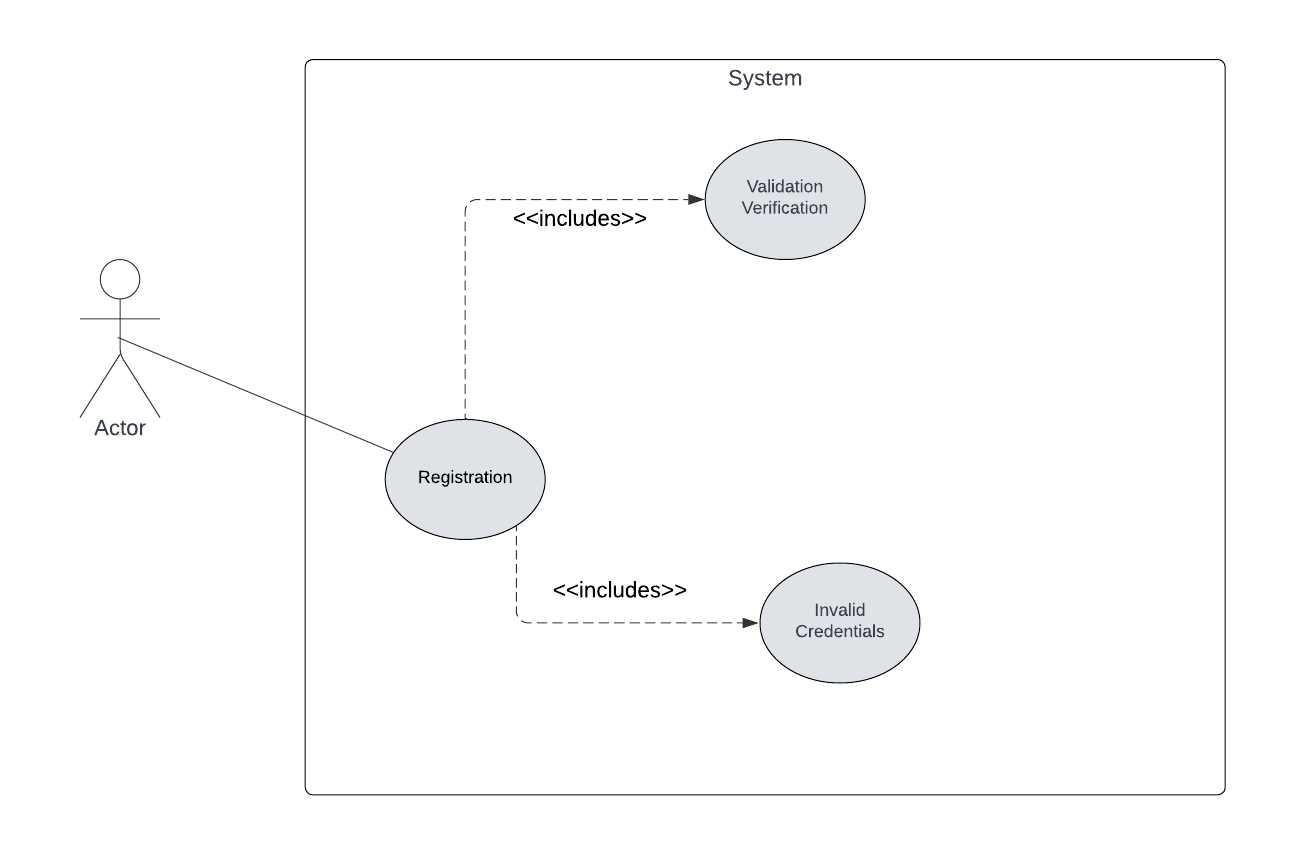


Figure 1: UC-001

### User Login

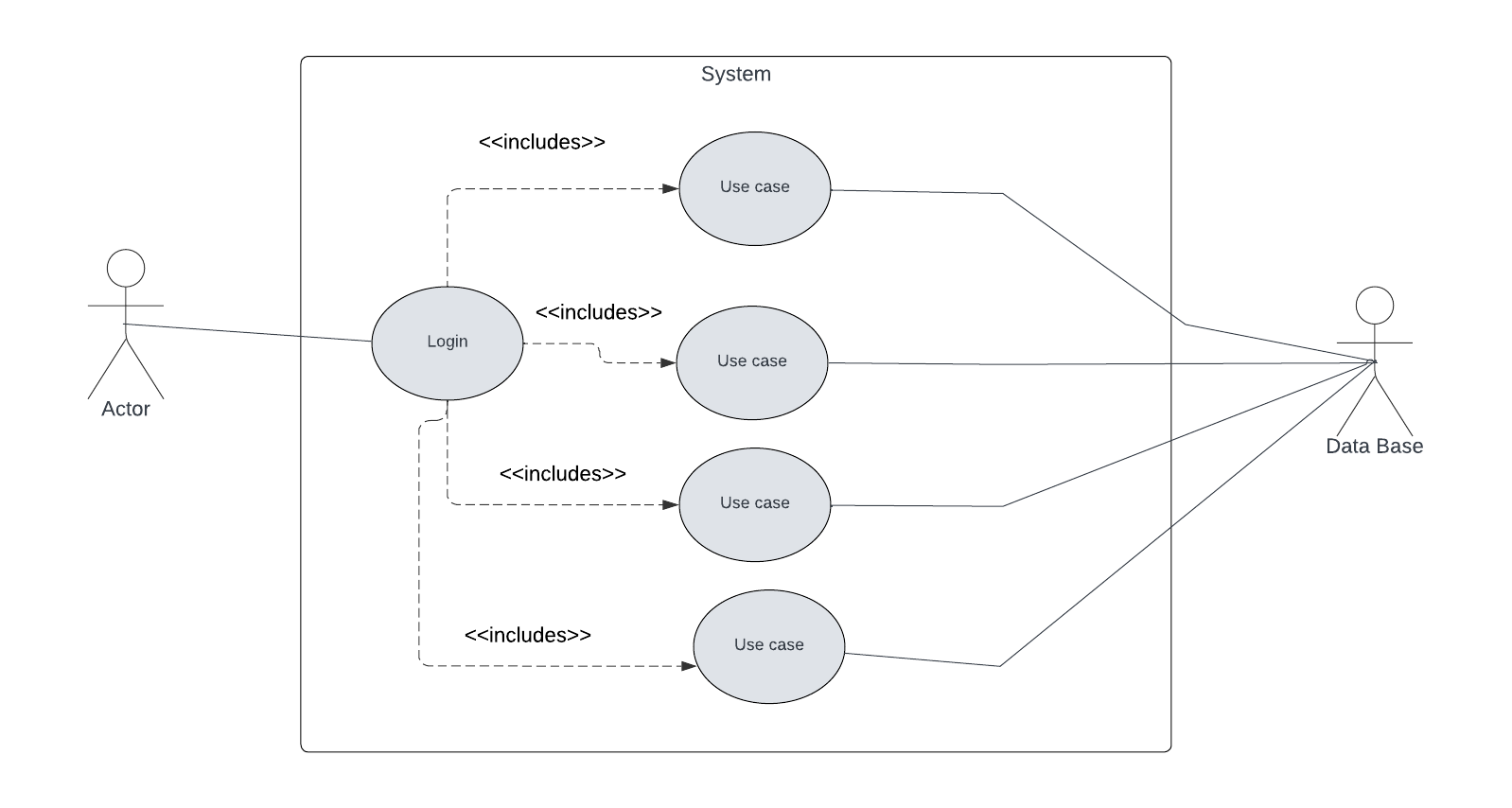


Figure 2: UC-002

### User Recognition

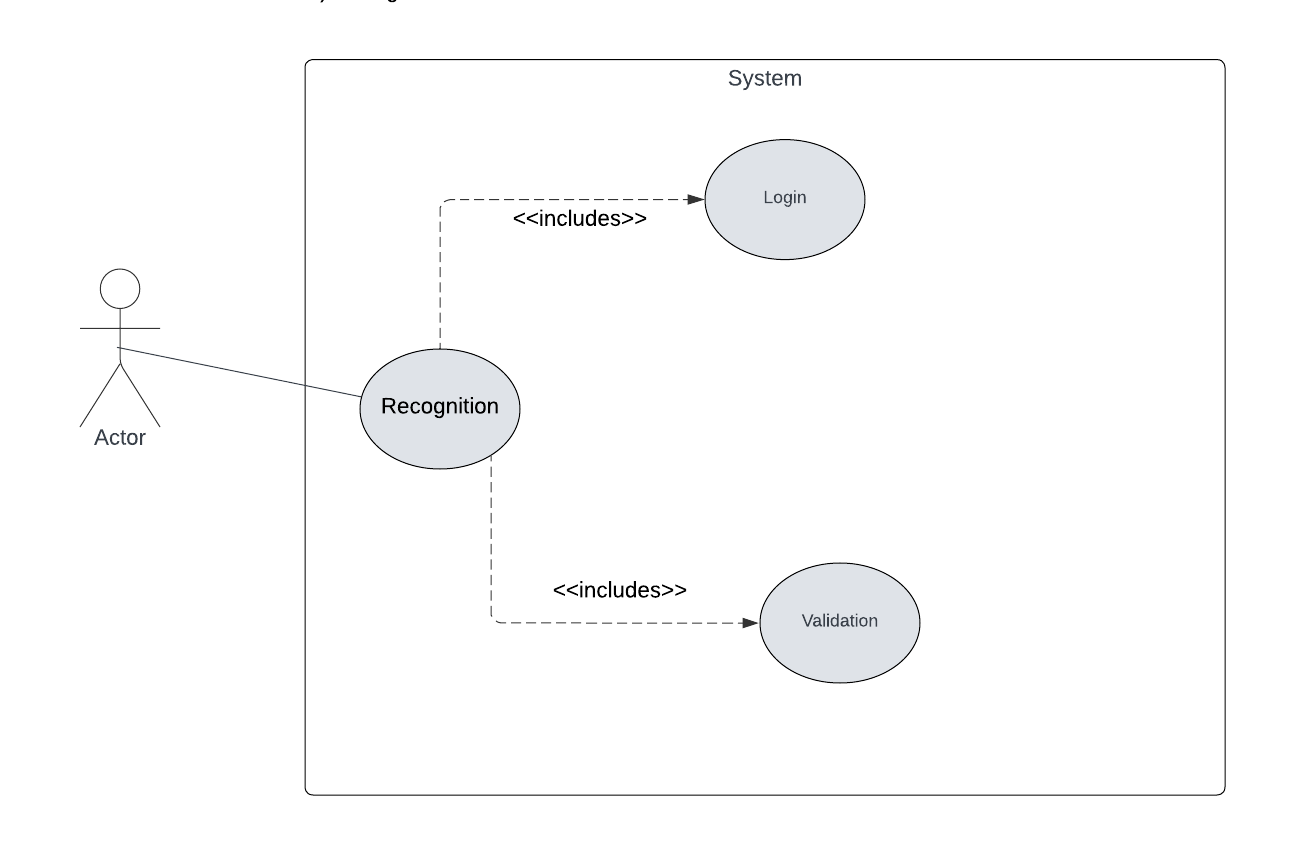


Figure 3: UC-003

### Initiate Prediction Process

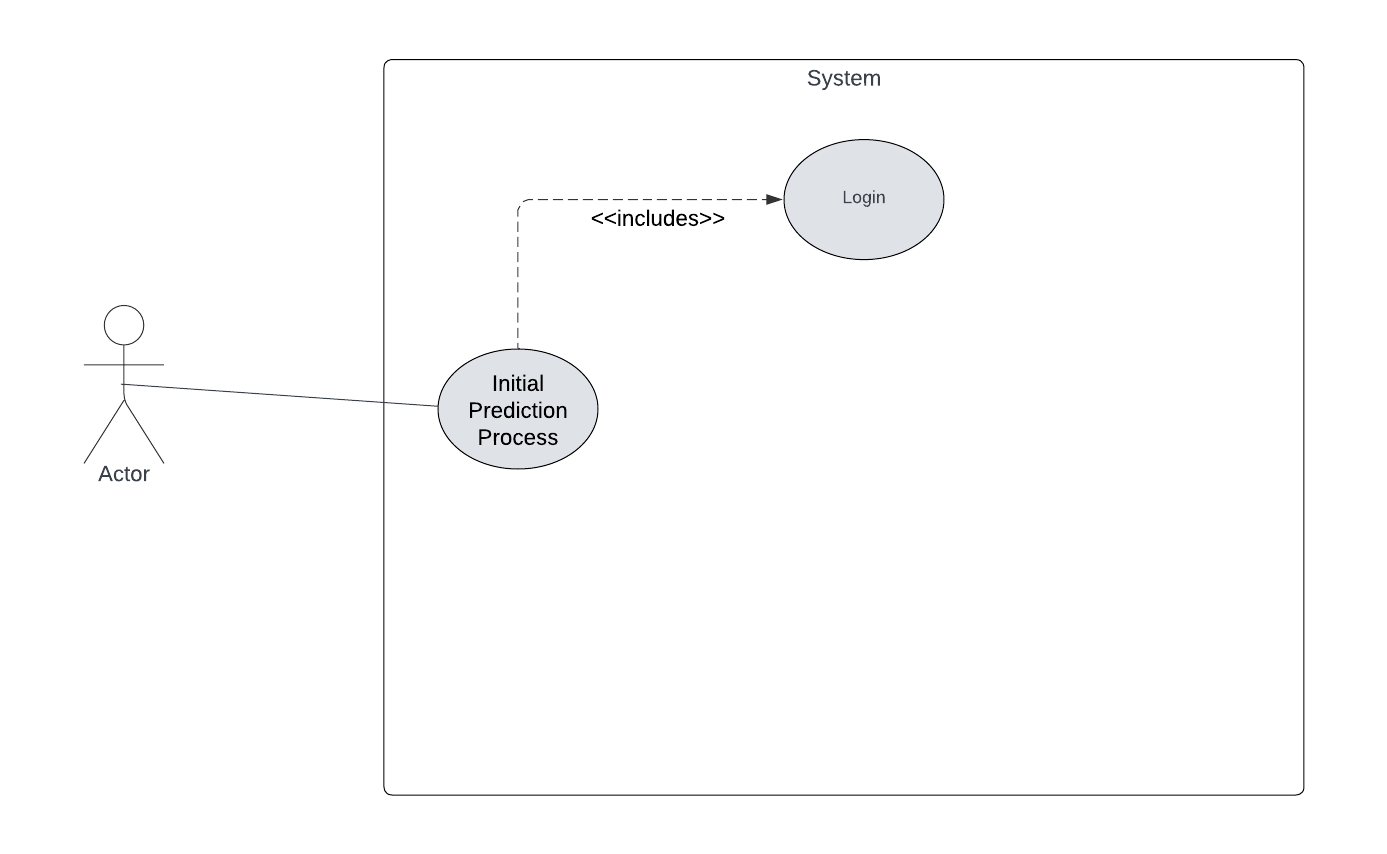


Figure 4: UC-004

### Upload ECG Signal Data

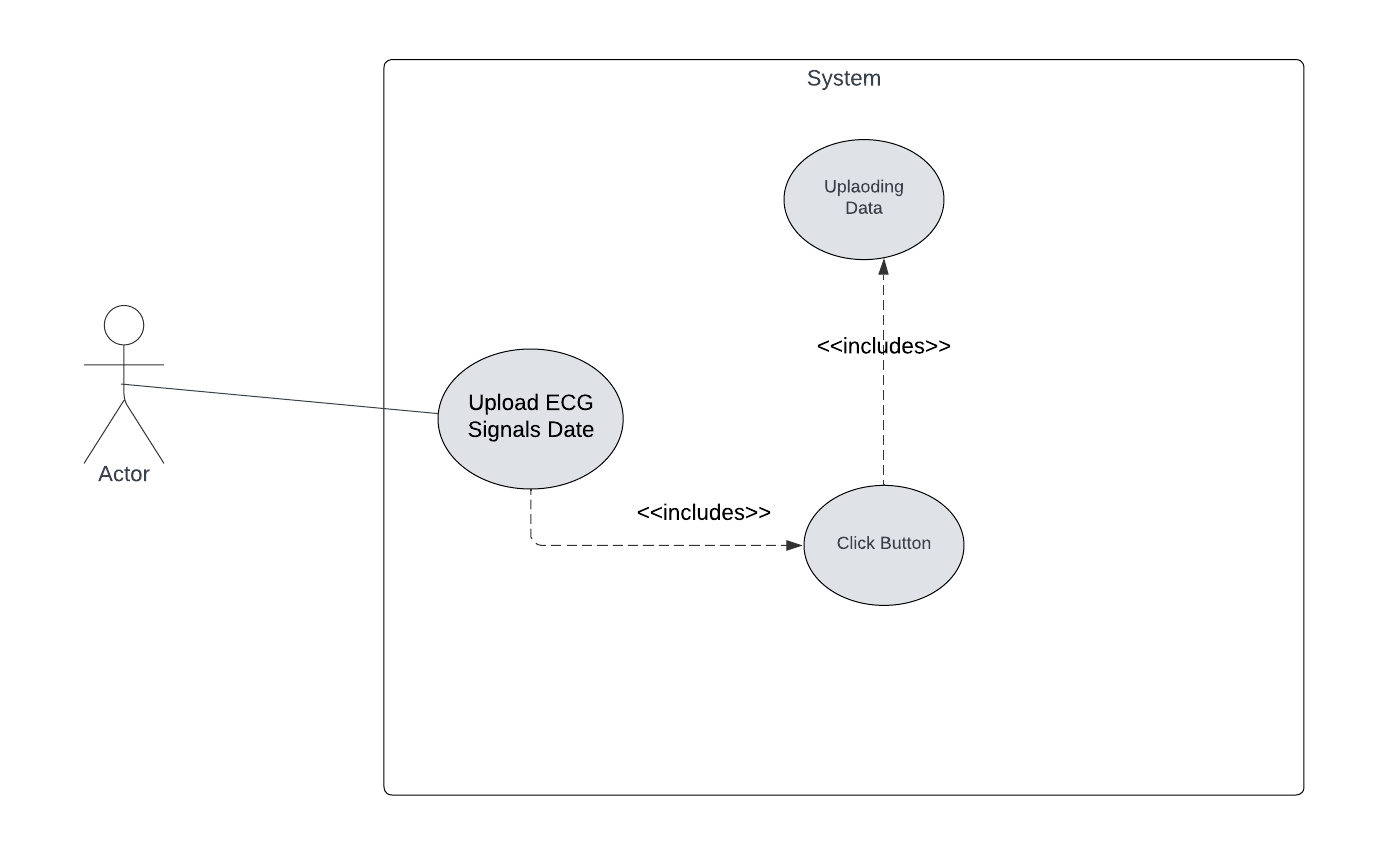


Figure 5: UC-005

### ECG Signal Visualization

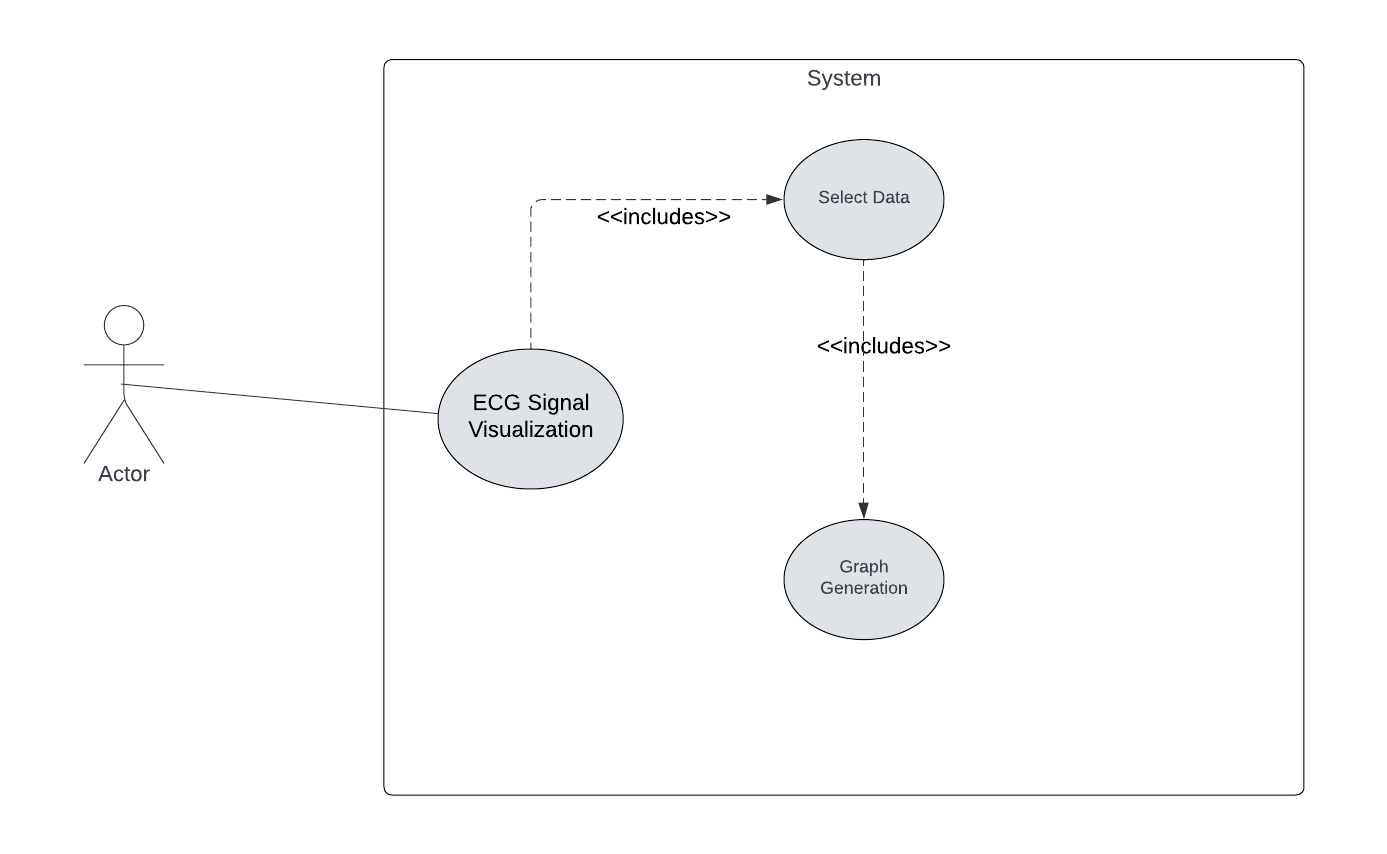


Figure 6: UC-006

### ECG Disease Prediction

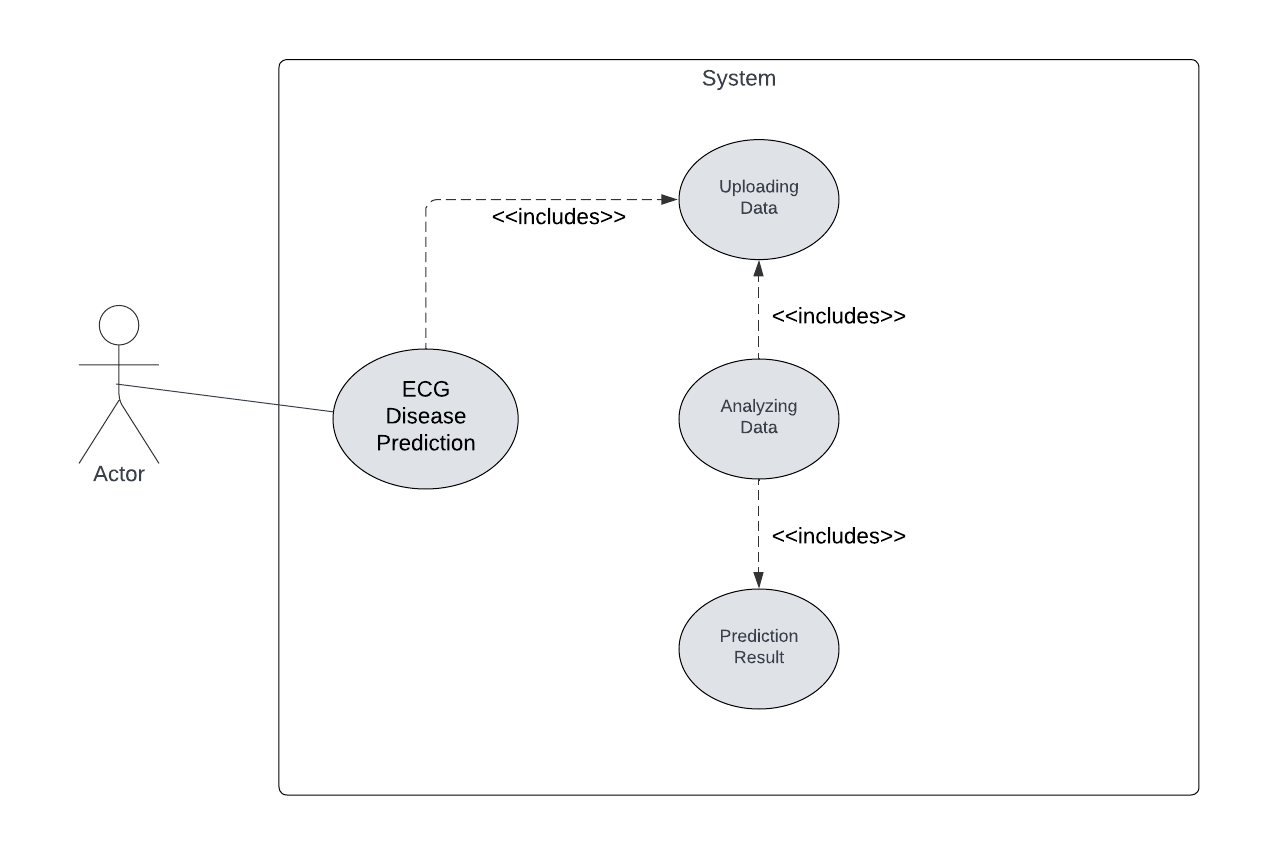


Figure 7: UC-007

### ECG Image Upload

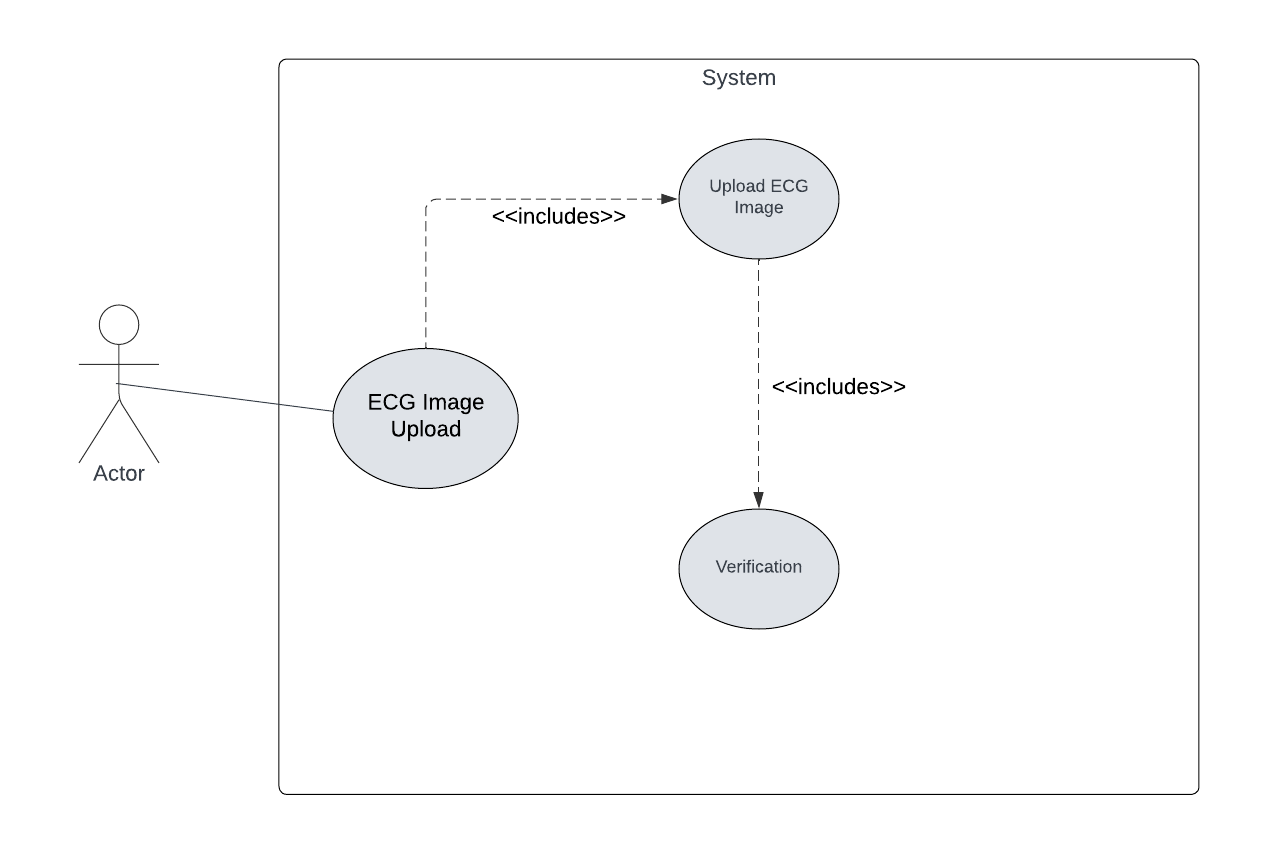


Figure 8: UC-008

### ECG Image Disease Prediction

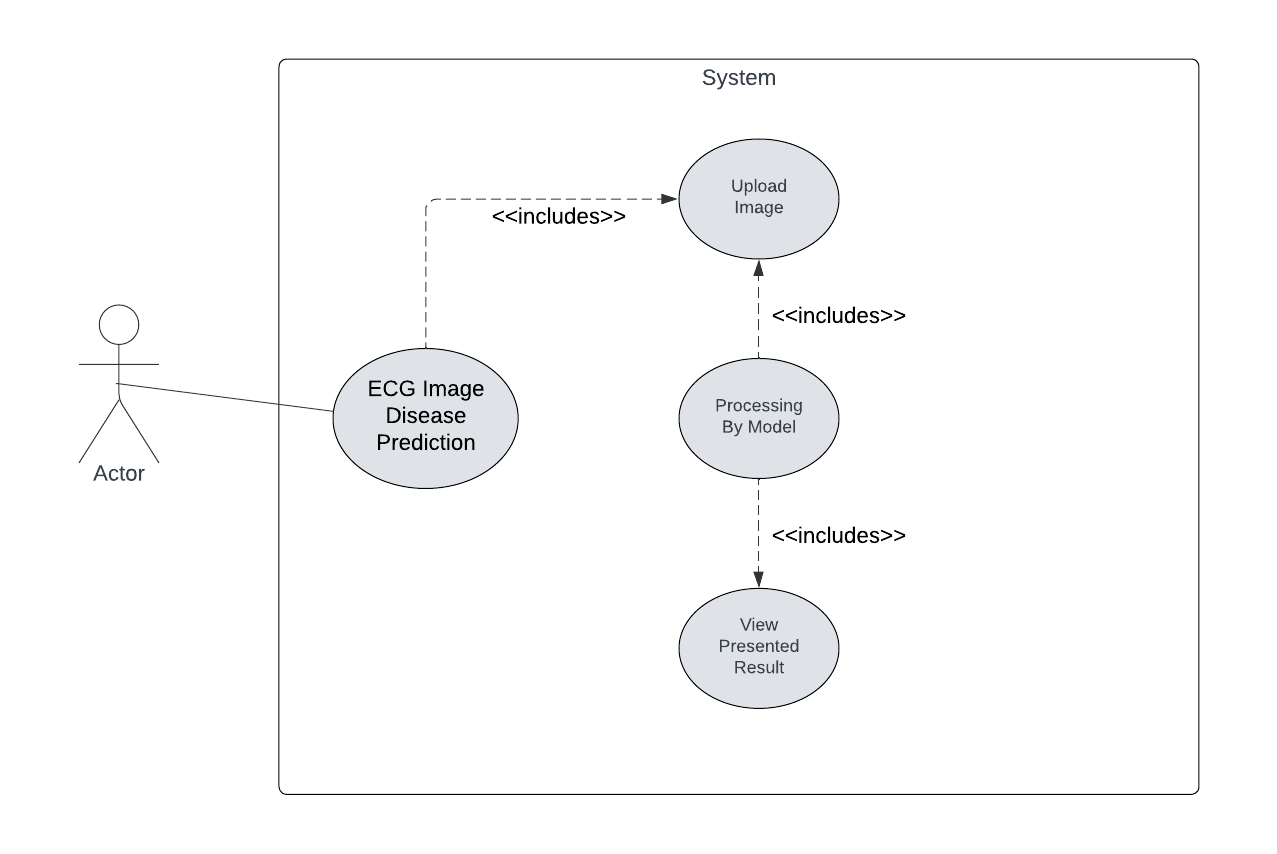


Figure 9: UC-009

### Prediction Display

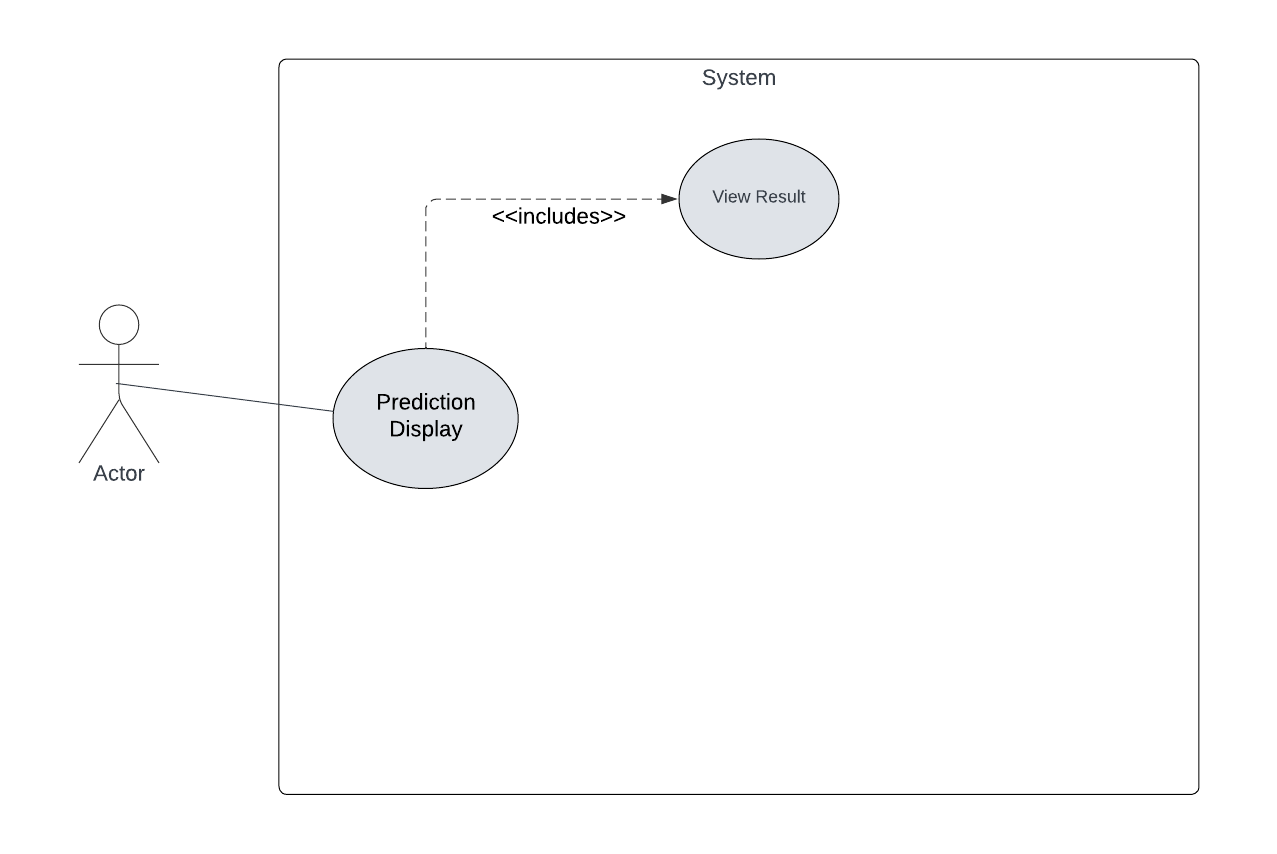


Figure 10: UC-010

### Chat Bot Integration

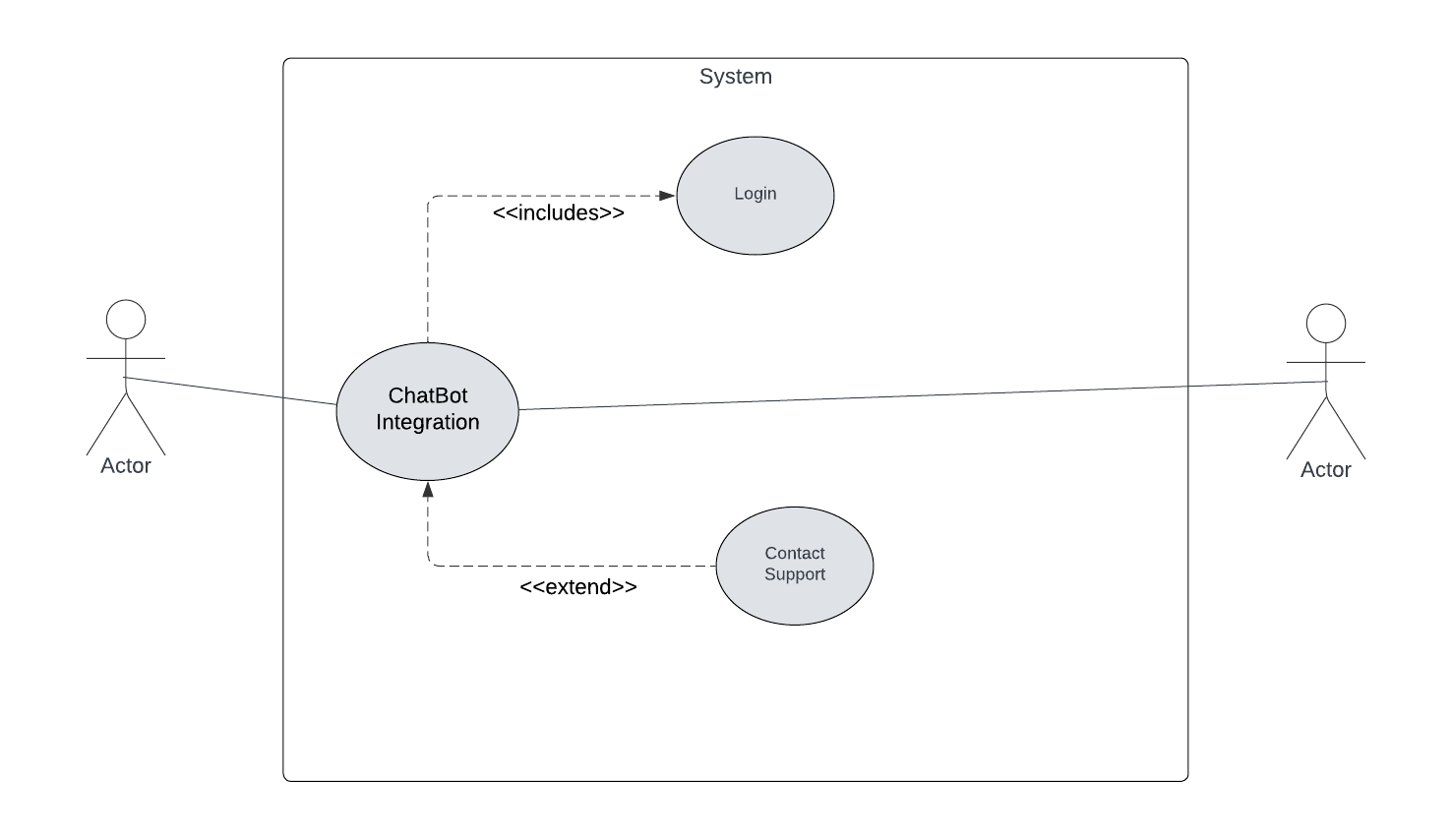


Figure 11: UC-011

### Medical History Tracking

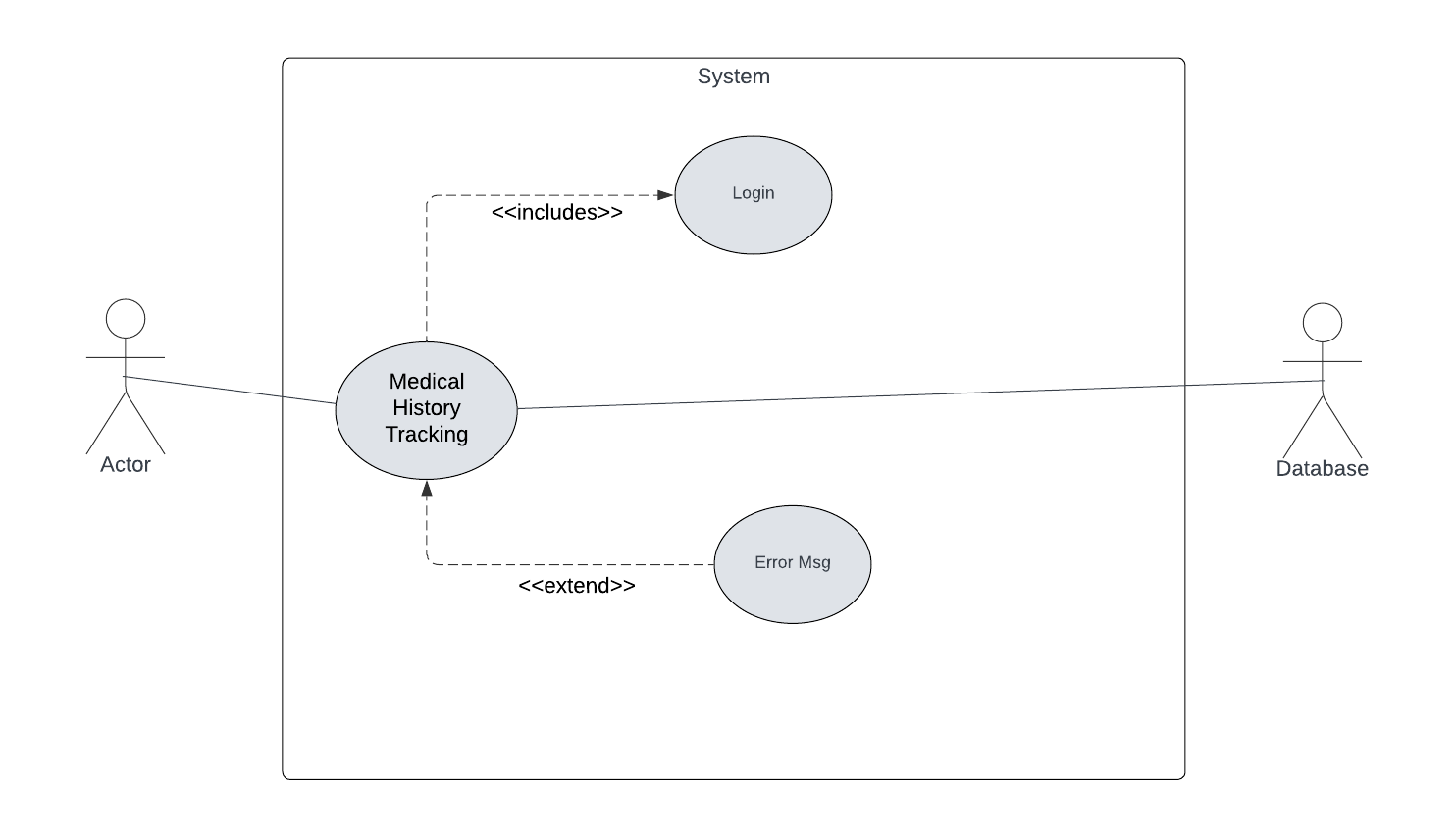


Figure 12: UC-012

### Doctor Browsing and Filtering

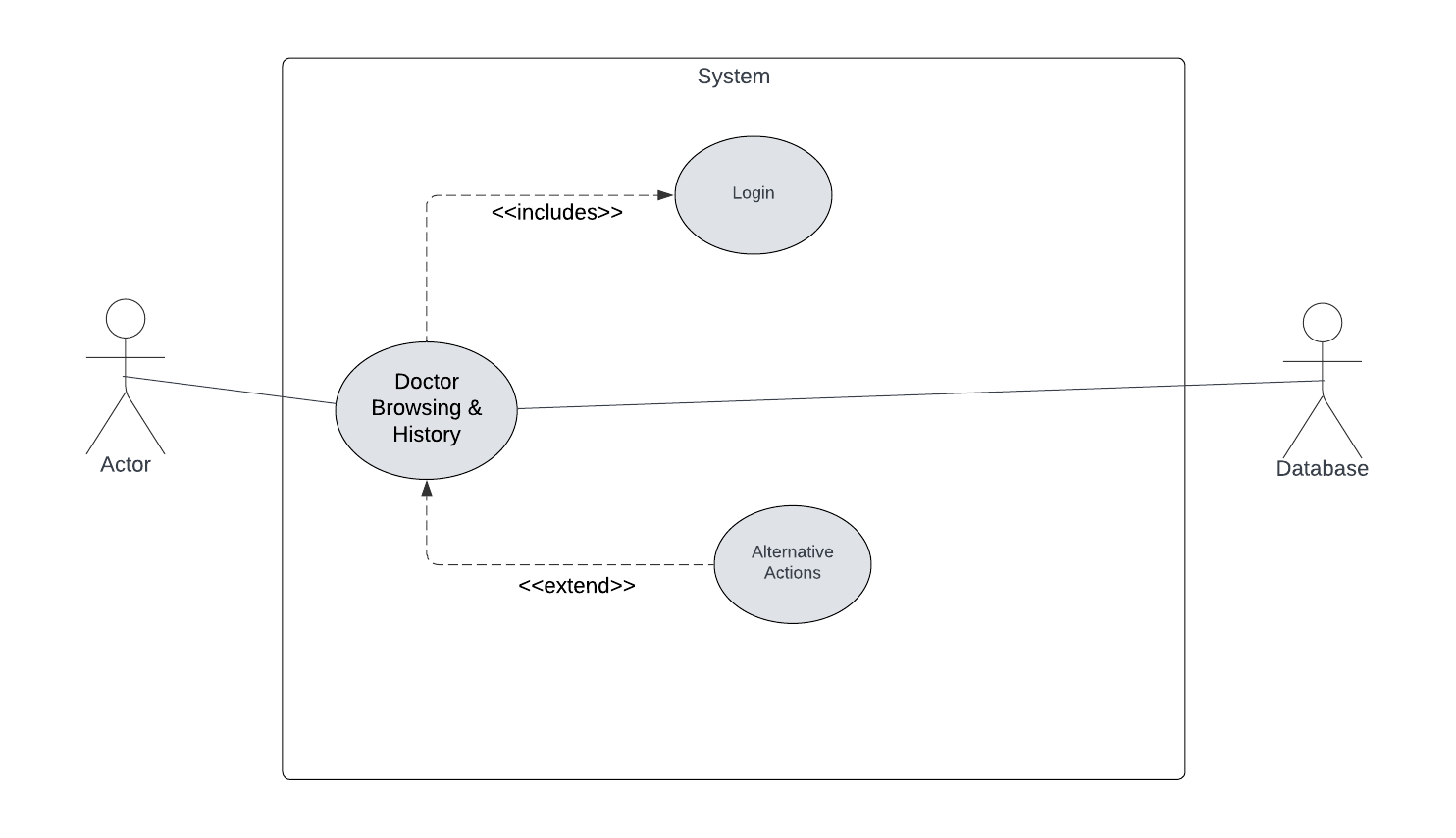


Figure 13: UC-013

### Appointment Scheduling

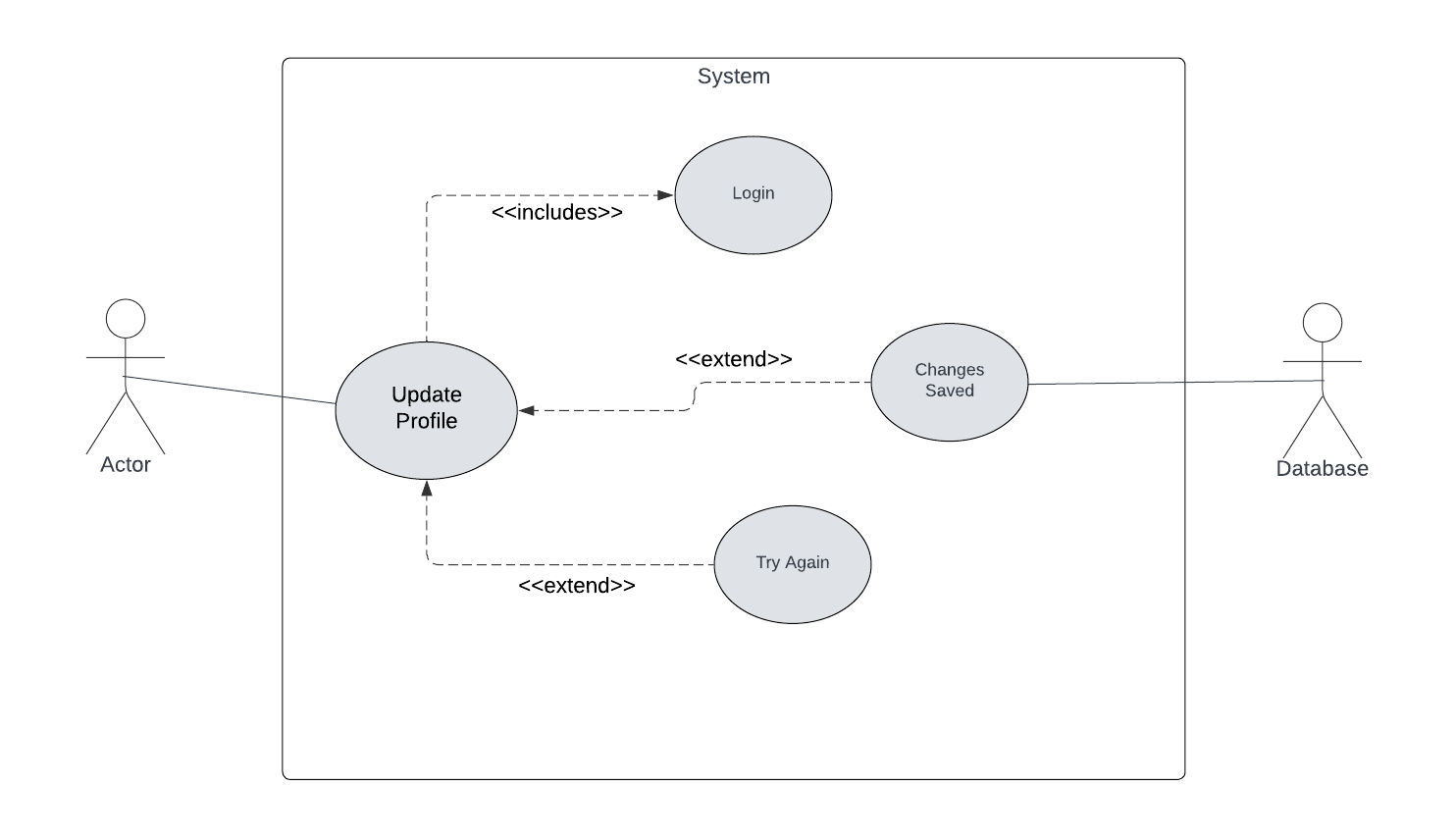


Figure 14: UC-014

### Doctor Profile Management

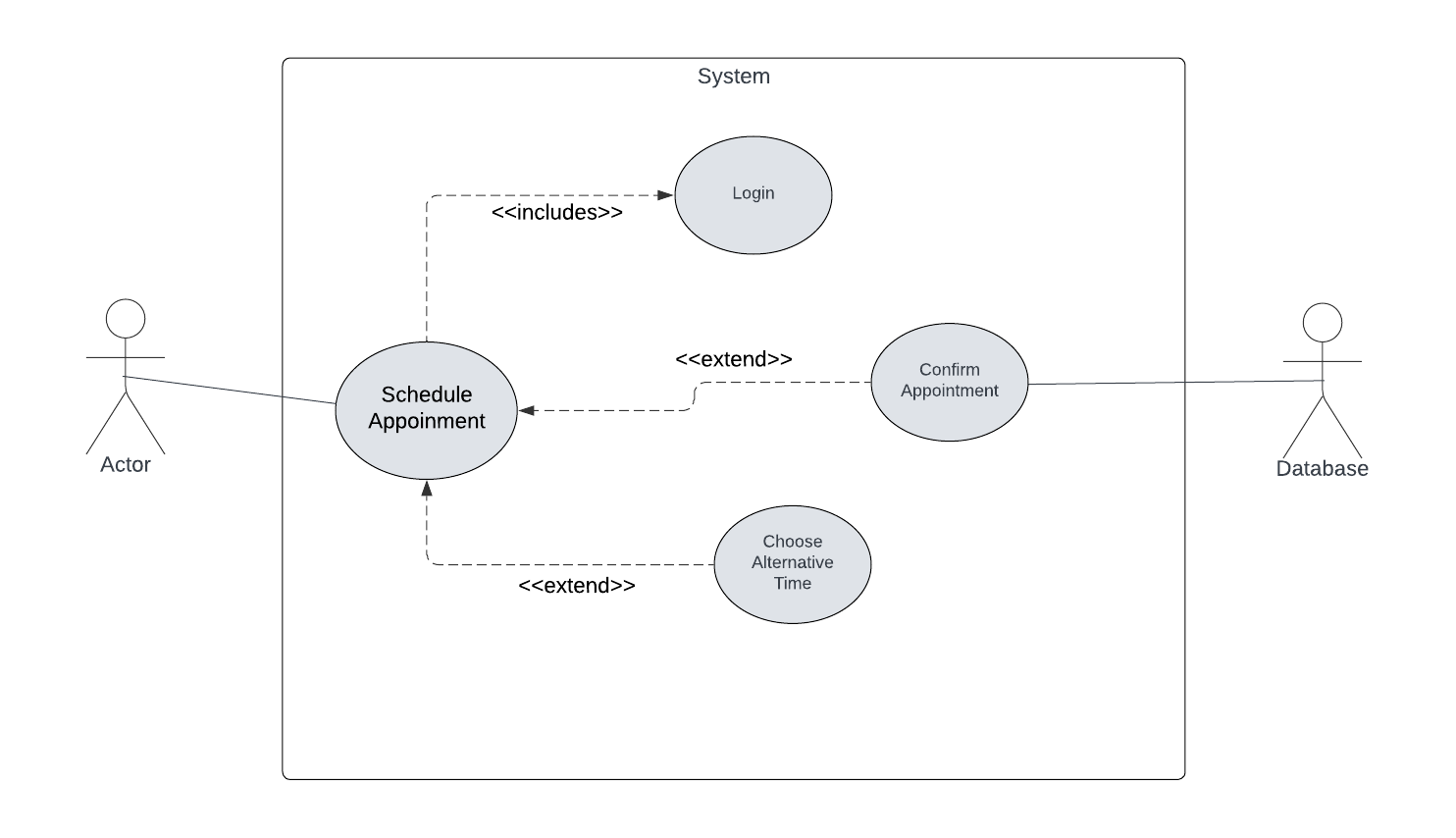


Figure 15: UC-015

### Appointment Management

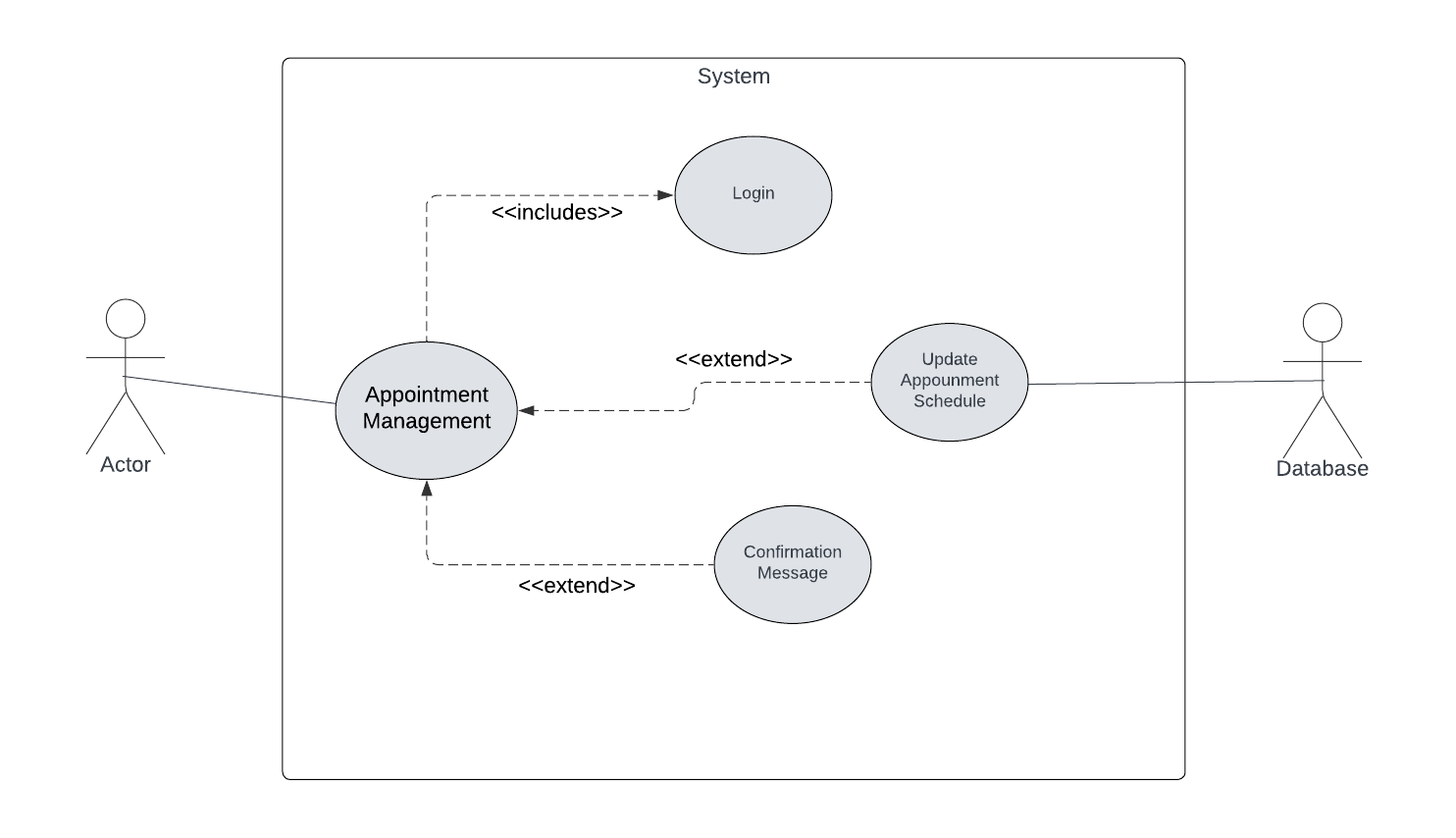


Figure 16: UC-016

### Doctor Patient Interaction

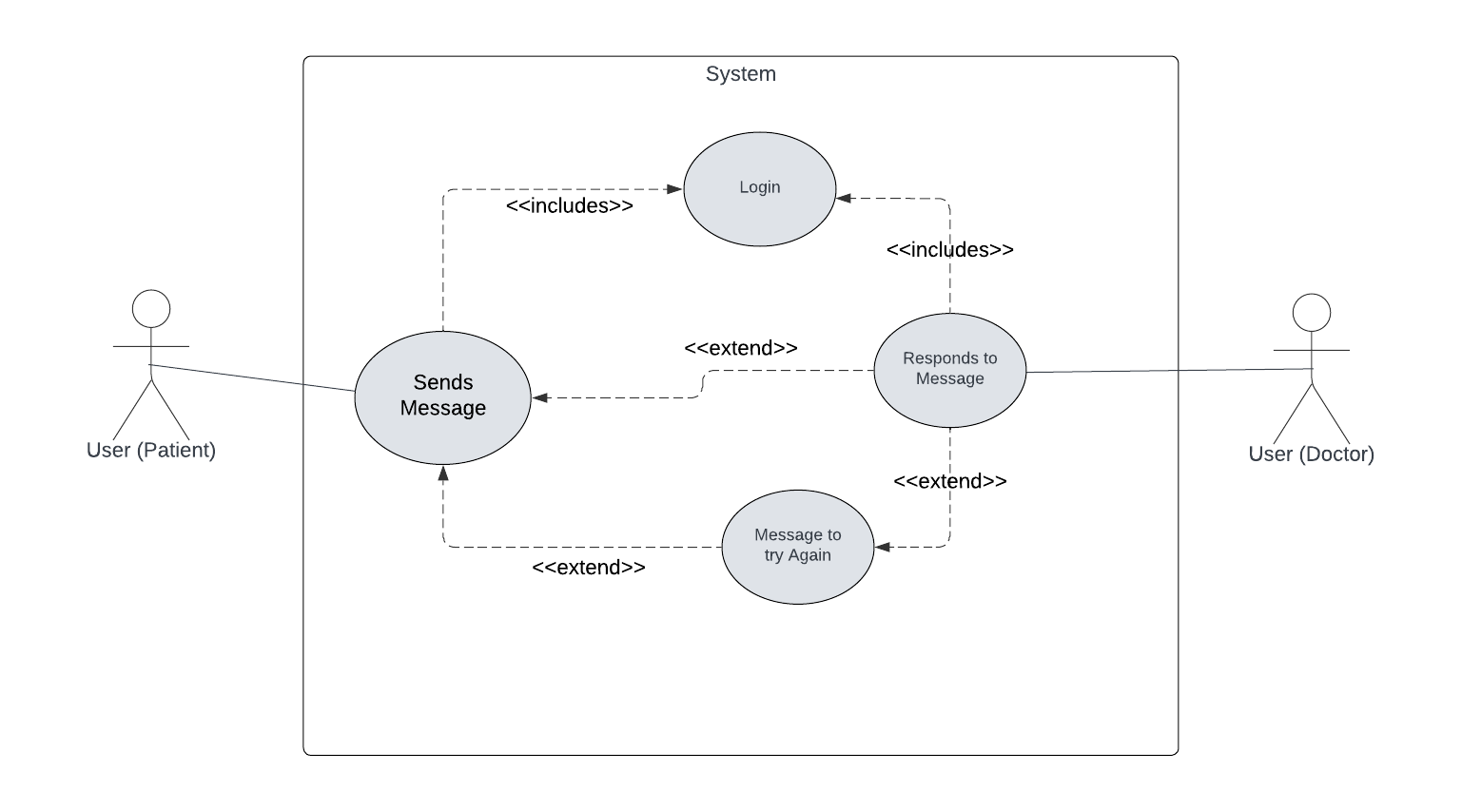


Figure 17: UC-017

### Review and Rating System

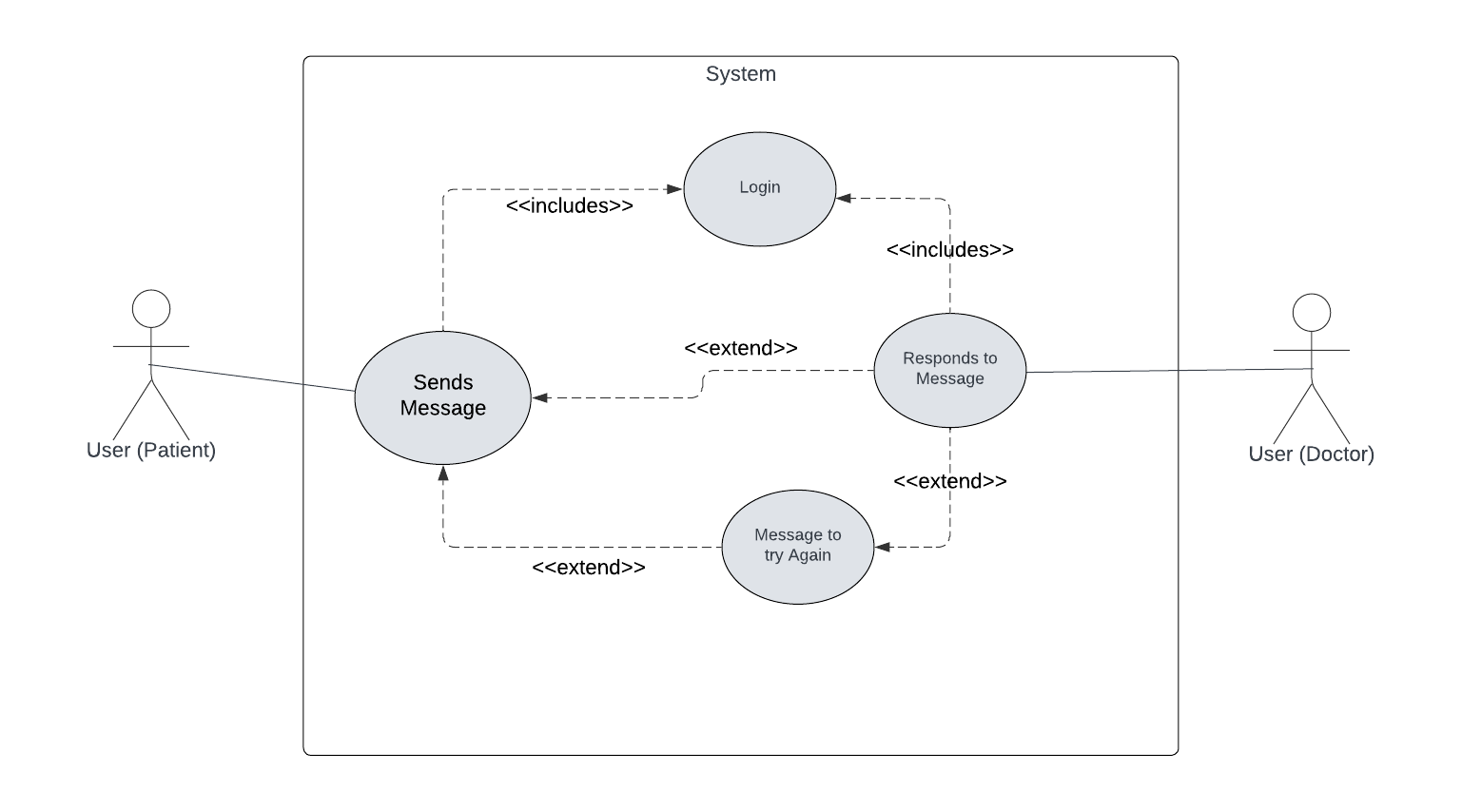


Figure 18: UC-018

### Admin Panel

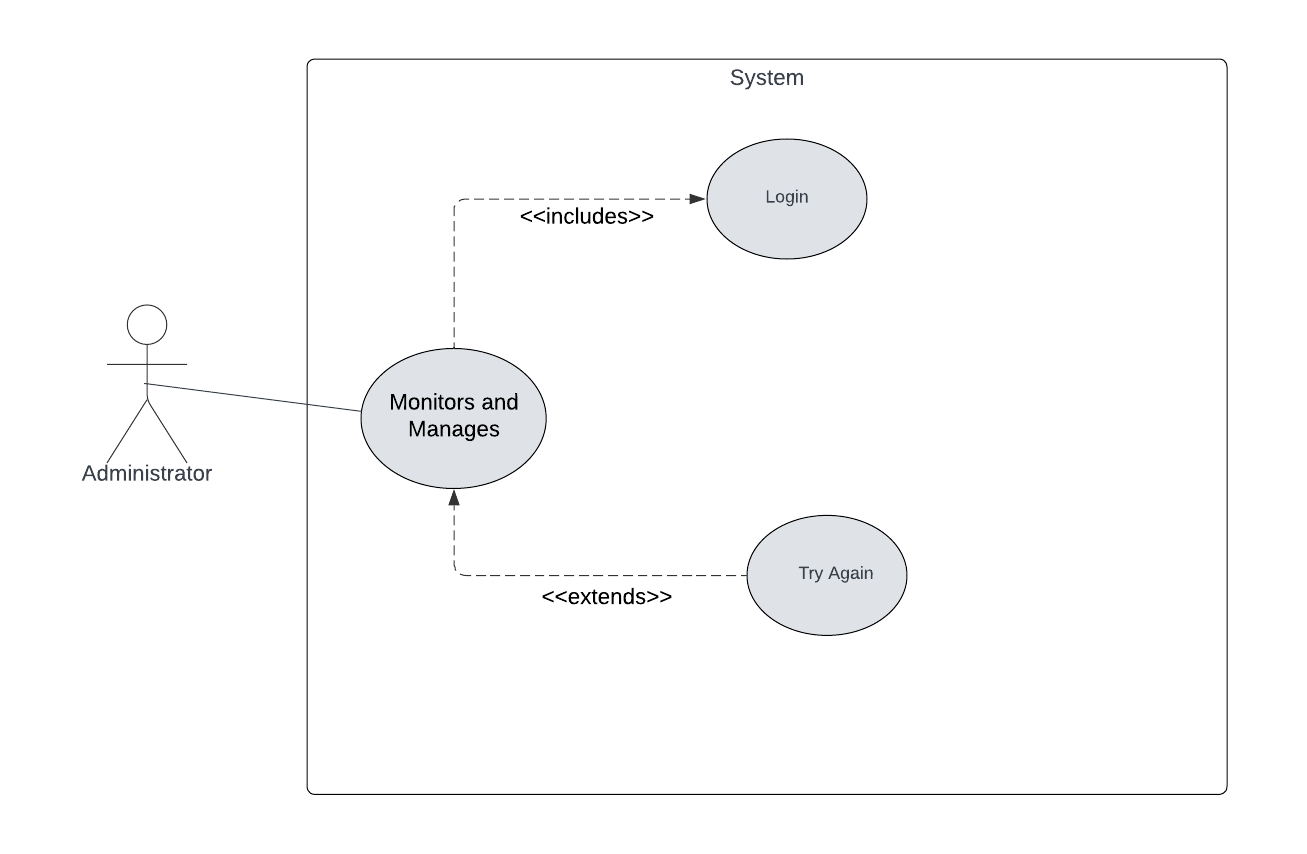


Figure 19: UC-019

## Software Development Life Cycle Model

### Waterfall Model:

The Software development Life cycle we will be utilizing will be the simple and old-fashioned Waterfall Model. The waterfall model works on preceding phase that acts as input for every next phase because of the sequential design process where progress will be seen as steadily flowing through each phase of requirements, analysis, design, coding, testing and operations.

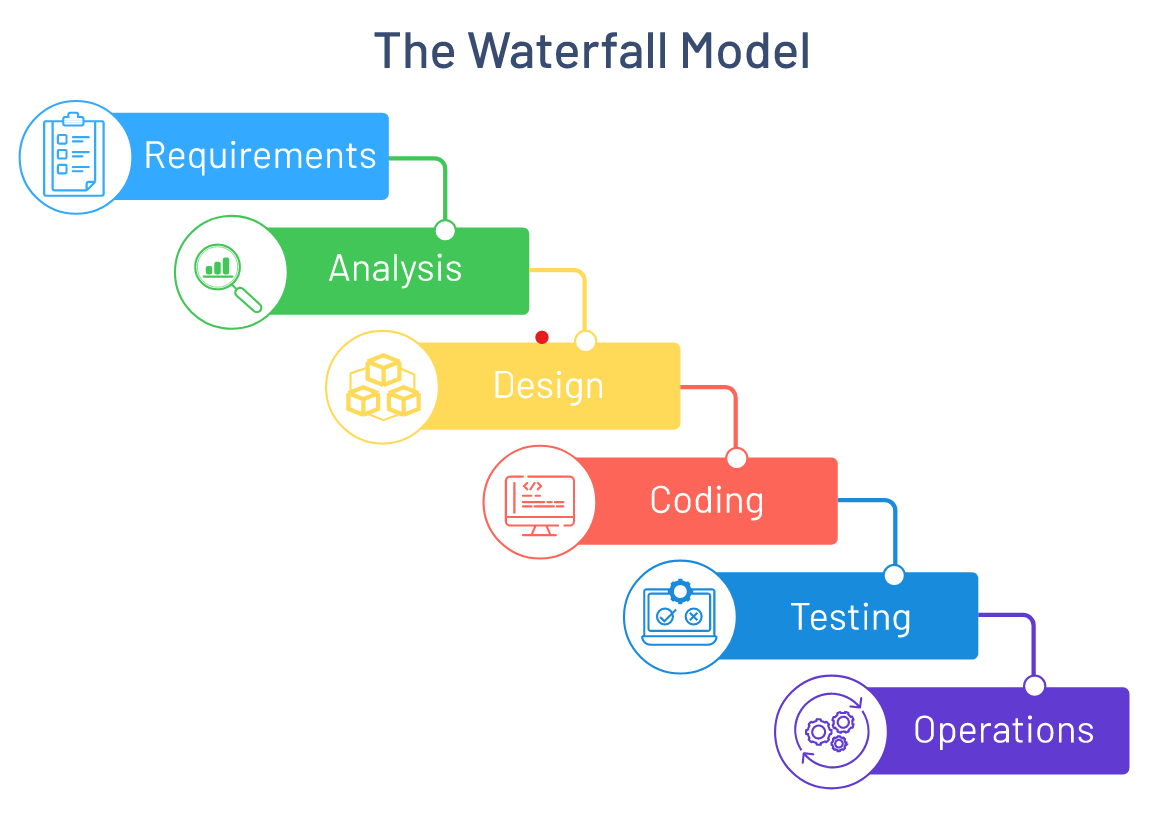


Figure 21: The Waterfall Model

#### Requirements:

The requirements phase in context to our project would be the functional requirements and non-functional requirements designing and discussion with our respective supervisor. As our analysis on the requirements is dependent on completion of this phase, we won’t move on until the scope of functionality of our project is finalized.

#### Analysis:

In the analysis phase, we will reflect on our functional as well and non-functional requirements and perform formal analysis by observing competency required for using the software tools including the frameworks and technologies we will utilize to develop our frontend client for user, set up servers to handle requests as well the API we will create to deploy our model.

#### Design:

After the completion of analysis phase, we will begin designing our project. This designing will cover each aspect of our project. From designing the schema for our backend database to designing high fidelity prototypes for user interface, all of the designing and wireframing will be done in this phase so the development can be started.

#### Coding:

Once the preceding activity of designing all components of the project is finished, we can start working on the development. In this phase, we will begin developing our frontend which will be convenient as we will have prototypes prepared, followed by setting up our server for interaction with database (which will also follow the schema we would have designed in design phase), and finally we start with the development of our model, which will go through every step of developing workflow of a machine learning model. After our model is trained and tested, we will move on to the deployment part of it, which we will execute by converting it to an API.

#### Testing:

Once our project is fully development, we will begin testing it using different types of testing to fully make it bug free. We will begin by unit testing, where we will test individual methods and functions, components, or modules we will have in our project. Followed by unit testing, we will have integration testing that will substantiate that different auxiliary modules or services being used by our projects are performing well together. Finally, we will have acceptance testing that will verify if our project satisfies our business perspective requirements (our monetization plans) and system testing in which we will conduct complete integrated testing of the entire system, along with its specified requirements. So, that our final project will be a debugged version of the original, presumably free of any possible defects.

#### Operations:

After the complete development and testing of our project, it will be ready for deployment and maintenance. We will begin by utilizing a deploying service (like Azure, AWS) and set up all components of our project put up globally. With all our servers and frontend running live, we will transition into maintenance to work on the future bugs and errors that may occur.

### Justification for Using this Model

Our project follows a sequential following of steps that are dependent of their preceding phases without needing to recur on the previous step ever again. In context of our project, we will begin by defining our requirements, which once finalized won’t require us to come back again to change them again. Similarly, the analysis phase will also occur only once, and all the analysis of the requirements will be done in this phase. The same pattern will follow in other phases of designing, coding, testing, and operations/maintenance. In every phase, that relevant activity will be completed and won’t require us to ever involve in it again.

Thus, there won’t ever be any need for any kind of iterative go through over all the activities again as these activities would only happen once and for all. Due to these reasons, any kind of incremental or iterative process model like agile or hybrid won’t be required and just would be an overkill for this kind of project. The simple waterfall will do the trick for us. The waterfall model is a type of software development process that is linear and sequential. In this model, each stage of the development process is completed before moving on to the next stage. This means that there is no overlap or iteration between the different stages (just like our project). Because of this, the waterfall model is best suited for projects where the requirements are well understood and relatively unchanging.

Other process models including iterative and incremental based approaches like agile won’t be necessary here as they would be an overkill. These techniques shine in cases where the requirements are volatile and changing rapidly. Some cases also include in which the client is asking for improvement for customer satisfaction. None of these scenarios apply to us. Our Final Year Project will be developed in a linear flow, and once requirements are finalized, they won’t be changed again. Thus, waterfall model would be more sufficient than agile or any other process model.

Chapter 03:  System Design

# System Design

## Work Breakdown Structure (WBS)

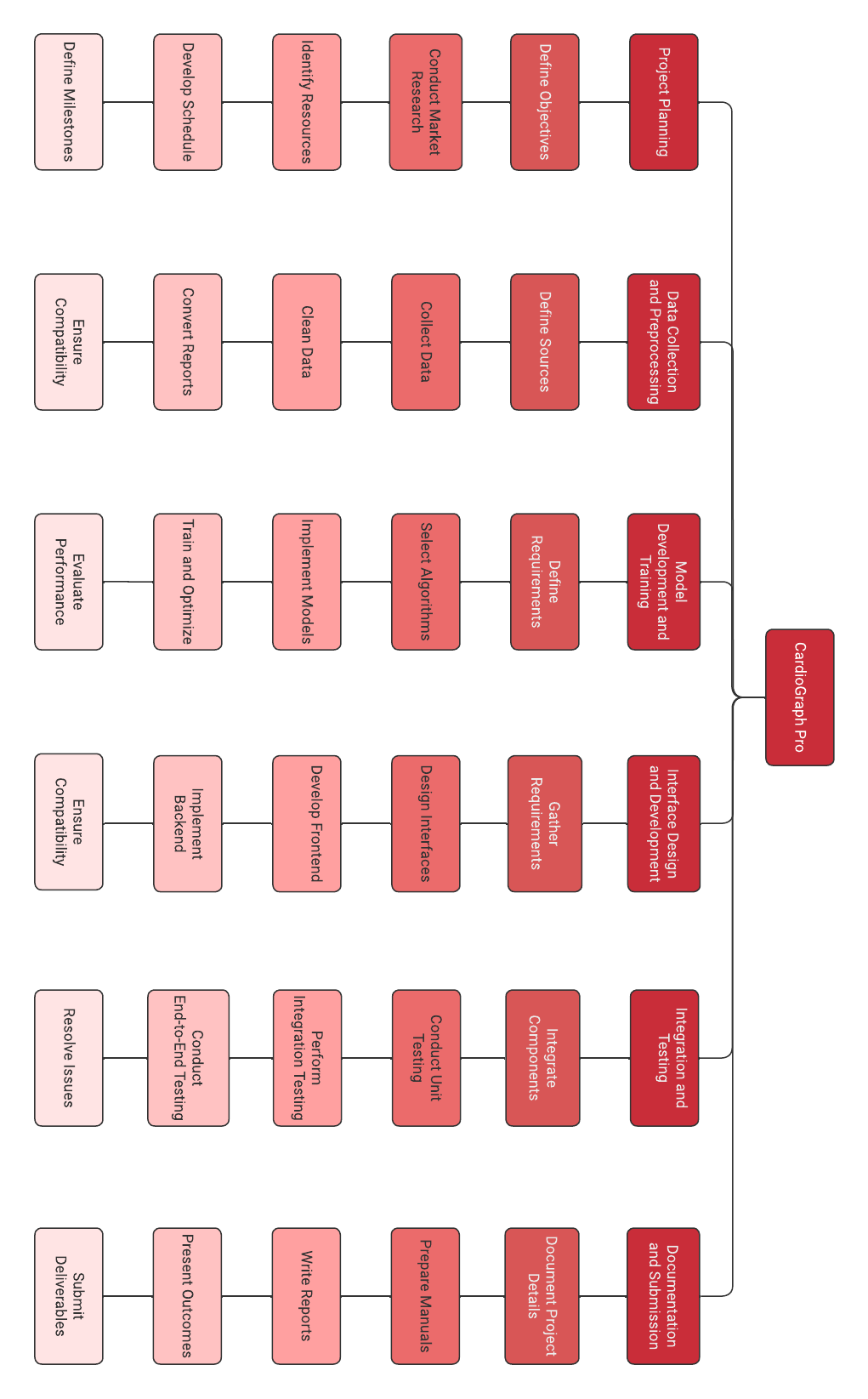


Figure 22: Work Breakdown Structure (WBS)

## Activity Diagram

### User Sign Up

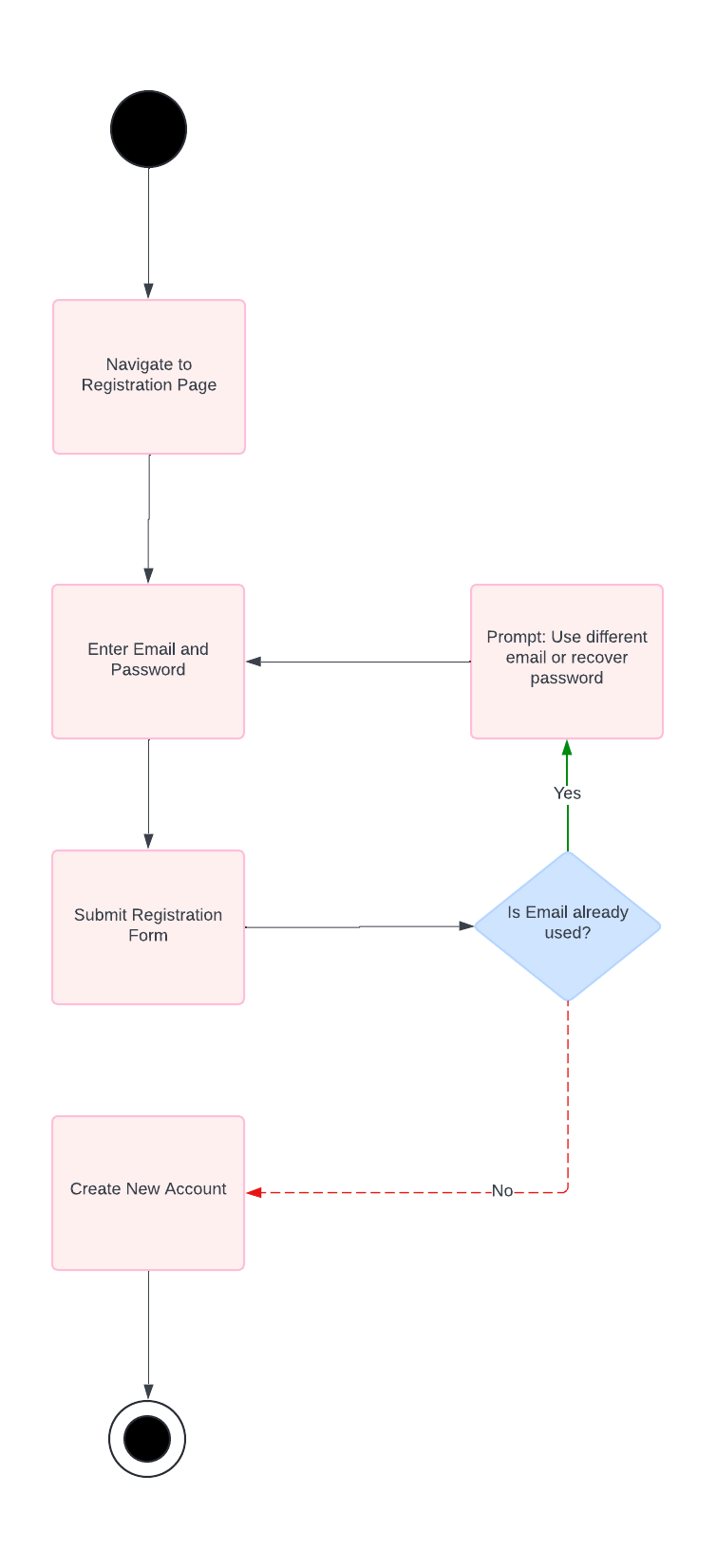


Figure 23: Activity - User Sign Up

### User Login

A diagram of a password

Description automatically generated

Figure 24: Activity - User Login

### User Recognition

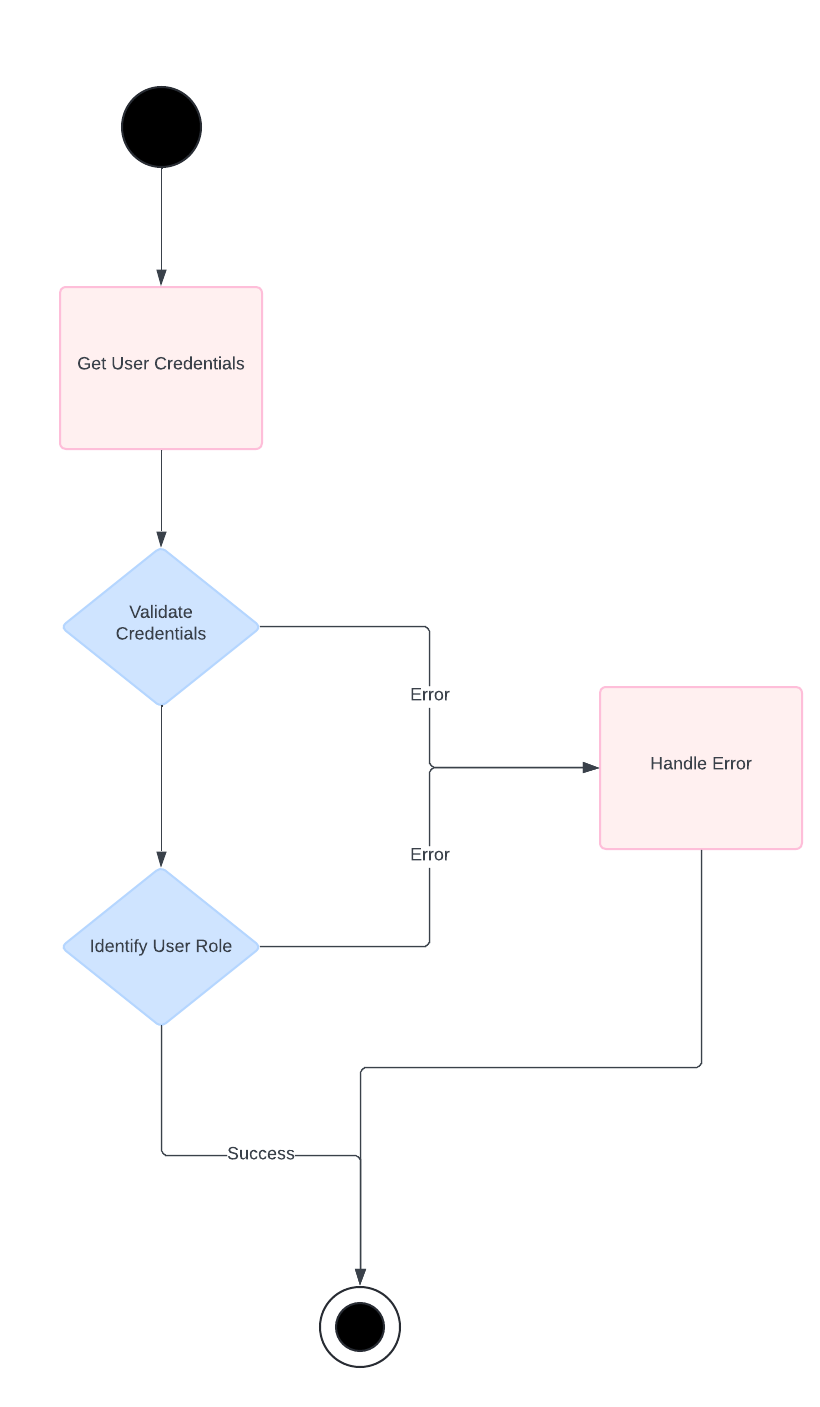


Figure 25: Activity - User Recognition

### Initiate Prediction Process

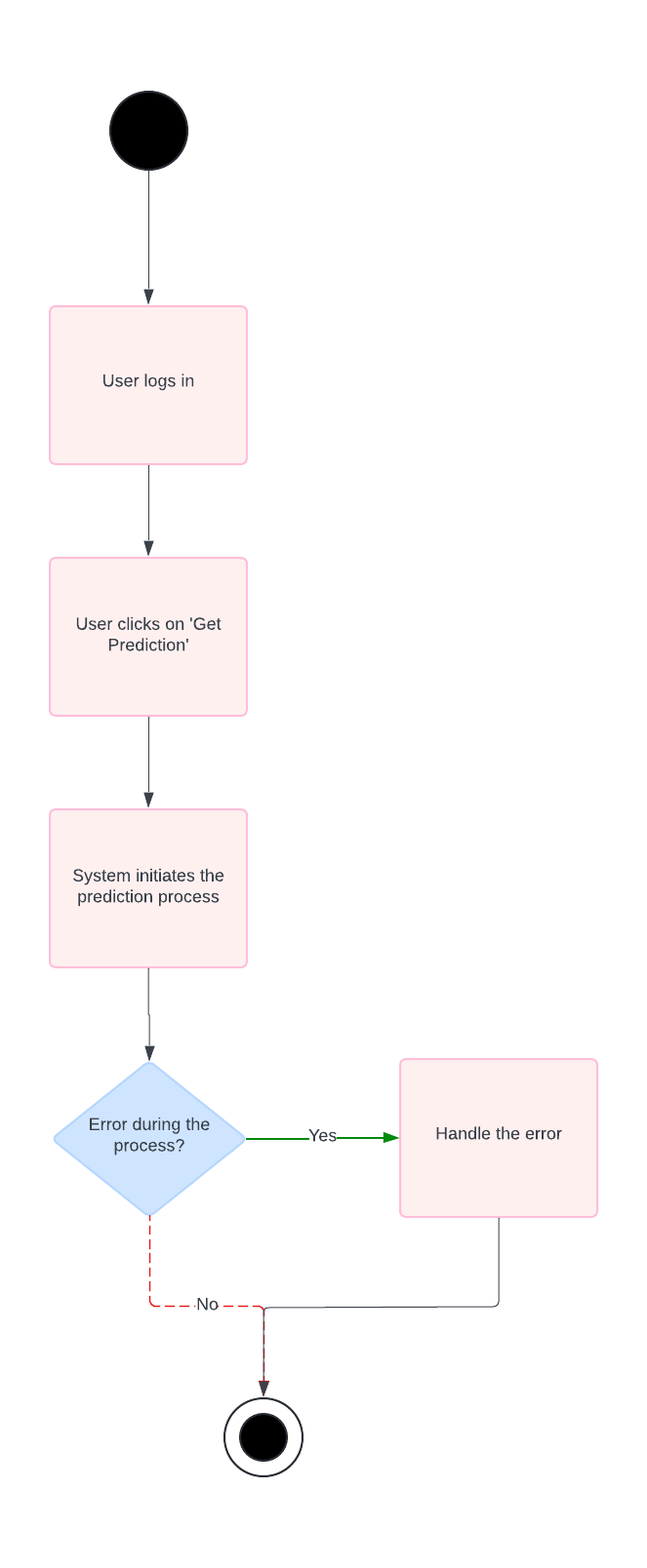


Figure 26: Activity - Initiate Prediction Process

### Upload ECG Signal Data

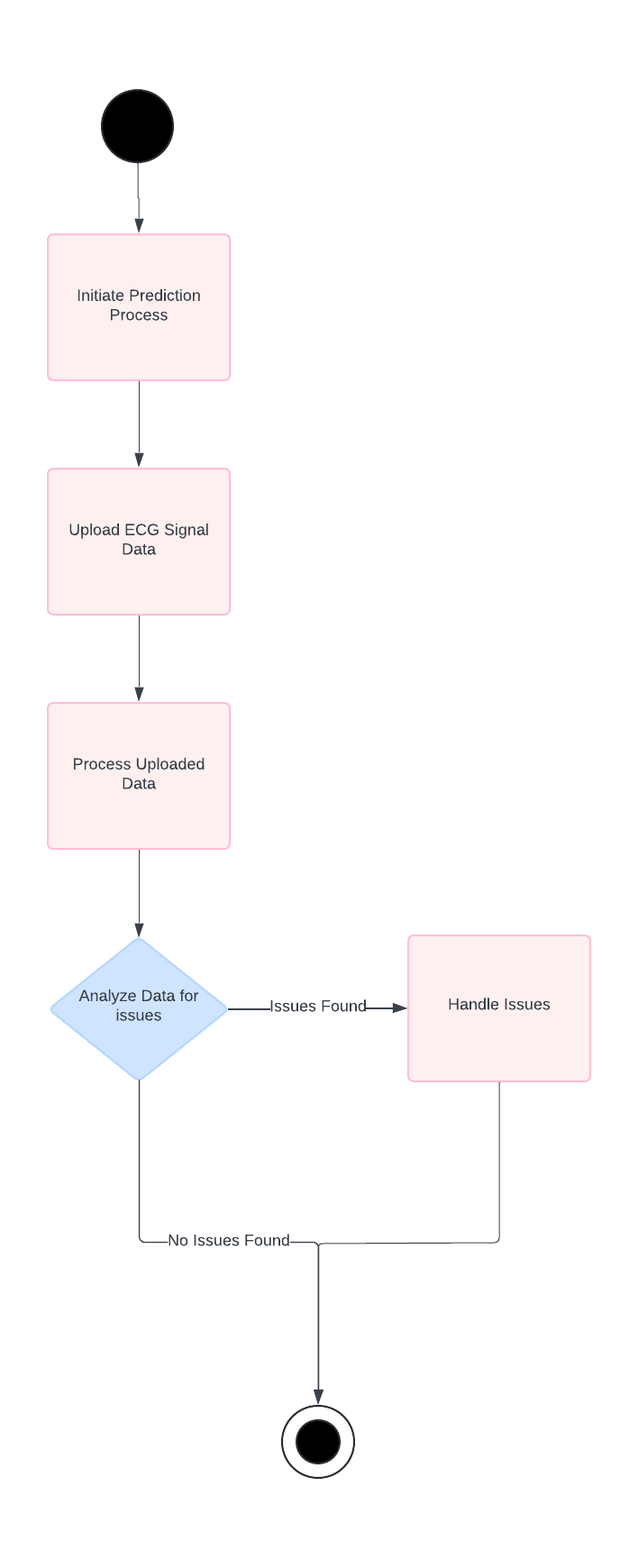


Figure 27: Activity - Upload ECG Signal Data

### ECG Signal Visualization

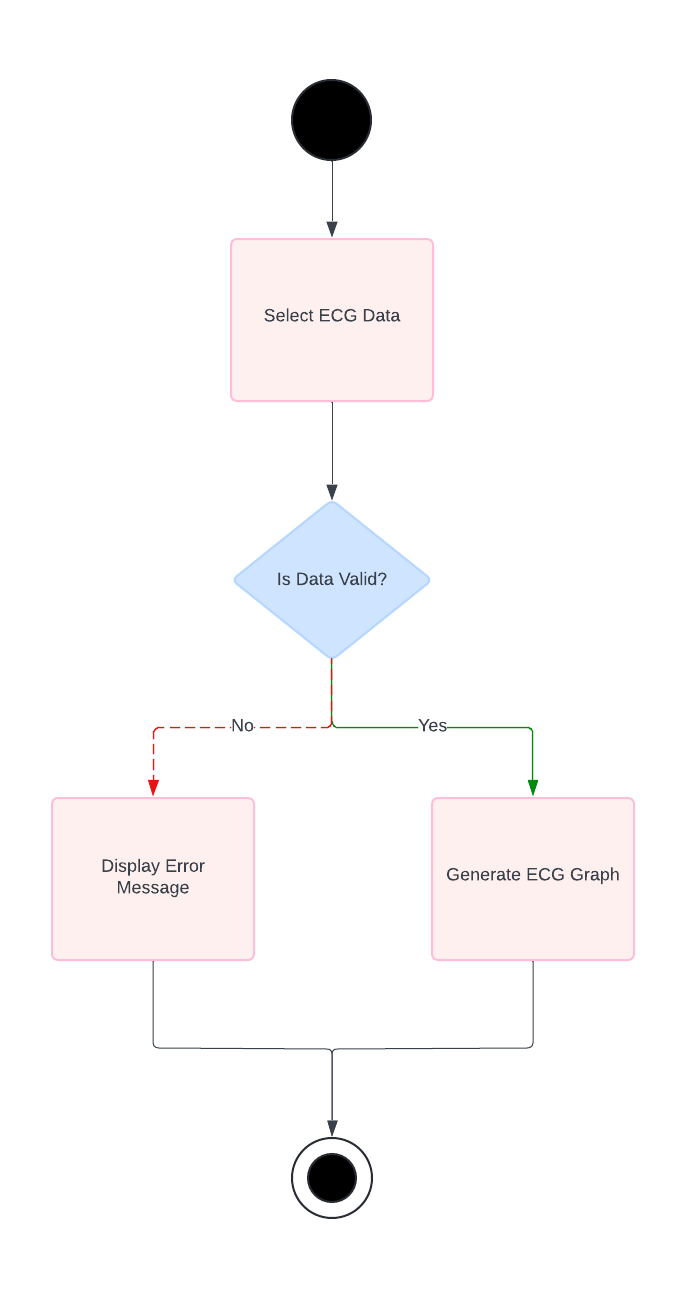


Figure 28: Activity - ECG Signal Visualization

### ECG Disease Prediction

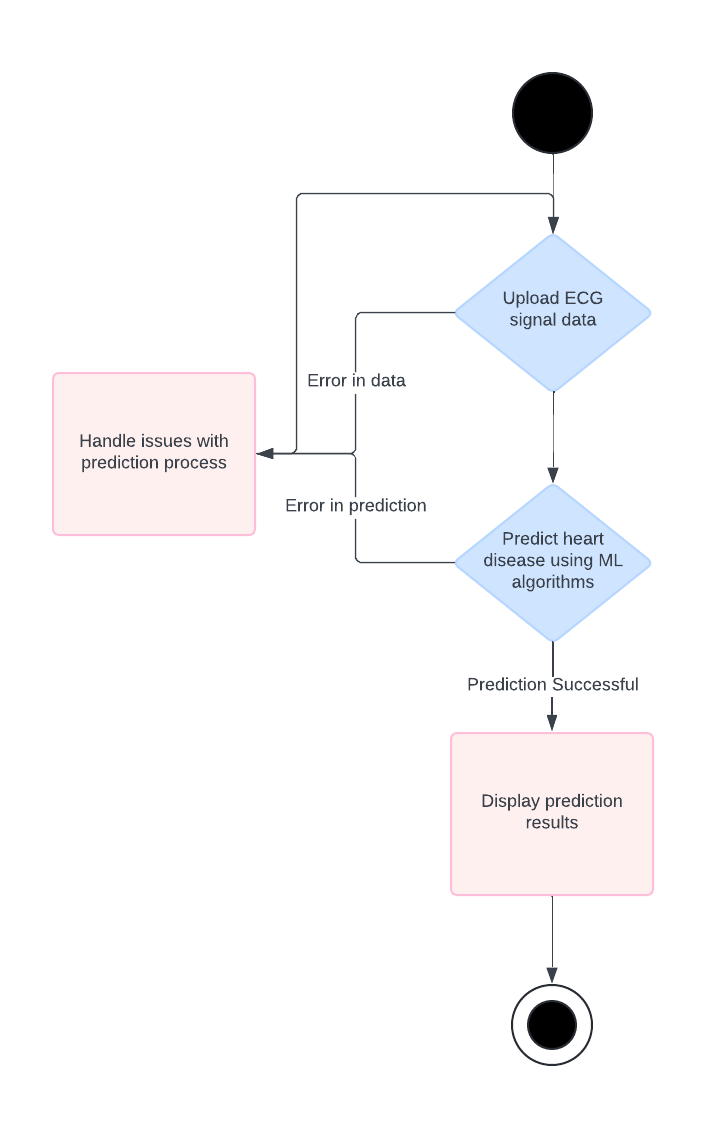


Figure 29: Activity - ECG Disease Prediction

### ECG Image Upload

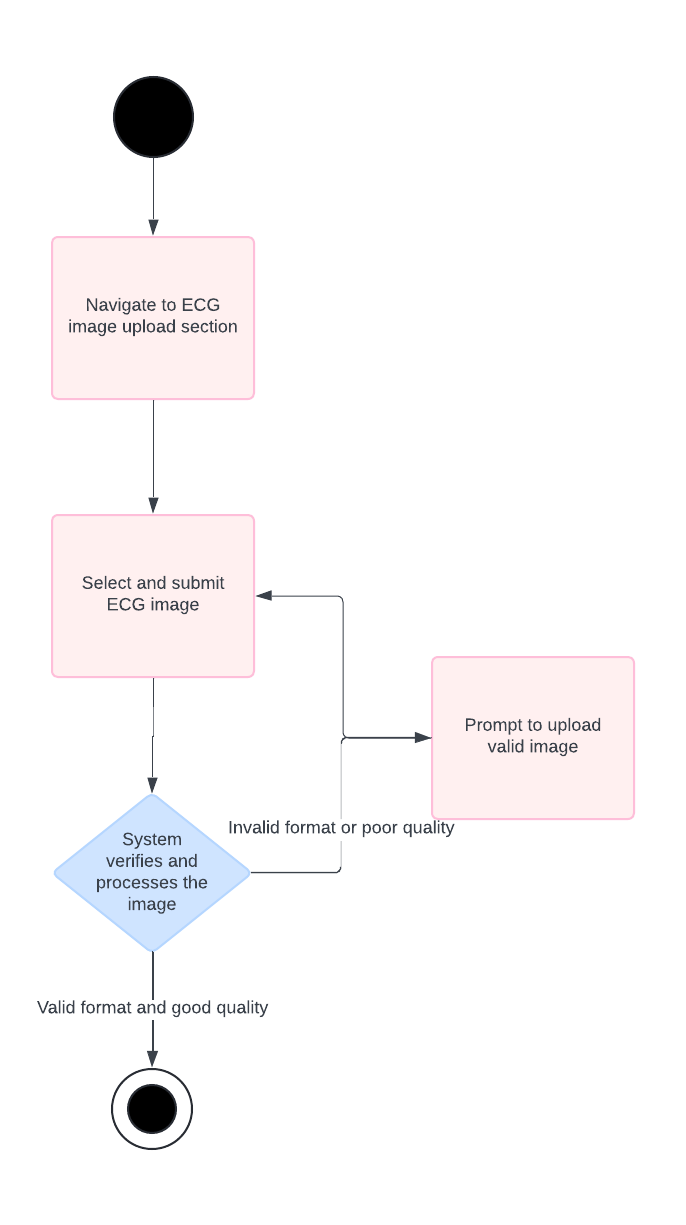


Figure 30: Activity - ECG Image Upload

### ECG Image Disease Prediction

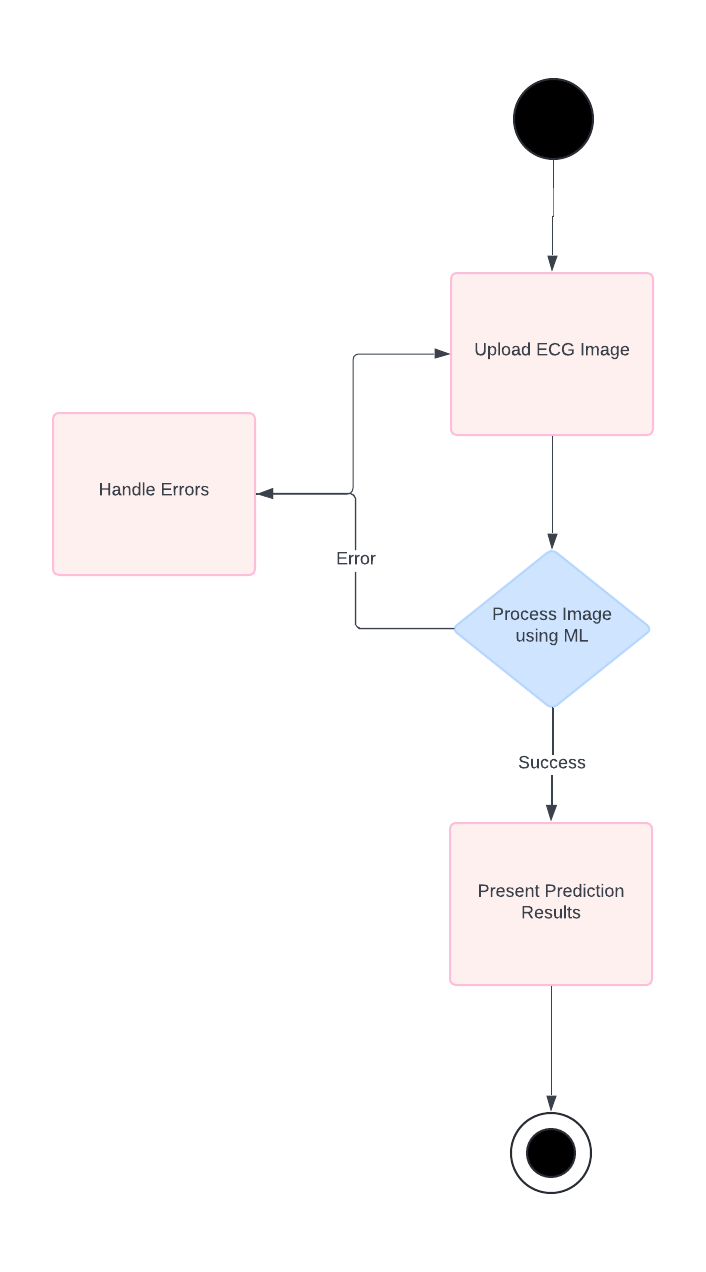


Figure 31: Activity - ECG Image Disease Prediction

### Prediction Display

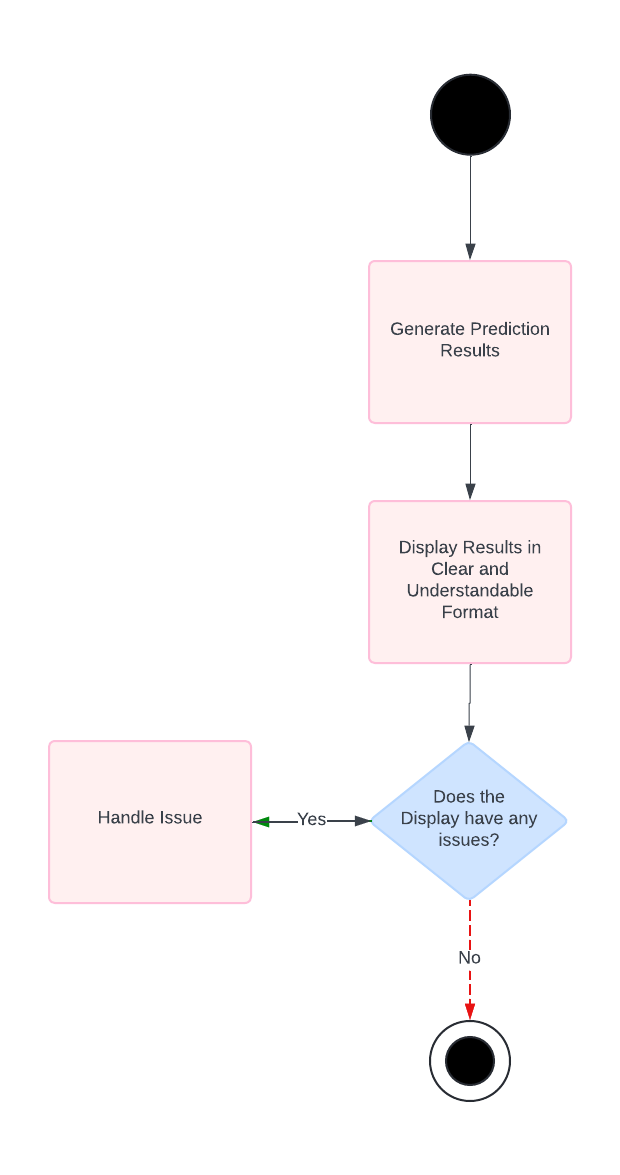


Figure 32: Activity - Prediction Display

### Chat Bot Integration

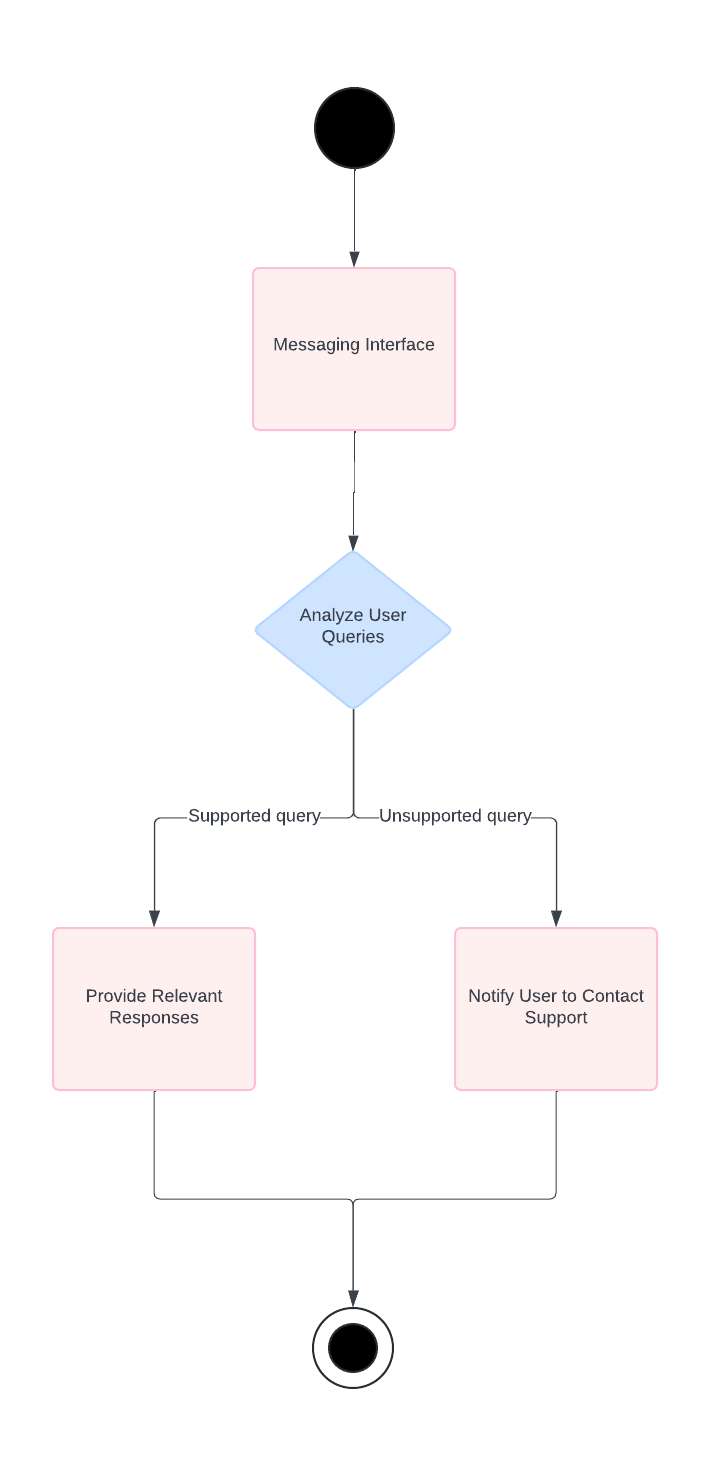


Figure 33: Activity - Chat Bot Integration

### Medical History Tracking

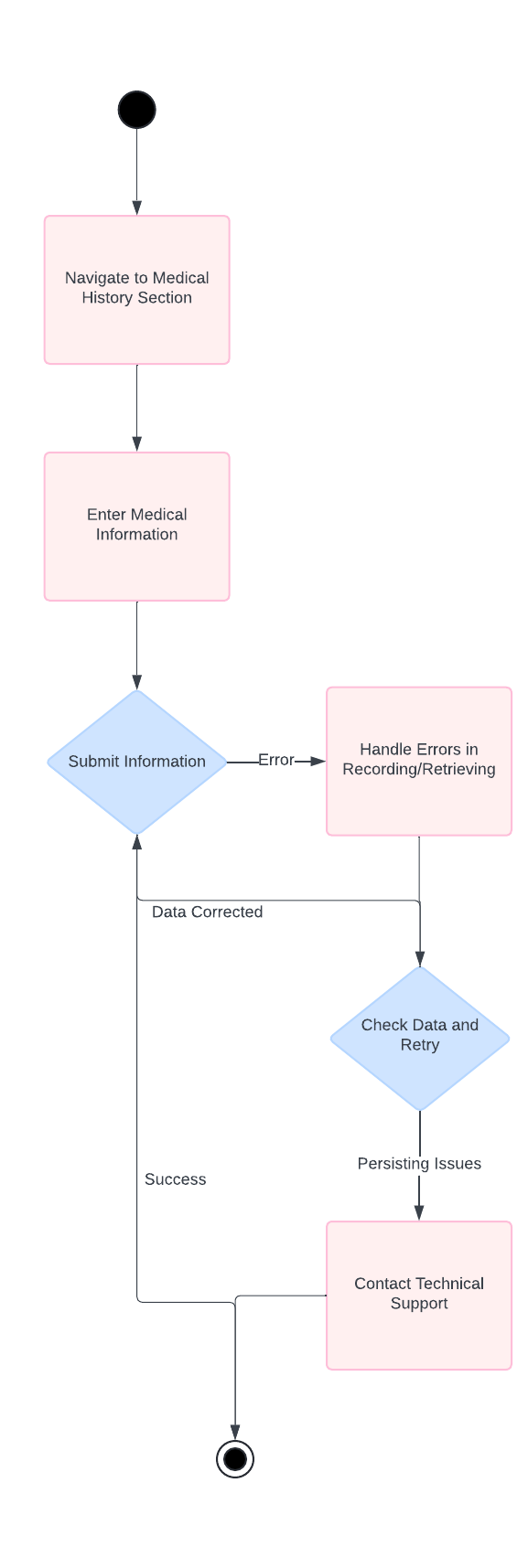


Figure 34: Activity - Medical History Tracking

### Doctor Browsing and Filtering

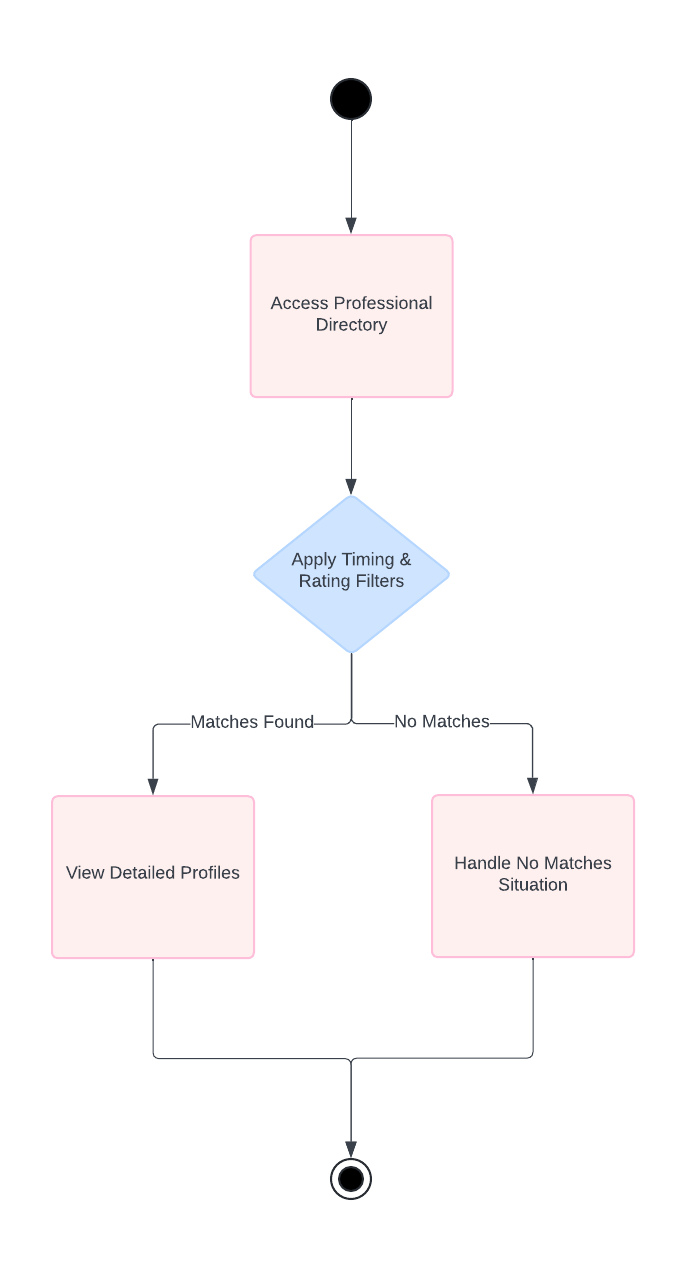


Figure 35: Activity - Doctor Browsing and Filtering

### Appointment Scheduling

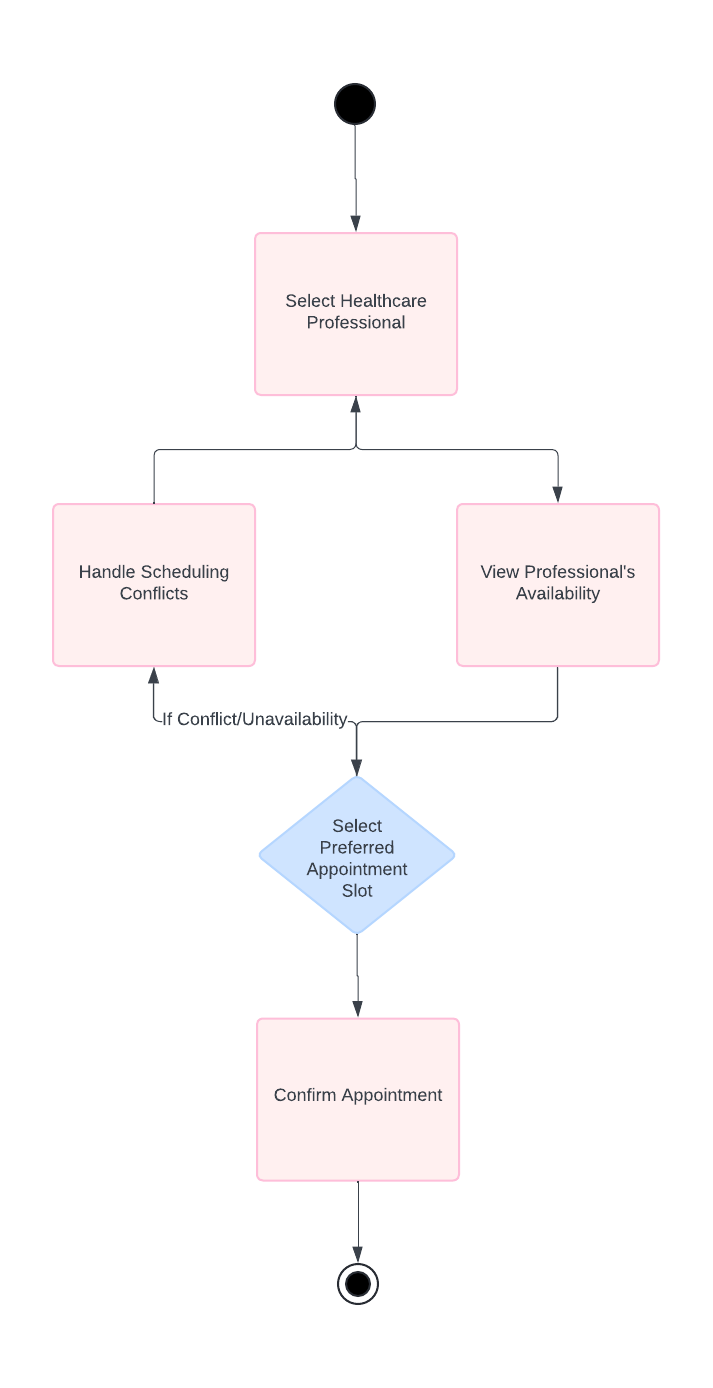


Figure 36: Activity - Appointment Scheduling

### Doctor Profile Management

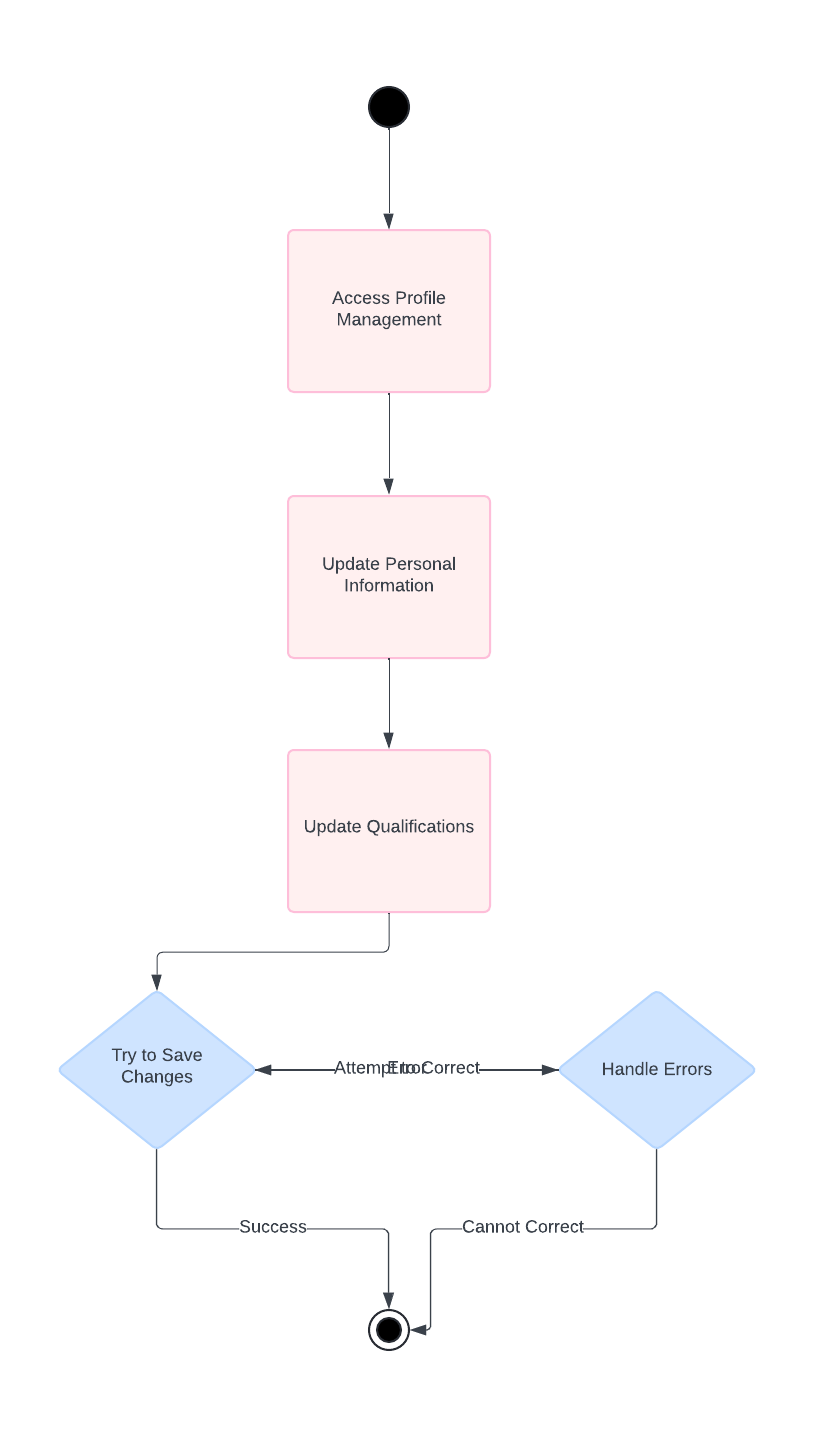


Figure 37: Activity - Doctor Profile Management

### Appointment Management

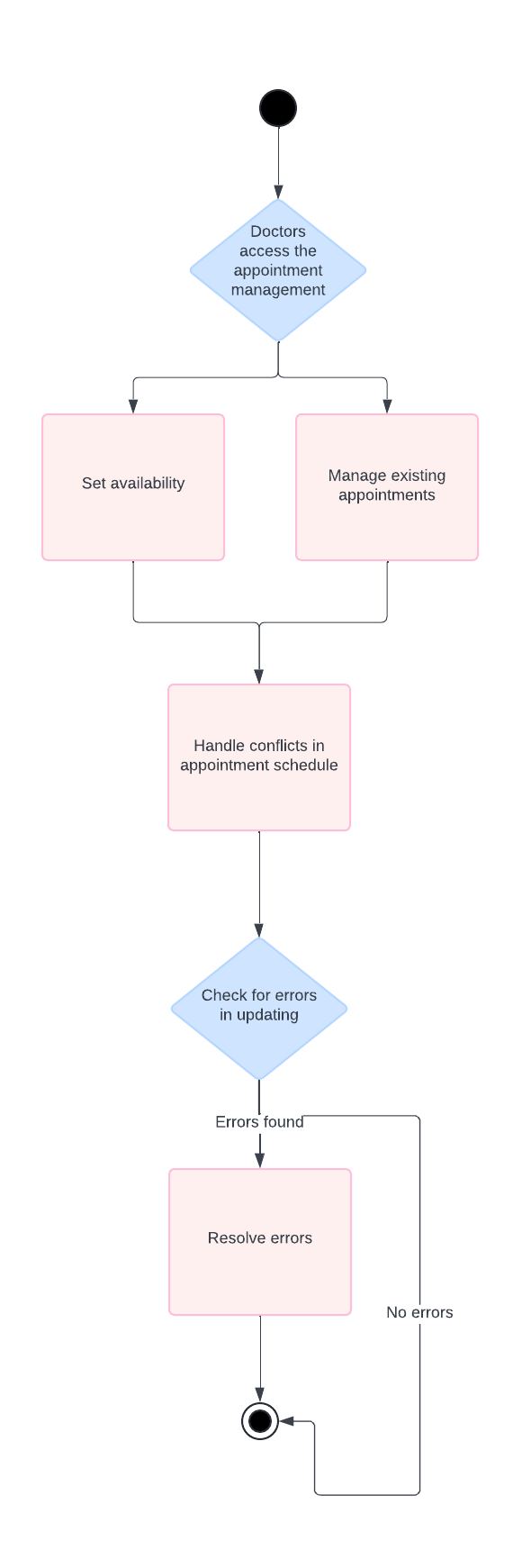


Figure 38: Activity - Appointment Management

### Doctor Patient Interaction

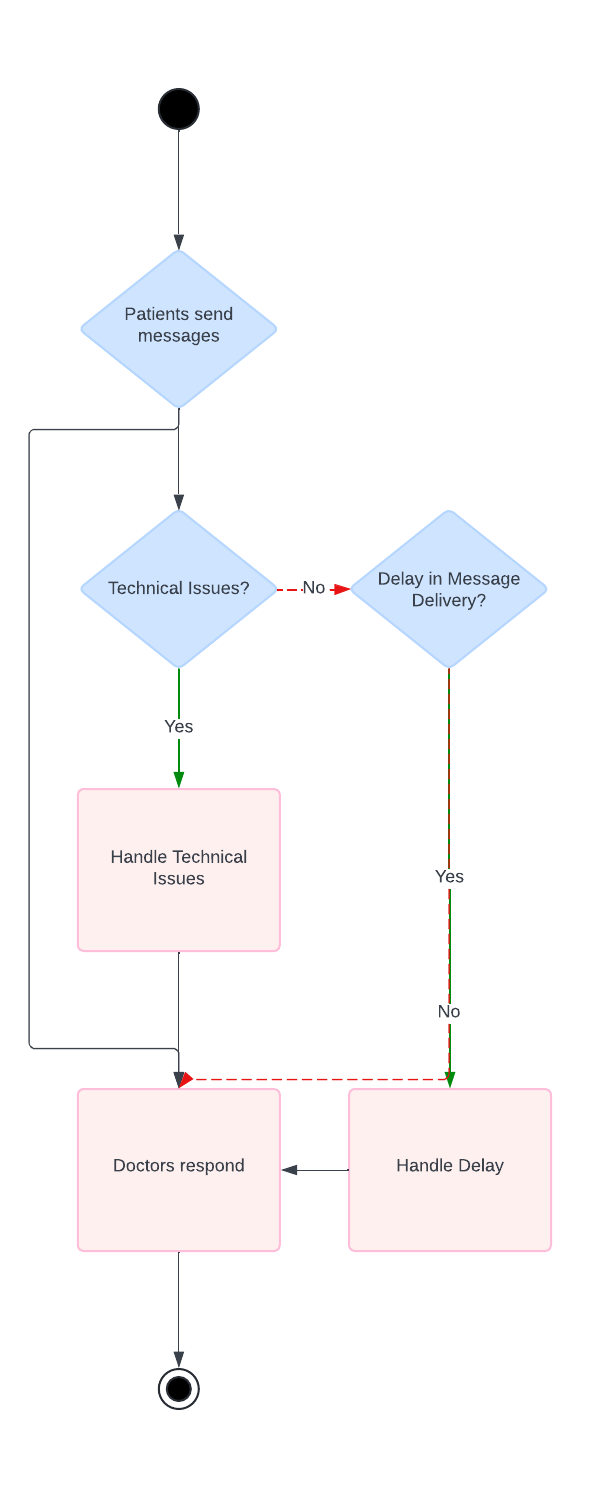


Figure 39: Activity - Doctor Patient Interaction

### Review and Rating System

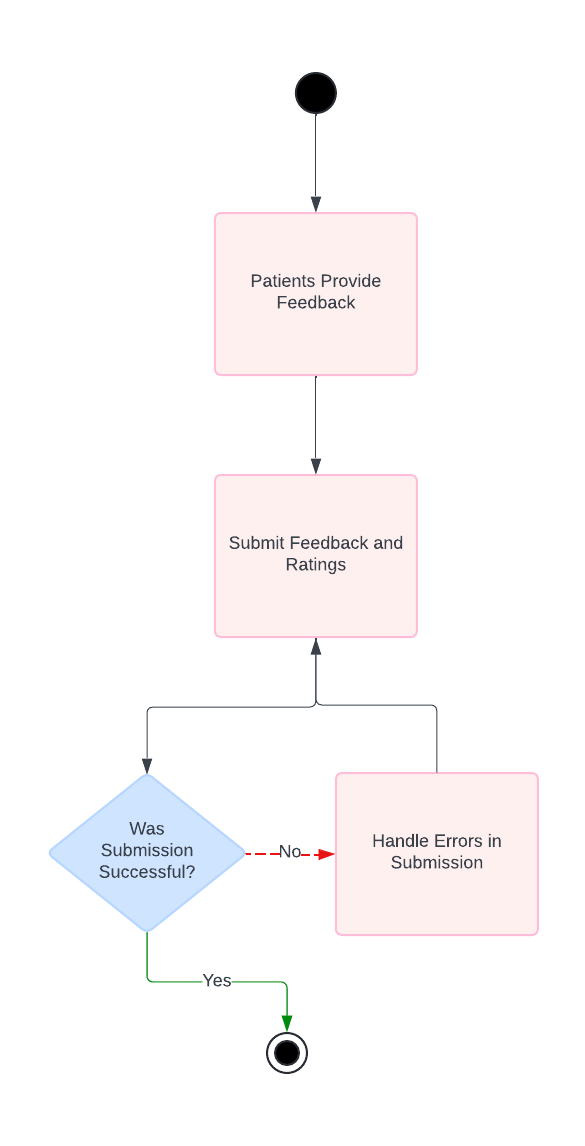


Figure 40: Activity - Review and Rating System

### Admin Panel

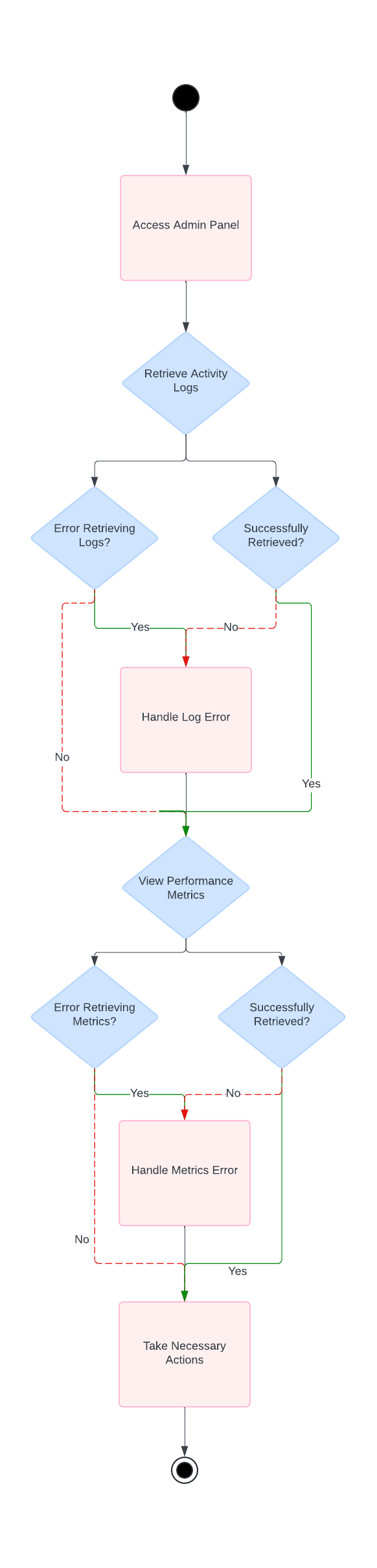


Figure 41: Activity - Admin Panel

## Sequence Diagram

### User Sign Up

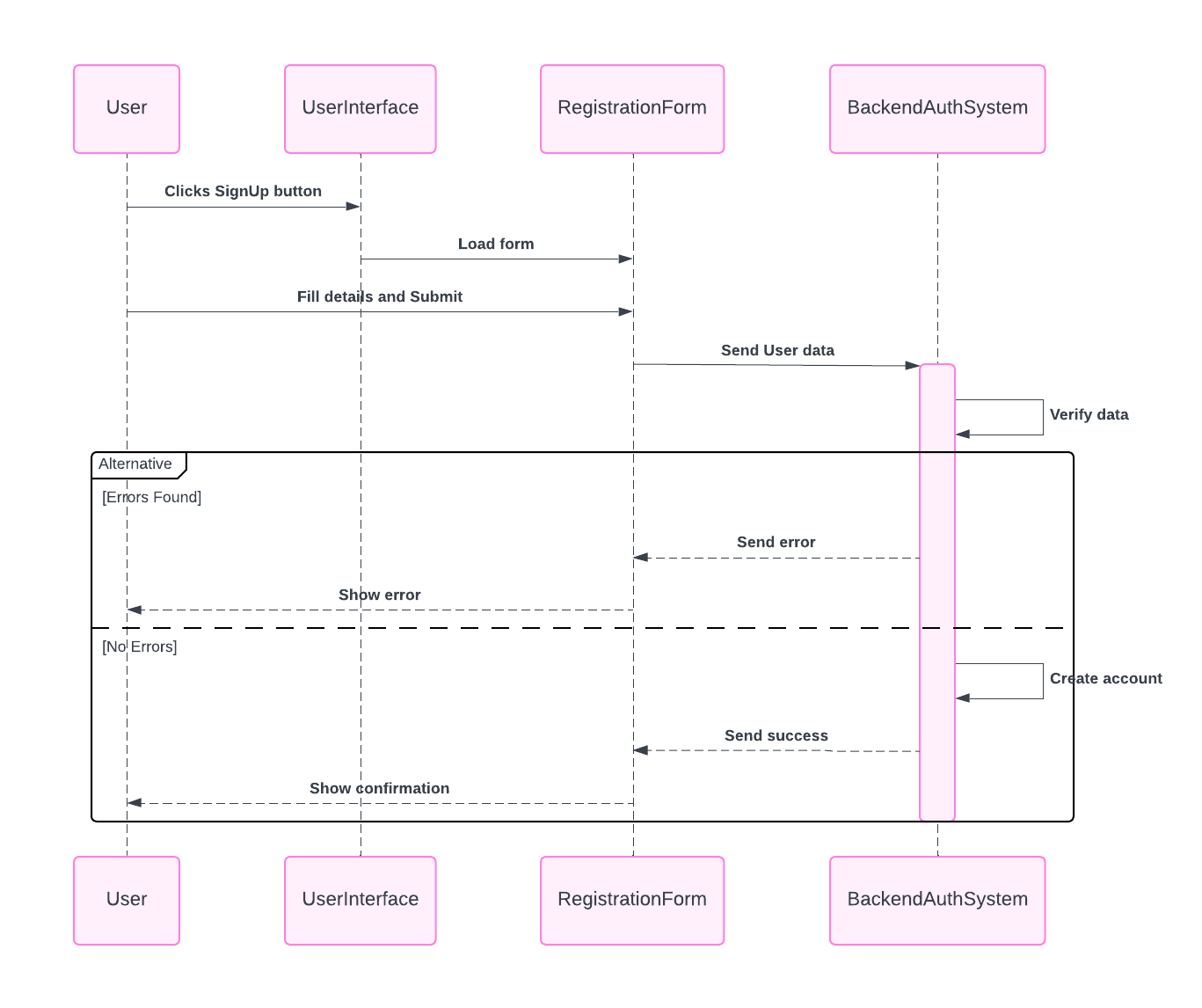


Figure 43: Sequence - User Sign Up

### User Login

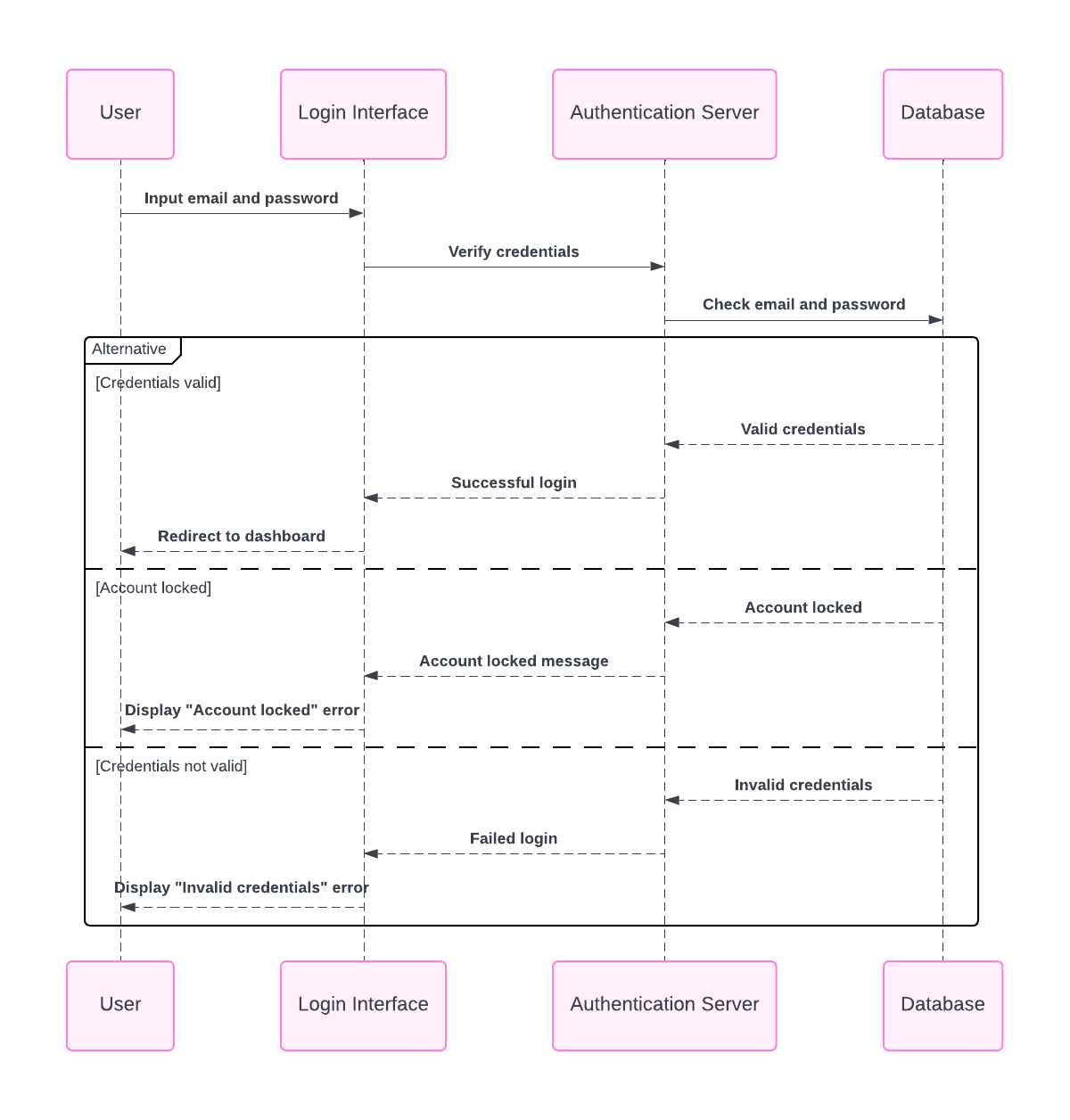


Figure 44: Sequence - User Login

### User Recognition

A diagram of a medical procedure

Description automatically generated

Figure 45: Sequence - User Recognition

### Initiate Prediction Process

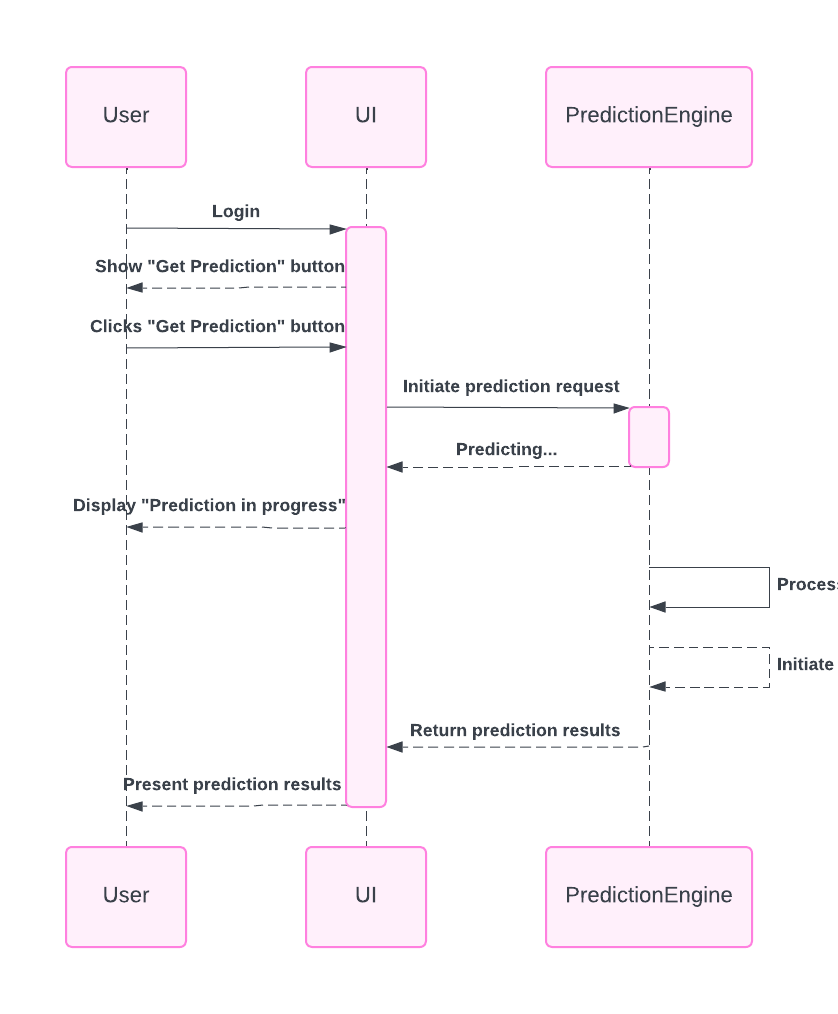


Figure 46: Sequence - Initiate Prediction Process

### Upload ECG Signal Data

A diagram of a computer

Description automatically generated

Figure 47: Sequence - Upload ECG Signal Data

### ECG Signal Visualization

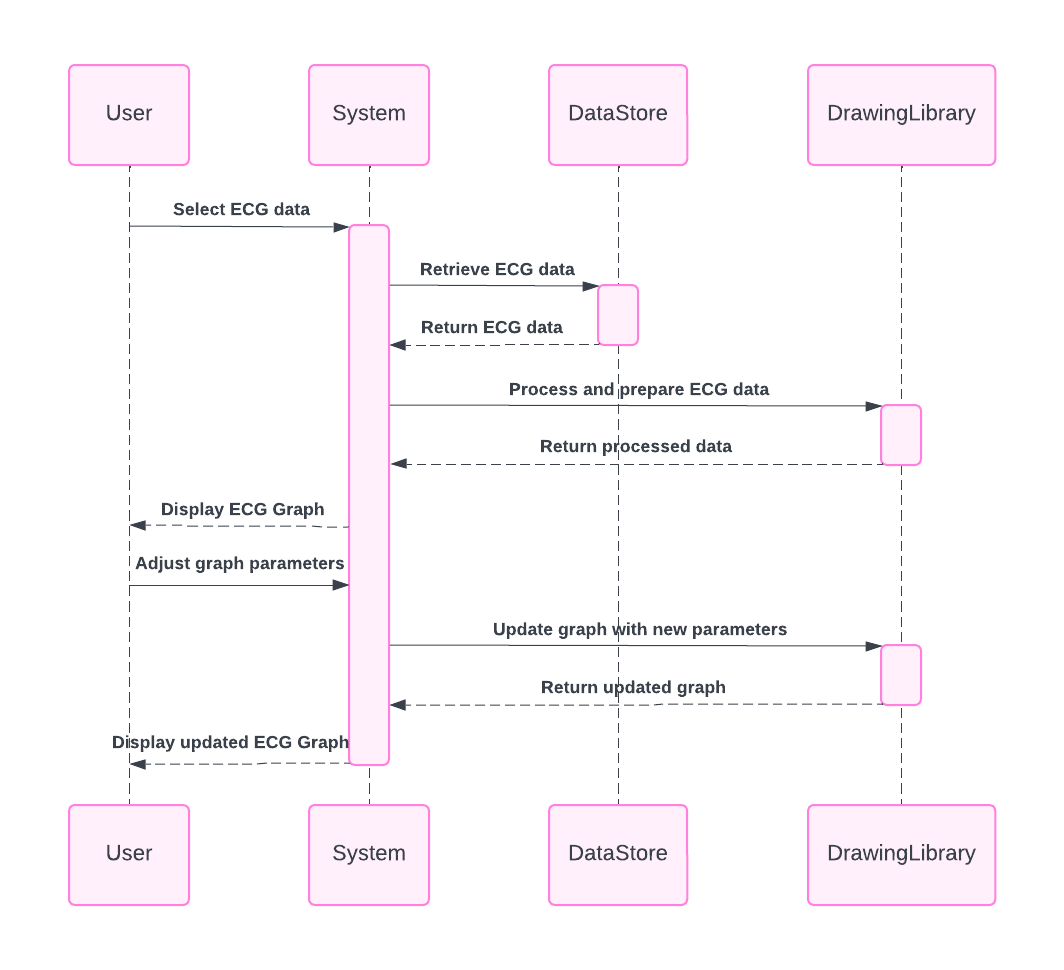


Figure 48: Sequence - ECG Signal Visualization

### ECG Disease Prediction

A diagram of a machine learning model

Description automatically generated

Figure 49: Sequence - ECG Disease Prediction

### ECG Image Upload

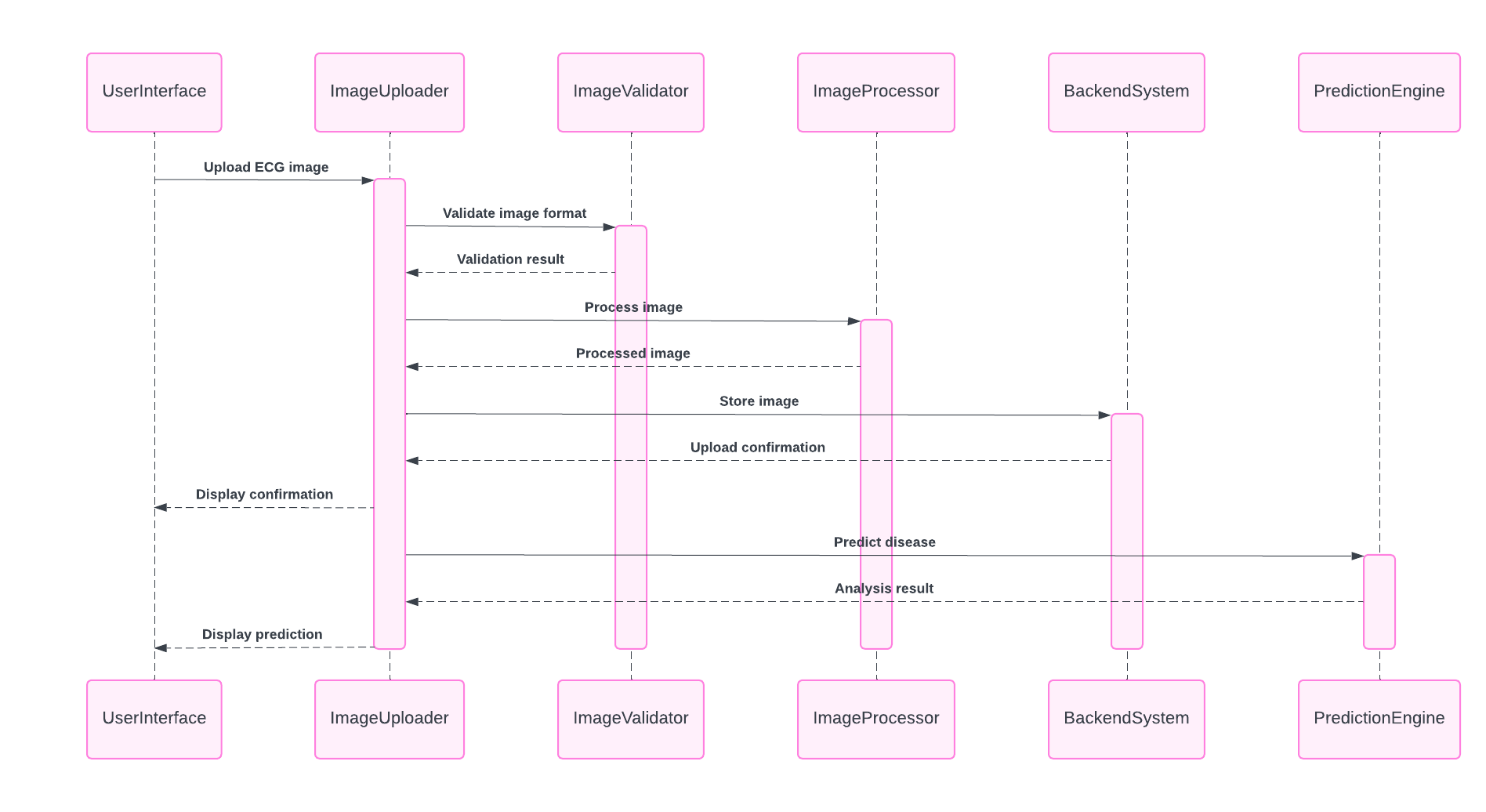


Figure 50: Sequence - ECG Image Upload

### ECG Image Disease Prediction

A diagram of a process

Description automatically generated

Figure 51: Sequence - ECG Image Disease Prediction

### Prediction Display

A diagram of a process

Description automatically generated

Figure 52: Sequence - Prediction Display

### Chat Bot Integration

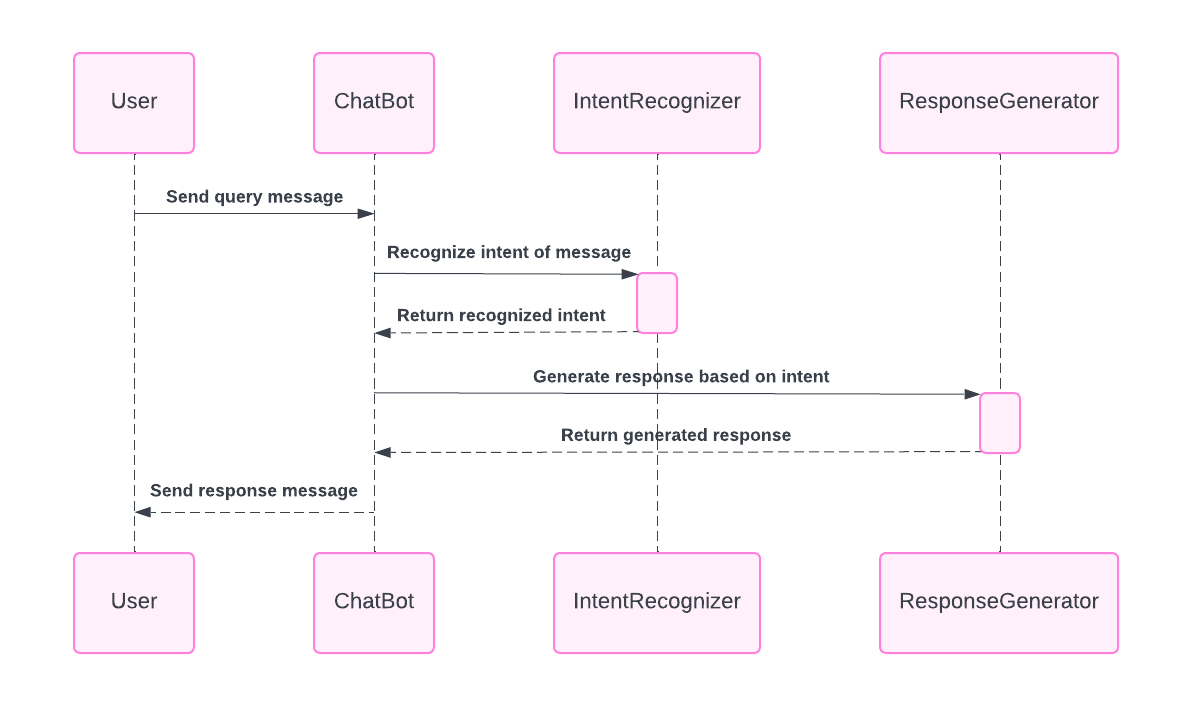


Figure 53: Sequence - Chat Bot Integration

### Medical History Tracking

A diagram of a data flow

Description automatically generated

Figure 54: Sequence - Medical History Tracking

### Doctor Browsing and Filtering

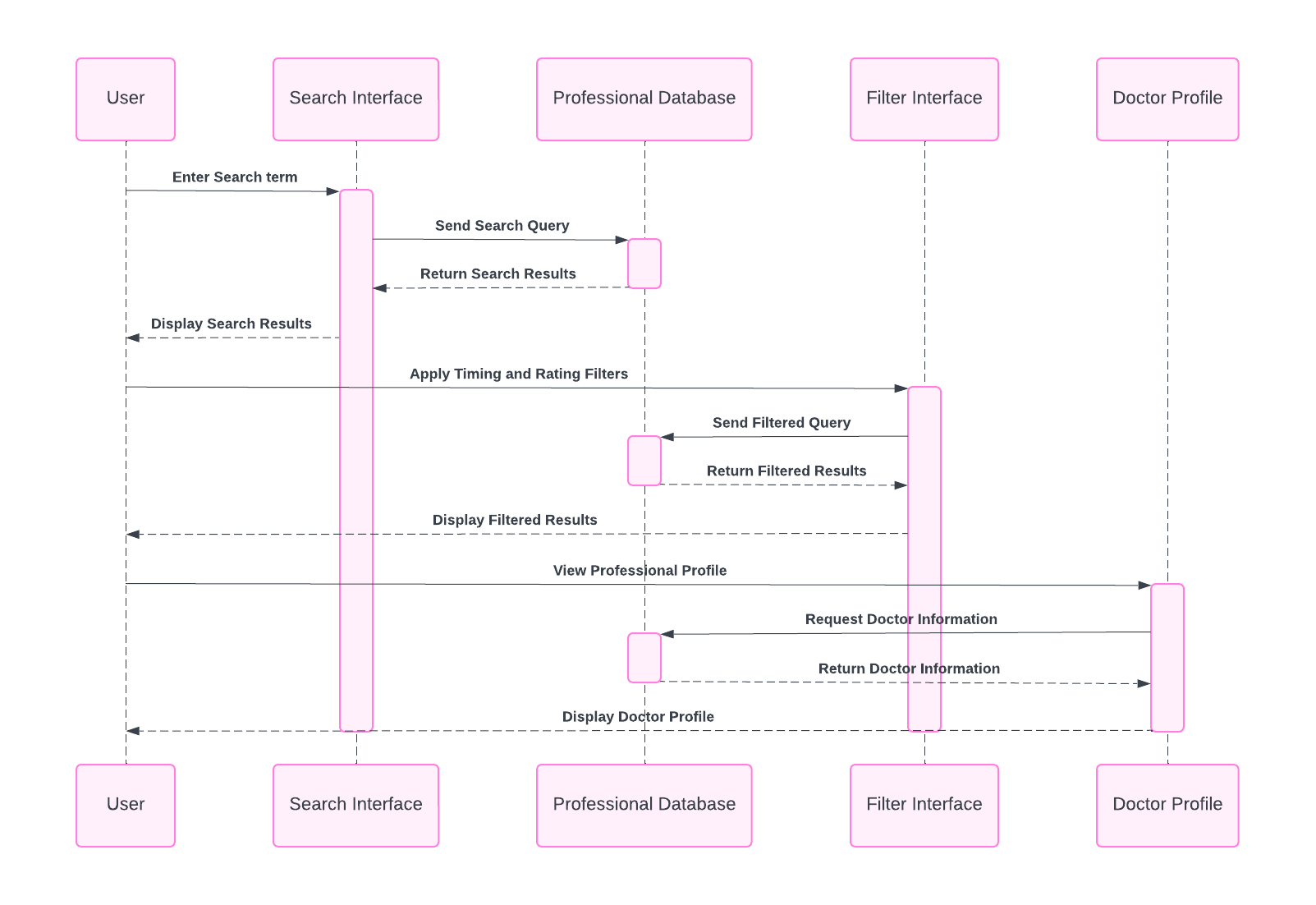


Figure 55: Sequence - Doctor Browsing and Filtering

### Appointment Scheduling

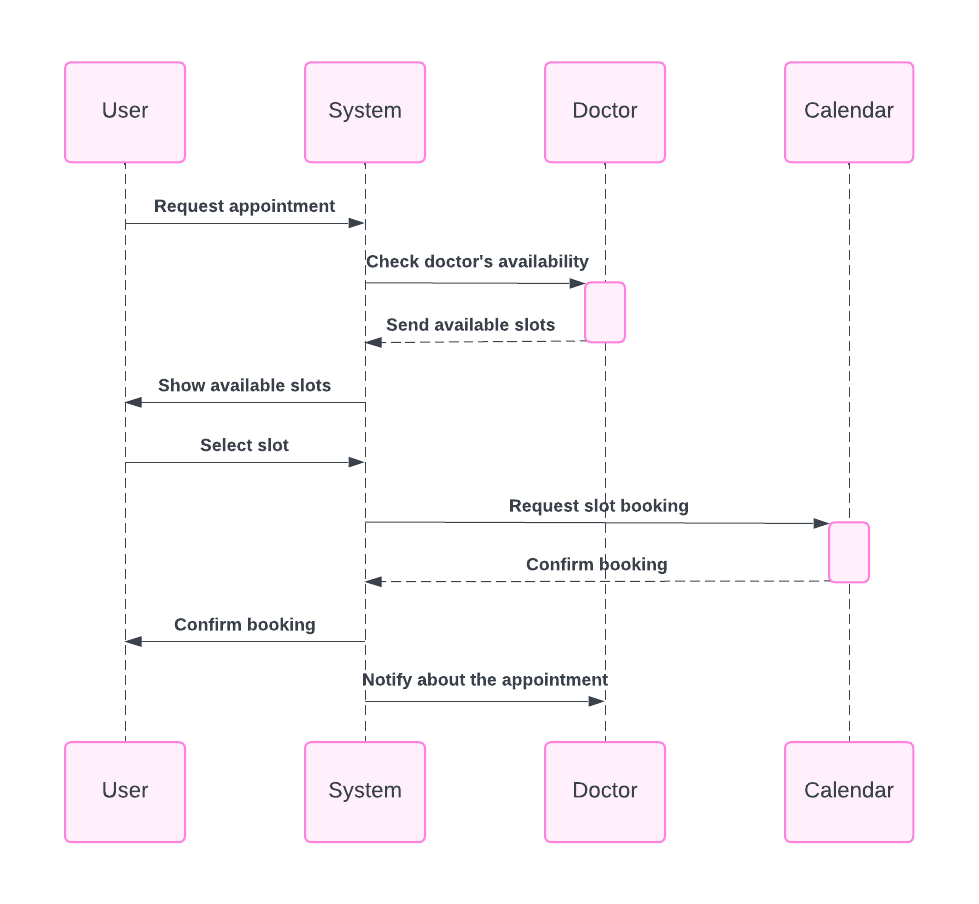


Figure 56: Sequence - Appointment Scheduling

### Doctor Profile Management

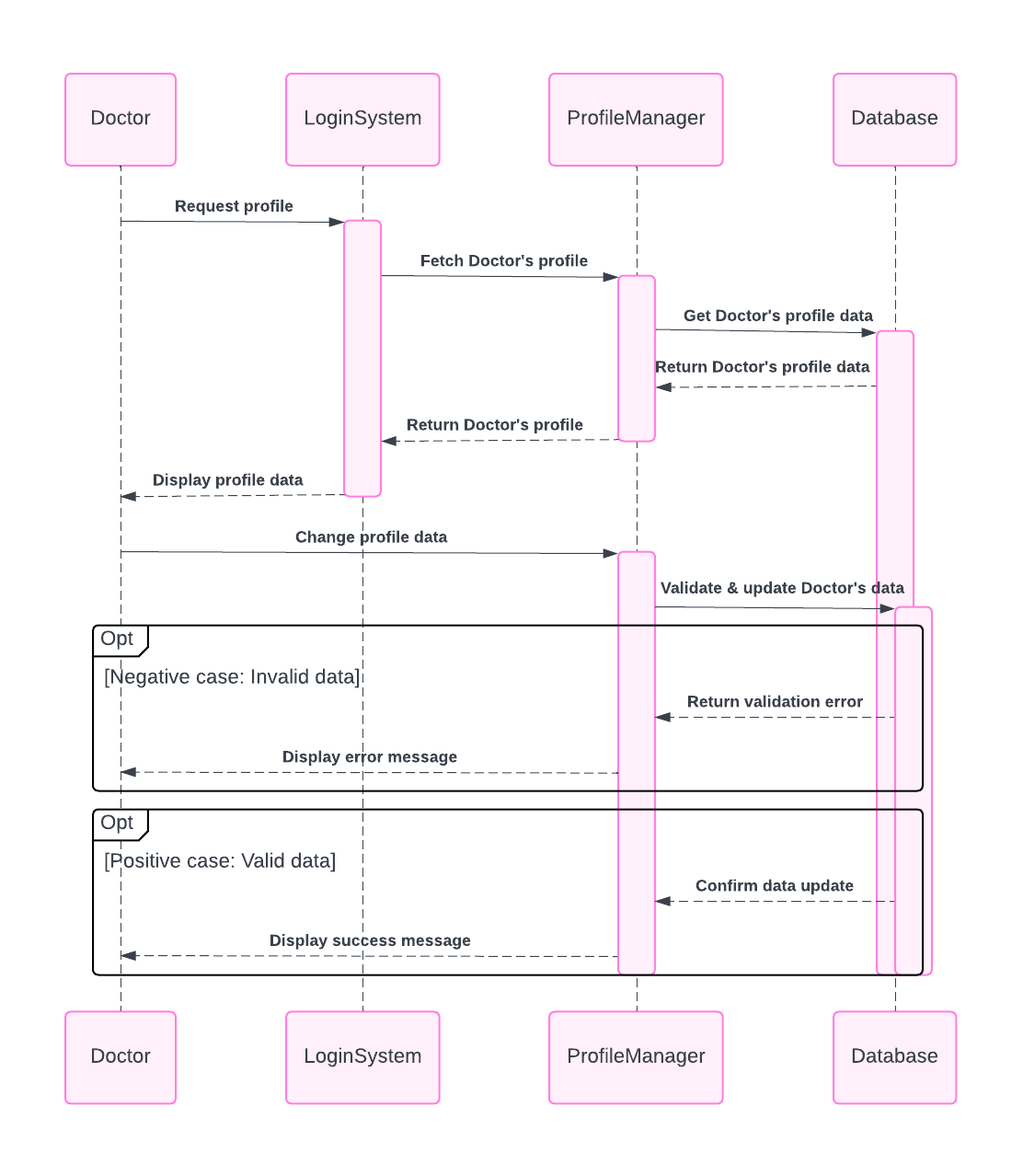


Figure 57: Sequence - Doctor Profile Management

### Appointment Management

A diagram of a schedule

Description automatically generated

Figure 58: Sequence - Appointment Management

### Doctor Patient Interaction

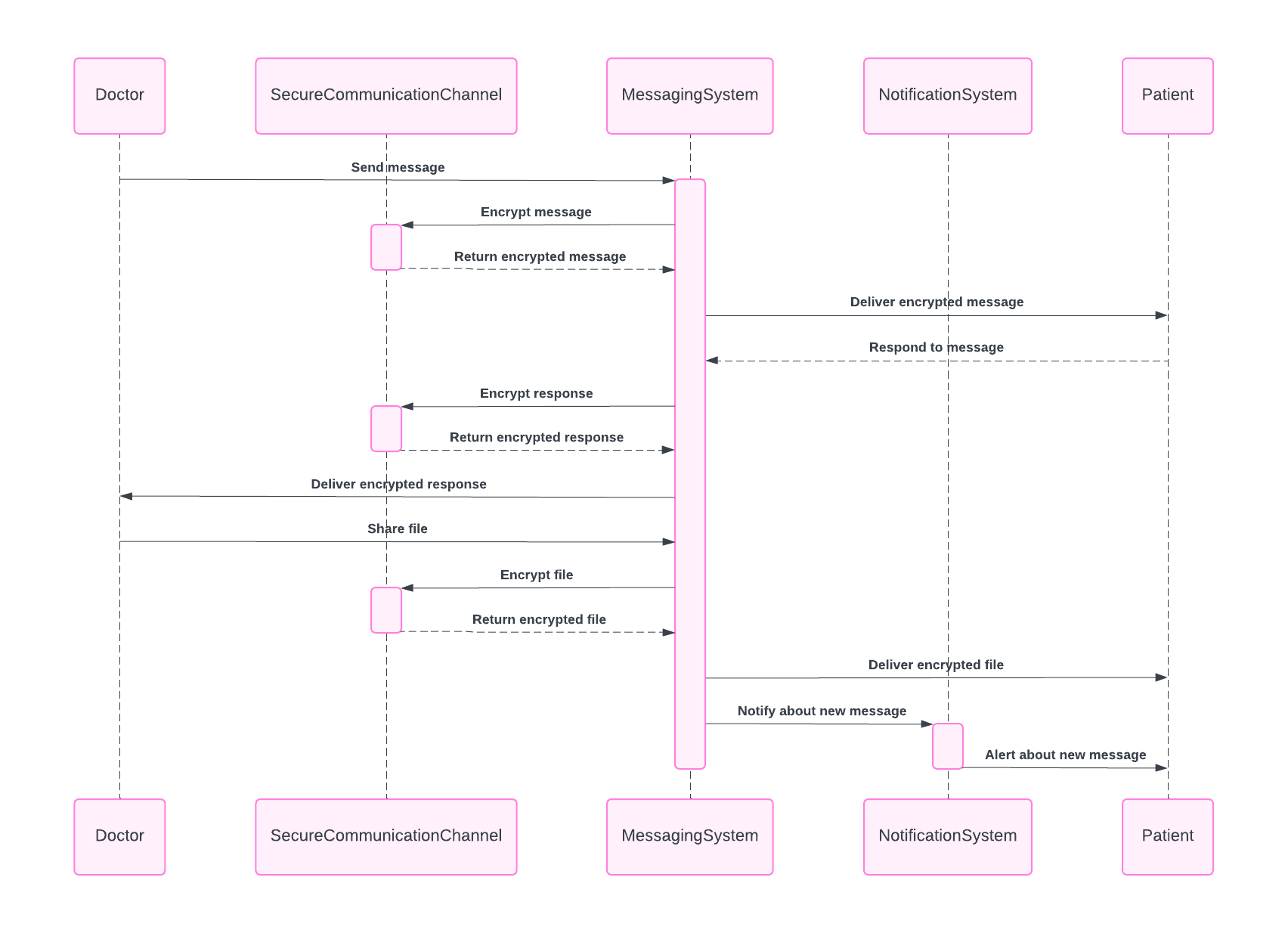


Figure 59: Sequence - Doctor Patient Interaction

### Review and Rating System

A diagram of a patient

Description automatically generated

Figure 60: Sequence - Review and Rating System

### Admin Panel

A diagram of a software company

Description automatically generated

Figure 61: Sequence - Admin

## Software Architecture

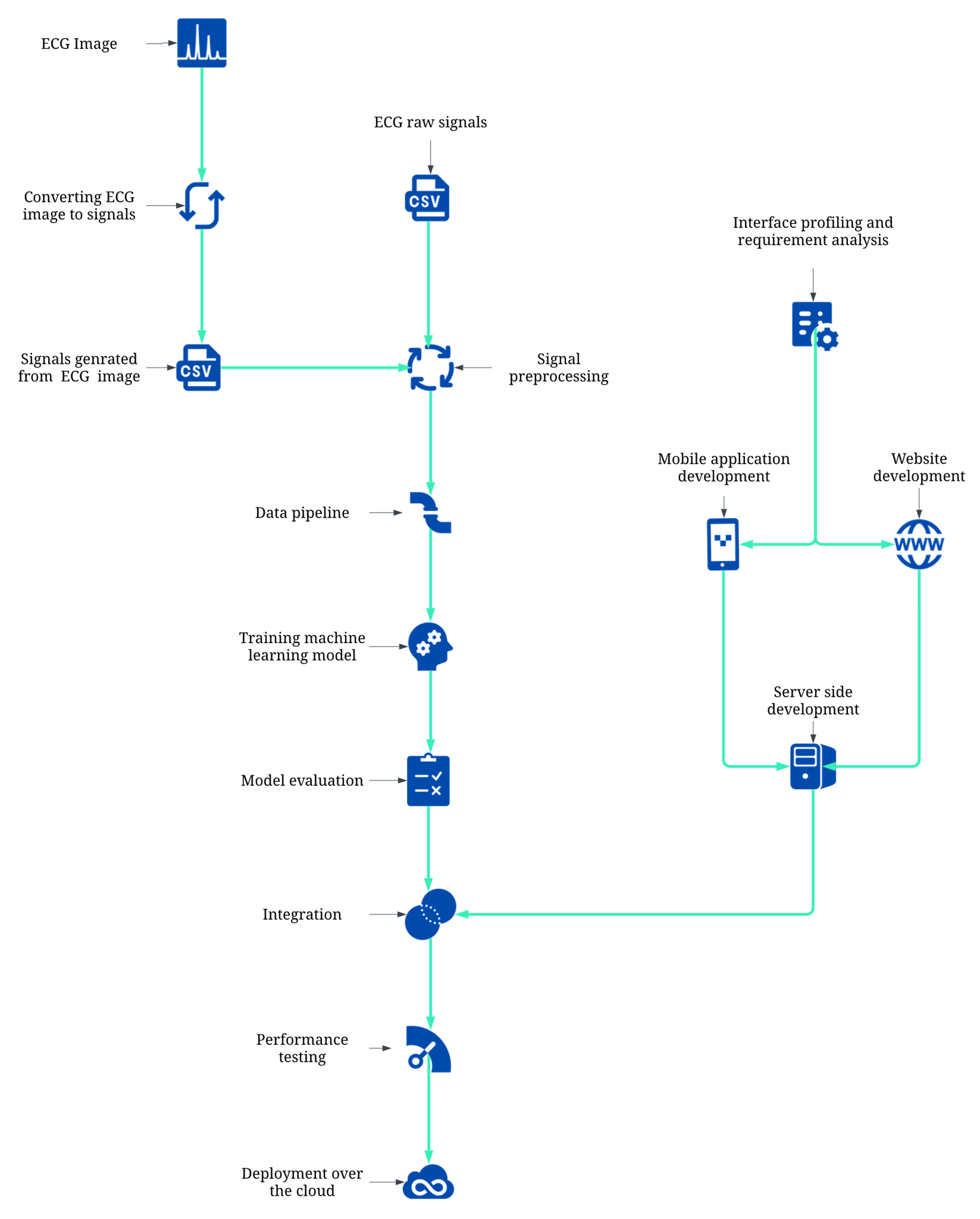


Figure 63: Software Architecture

## Class Diagram



Figure 64: Class Diagram

## Database Diagram

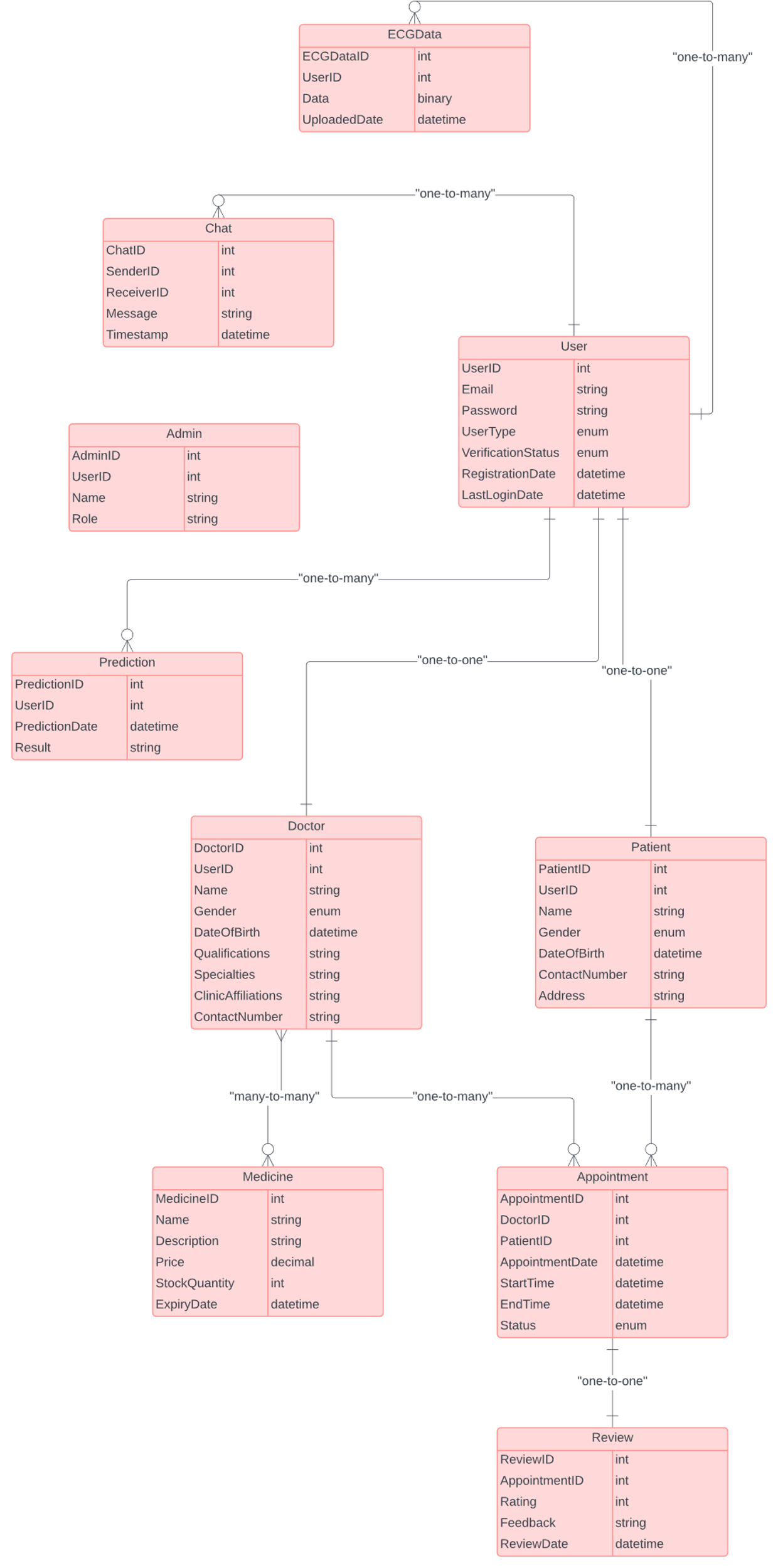


Figure 65: Database Diagram

## Network Diagram (Gantt Chart)

Table 42: Network Diagram (Gantt Chart)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Activity** | |  |  |  | | --- | --- | --- | | Semester 7 | Summer Break | Semester8 | |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec | |
| Project Planning and Research | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |
| Literature Review | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |
| Dataset Collection | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |
| Data Preprocessing and Pipeline Designing | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |
| Interface Profiling and Requirement Analysis | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |
| Model Development and Training | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |
| Website Development | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |
| Model Evaluation and Validation | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |
| Integration and Testing | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |
| Mobile App Development | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |
| Model Optimization and Enhancement | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |
| Interface Refinements | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |
| Backend Development and Integration | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |
| Deployment Process | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |
| Documentation and Reporting | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | |

Chapter 4 System Testing

# System Testing

## Test Cases

### User Sign Up

Table 43: TC-001

|  |  |
| --- | --- |
| **Test Case ID** | **TC001** |
| Test Case Name | User Sign Up - Valid Input |
| Test Objective | To verify that a new user can successfully sign up with valid input. |
| Test Inputs | Valid email address and password. |
| Expected Result | The user should be successfully registered and redirected to their dashboard. |
| Test Steps | 1. Navigate to the registration page. 2. Enter a valid email address and password. 3. Submit the registration form. 4. Verify that the user is redirected to their dashboard. |
| Actual Result | The user was successfully registered, and they were redirected to their dashboard. |
| Pass/Fail |  |
| Notes/Comments | No issues encountered during the sign-up process. |

### User Login

Table 44: TC-002

|  |  |
| --- | --- |
| **Test Case ID** | **TC002** |
| Test Case Name | User Login - Valid Credentials |
| Test Objective | To verify that a registered user can successfully log in with valid credentials. |
| Test Inputs | Registered email address and password. |
| Expected Result | The user should be successfully logged in and redirected to their dashboard. |
| Test Steps | 1. Navigate to the login page. 2. Enter registered email address and password. 3. Submit the login form. 4. Verify that the user is redirected to their dashboard. |
| Actual Result | The user was successfully logged in, and they were redirected to their dashboard. |
| Pass/Fail |  |
| Notes/Comments | No issues encountered during the login process. |

### User Recognition

Table 45: TC-003

|  |  |
| --- | --- |
| **Test Case ID** | **TC003** |
| Test Case Name | User Recognition - Valid Credentials |
| Test Objective | To verify that the system correctly identifies the user's role based on valid credentials. |
| Test Inputs | Valid user credentials. |
| Expected Result | The system should identify the user's role (doctor or patient) accurately. |
| Test Steps | 1. Log in or sign up with valid credentials. 2. Verify that the system correctly identifies the user's role. |
| Actual Result | The system identified the user's role as a patient. |
| Pass/Fail | Pass |
| Notes/Comments | User role identification functioned as expected. |

### Initiate Prediction Process

Table 46: TC-004

|  |  |
| --- | --- |
| **Test Case ID** | **TC004** |
| Test Case Name | Initiate Prediction Process - Button Click |
| Test Objective | To verify that users can initiate the prediction process by clicking the "Get Prediction" button. |
| Test Inputs | Click on the "Get Prediction" button. |
| Expected Result | The prediction process should be initiated. |
| Test Steps | 1. Log in to the system. 2. Click on the "Get Prediction" button. 3. Verify that the prediction process starts. |
| Actual Result | Upon clicking the "Get Prediction" button, the prediction process initiated successfully. |
| Pass/Fail |  |
| Notes/Comments | The prediction process started without any issues. |

### Upload ECG Signal Data

Table 47: TC-005

|  |  |
| --- | --- |
| **Test Case ID** | **TC005** |
| Test Case Name | Upload ECG Signal Data - Prompt Display |
| Test Objective | To verify that users are prompted to upload ECG signal data after clicking the "Get Prediction" button. |
| Test Inputs | Click on the "Get Prediction" button. |
| Expected Result | Users should be prompted to upload ECG signal data. |
| Test Steps | 1. Log in to the system. 2. Click on the "Get Prediction" button. 3. Verify that a prompt appears to upload ECG signal data. |
| Actual Result | After clicking the "Get Prediction" button, a prompt appeared asking to upload ECG signal data. |
| Pass/Fail |  |
| Notes/Comments | The prompt for uploading ECG signal data was displayed as expected. |

### ECG Signal Visualization

Table 48: TC-006

|  |  |
| --- | --- |
| **Test Case ID** | **TC006** |
| Test Case Name | ECG Signal Visualization - Graph Display |
| Test Objective | To verify that ECG signals are visually represented as an ECG graph. |
| Test Inputs | ECG signal data. |
| Expected Result | Users should be able to view ECG signals graphically plotted on the system interface. |
| Test Steps | 1. Upload valid ECG signal data. 2. View the ECG graph on the system interface. |
| Actual Result | The ECG signals were graphically plotted on the system interface as an ECG graph. |
| Pass/Fail |  |
| Notes/Comments | The ECG graph accurately depicted the characteristics of the ECG signals. |

### ECG Disease Prediction

Table 49: TC-007

|  |  |
| --- | --- |
| **Test Case ID** | **TC007** |
| Test Case Name | ECG Disease Prediction - Analysis Trigger |
| Test Objective | To verify that the system predicts heart disease after analysing the uploaded ECG data. |
| Test Inputs | Uploaded ECG signal data. |
| Expected Result | Prediction results regarding potential cardiac abnormalities or conditions should be presented to the user. |
| Test Steps | 1. Upload valid ECG signal data. 2. Analyse the data using the machine learning model. 3. Verify the prediction results. |
| Actual Result | The system successfully analysed the uploaded ECG data and presented prediction results regarding potential cardiac abnormalities. |
| Pass/Fail |  |
| Notes/Comments | The prediction results were clear and accurately presented to the user. |

### ECG Image Upload

Table 50: TC-008

|  |  |
| --- | --- |
| **Test Case ID** | **TC008** |
| Test Case Name | ECG Image Upload - Image Selection |
| Test Objective | To verify that users can select and upload ECG images for the prediction. |
| Test Inputs | Select an ECG image for upload. |
| Expected Result | Users should be able to upload ECG images from their device. |
| Test Steps | 1. Navigate to the ECG image upload section. 2. Select an ECG image for upload. 3. Verify successful upload. |
| Actual Result | Users were able to select and upload ECG images from their device successfully. |
| Pass/Fail |  |
| Notes/Comments | The system accepted common image formats such as JPEG, PNG, and JPG for upload. |

### ECG Image Disease Prediction

Table 51: TC-009

|  |  |
| --- | --- |
| **Test Case ID** | **TC009** |
| Test Case Name | ECG Image Disease Prediction - Analysis Trigger |
| Test Objective | To verify that the system analyses uploaded ECG images to predict heart disease using machine learning algorithms. |
| Test Inputs | Uploaded ECG image. |
| Expected Result | Prediction results regarding potential cardiac abnormalities or conditions should be presented to the user. |
| Test Steps | 1. Upload a valid ECG image. 2. Process the image using machine learning algorithms. 3. Verify the prediction results. |
| Actual Result | The system successfully analysed the uploaded ECG image and presented prediction results regarding potential cardiac abnormalities. |
| Pass/Fail |  |
| Notes/Comments | The prediction results were based on the analysis of the uploaded ECG image. |

### Prediction Display

Table 52: TC-010

|  |  |
| --- | --- |
| **Test Case ID** | **TC010** |
| Test Case Name | Prediction Display - Result Presentation |
| Test Objective | To verify that the system presents the predictions of heart disease to the user in a clear and understandable format. |
| Test Inputs | Prediction results. |
| Expected Result | Prediction results should be displayed clearly and intuitively on the system interface. |
| Test Steps | 1. View the prediction results on the system interface. 2. Verify the clarity and understandability of the displayed predictions. |
| Actual Result | The prediction results were presented in a clear and understandable format on the system interface. |
| Pass/Fail |  |
| Notes/Comments | The presentation of prediction results met the user's expectations and provided relevant insights into potential cardiac abnormalities. |

### Chat Bot Integration

Table 53: TC-011

|  |  |
| --- | --- |
| **Test Case ID** | **TC011** |
| Test Case Name | Chat Bot Integration - Interaction |
| Test Objective | To verify that the chat bot feature interacts with users effectively, understanding queries related to prediction analysis, health concerns, medication inquiries, and dietary requirements. |
| Test Inputs | User queries. |
| Expected Result | The chat bot should provide relevant responses and assistance based on user queries. |
| Test Steps | 1. Interact with the chat bot through the messaging interface. 2. Pose various queries related to prediction analysis, health concerns, medication inquiries, and dietary requirements. 3. Evaluate the chat bot's responses. |
| Actual Result | The chat bot effectively understood and responded to user queries, providing relevant assistance and information. |
| Pass/Fail |  |
| Notes/Comments | The chat bot's conversational abilities and understanding of natural language queries were commendable. |

### Medical History Tracking

Table 54: TC-012

|  |  |
| --- | --- |
| **Test Case ID** | **TC012** |
| Test Case Name | Medical History Tracking - Data Recording |
| Test Objective | To verify that users can track their medical history effectively on the platform. |
| Test Inputs | User-entered medical information. |
| Expected Result | User medical history should be securely recorded and accessible for reference. |
| Test Steps | 1. Navigate to the medical history section. 2. Enter relevant medical information. 3. Verify the recording and accessibility of the medical history data. |
| Actual Result | Users were able to track their medical history effectively on the platform, and the recorded information was securely stored and accessible. |
| Pass/Fail |  |
| Notes/Comments | The medical history tracking feature provided users with a convenient way to manage their health records. |

### Doctor Browsing and Filtering

Table 55: TC-013

|  |  |
| --- | --- |
| **Test Case ID** | **TC013** |
| Test Case Name | Doctor Browsing and Filtering - Search Functionality |
| Test Objective | To verify that users can browse healthcare professionals and apply filters based on timing and rating. |
| Test Inputs | Search criteria such as timing and rating filters. |
| Expected Result | Users should be able to find healthcare professionals based on specified criteria and view detailed profiles. |
| Test Steps | 1. Access the directory of healthcare professionals. 2. Apply filters based on timing and rating. 3. View detailed profiles of filtered professionals. |
| Actual Result | Users were able to browse healthcare professionals and apply filters effectively, resulting in the display of relevant profiles matching the specified criteria. |
| Pass/Fail |  |
| Notes/Comments | The search functionality provided users with flexibility and options to find suitable healthcare professionals. |

### Appointment Scheduling

Table 56: TC-014

|  |  |
| --- | --- |
| **Test Case ID** | **TC014** |
| Test Case Name | Appointment Scheduling - Booking Process |
| Test Objective | To verify that users can schedule appointments with healthcare professionals seamlessly. |
| Test Inputs | Preferred appointment slots. |
| Expected Result | Users should be able to view professional availability, select preferred slots, and confirm appointments without issues. |
| Test Steps | 1. Select a healthcare professional. 2. View availability. 3. Choose a preferred appointment slot. 4. Confirm the appointment. |
| Actual Result | Users were able to schedule appointments with healthcare professionals seamlessly, with the system handling scheduling conflicts effectively. |
| Pass/Fail |  |
| Notes/Comments | The appointment scheduling process was straightforward and user-friendly, enhancing the overall user experience. |

### Doctor Profile Management

Table 57: TC-015

|  |  |
| --- | --- |
| **Test Case ID** | **TC015** |
| Test Case Name | Doctor Profile Management - Information Update |
| Test Objective | To verify that doctors can update their profile information accurately. |
| Test Inputs | Updated personal info, qualifications, specialties, etc. |
| Expected Result | Doctors should be able to modify and save changes to their profile information without errors. |
| Test Steps | 1. Access the profile management section. 2. Update personal information, qualifications, etc. 3. Save changes to the profile. |
| Actual Result | Doctors were able to update their profile information accurately, and the system saved changes without errors. |
| Pass/Fail |  |
| Notes/Comments | The profile management feature provided doctors with the necessary tools to maintain accurate and up-to-date profiles. |

### Appointment Management

Table 58: TC-016

|  |  |
| --- | --- |
| **Test Case ID** | **TC016** |
| Test Case Name | Appointment Management - Doctor Side |
| Test Objective | To verify that doctors can manage their appointment schedule effectively. |
| Test Inputs | Updating availability, managing existing appointments. |
| Expected Result | Doctors should be able to set availability, add/edit/cancel appointments, and handle conflicts seamlessly. |
| Test Steps | 1. Access the appointment management section. 2. Set availability or manage existing appointments. 3. Ensure the appointment schedule updates correctly. |
| Actual Result | Doctors were able to manage their appointment schedule effectively, with the system handling conflicts and updates accurately. |
| Pass/Fail |  |
| Notes/Comments | The appointment management feature provided doctors with the necessary tools to maintain their schedule and ensure smooth patient interactions. |

### Doctor Patient Interaction

Table 59: TC-017

|  |  |
| --- | --- |
| **Test Case ID** | **TC017** |
| Test Case Name | Doctor Patient Interaction - Messaging |
| Test Objective | To verify that patients can communicate with doctors through the messaging system. |
| Test Inputs | Patient inquiries, doctor responses. |
| Expected Result | Patients should be able to send messages to doctors and receive prompt responses. |
| Test Steps | 1. Patient sends a message to the doctor through the messaging interface. 2. Doctor receives and responds to the message. |
| Actual Result | Patients were able to communicate with doctors through the messaging system, and doctors responded promptly. |
| Pass/Fail |  |
| Notes/Comments | The messaging system facilitated efficient communication between doctors and patients, enhancing the overall patient experience. |

### Review and Rating System

Table 60: TC-018

|  |  |
| --- | --- |
| **Test Case ID** | **TC018** |
| Test Case Name | Review and Rating System - Feedback Submission |
| Test Objective | To verify that patients can submit feedback and ratings for healthcare professionals. |
| Test Inputs | Feedback and ratings for professional services. |
| Expected Result | Patients should be able to provide feedback and ratings accurately. |
| Test Steps | 1. Patient completes an appointment with a healthcare professional. 2. Patient provides feedback and ratings for the professional. |
| Actual Result | Patients were able to submit feedback and ratings for healthcare professionals accurately. |
| Pass/Fail | Pass |
| Notes/Comments | The review and rating system allowed patients to share their experiences and contribute to the evaluation of healthcare professionals. |

### Admin Panel

Table 61: TC-019

|  |  |
| --- | --- |
| **Test Case ID** | **TC019** |
| Test Case Name | Admin Panel - Activity Monitoring |
| Test Objective | To verify that administrators can monitor and manage doctor activities effectively through the admin panel. |
| Test Inputs | Viewing activity logs, performance metrics. |
| Expected Result | Administrators should be able to access comprehensive activity logs and performance metrics for doctors, enabling effective supervision and management. |
| Test Steps | 1. Access the admin panel. 2. View activity logs and performance metrics for doctors. 3. Take necessary actions based on the information. |
| Actual Result | Administrators were able to monitor and manage doctor activities effectively through the admin panel, with access to comprehensive logs and metrics. |
| Pass/Fail |  |
| Notes/Comments | The admin panel provided administrators with valuable insights into doctor activities, facilitating better oversight and decision-making. |

## Test Types

### Unit Testing

For unit testing, you would typically test individual units or components of your system in isolation. Since your project involves various functionalities and components, each unit test would focus on testing a specific function, class, or module to ensure it behaves as expected.

### Integration Testing

Integration testing involves testing the interaction between different components or modules to ensure they work together as expected. In your project, you would integrate various functionalities such as user authentication, prediction processing, appointment scheduling, etc., and test how they interact with each other.

### Acceptance Testing

Acceptance testing involves testing the system from an end-user's perspective to ensure it meets the specified requirements and works as expected in real-world scenarios. This testing phase usually involves stakeholders or end-users who validate whether the system meets their needs and expectations.

Chapter 5 Conclusion

# Conclusion

## Problems Faced and Lessons Learned

### Quality of data

The data set consists of anomalies we clean up the data and remove all anomalies we create a data pipeline which will automatically increase the quality of data.

### Class Imbalance

The data set has unequal distribution of samples across superclass and sub classes. It is leading us to bias prediction we will use data augmentation technique to balance the class data.

### Overfitting

ECG signals are complex and high dimensional. We use different models in LSTM it goes overfit due to biased data we make it correct by changing the split and train model to RNN.

### Interpreting Complex patterns

ECG signals have complex patterns and its high dimensional data it was difficult for model to interpret correctly to make predictions correct we reduced a feature size.

## Project Summary

CardioGraph Pro indicates a huge improvement in how heart disease is predicted and diagnosed by taking advantages of machine learning advanced data processing techniques and image processing techniques. An electrocardiogram (ECG) signal and demographic data will be analyzed critically as part of the project, to assist in the detection and diagnose of heart disease in its early stages. Our approach has been systematic throughout the project, starting with the collection of data from sources like the PTB-XL ECG dataset and PhysioNet WFDB database.

We collect the data and remove anomalies from data and then prepare it for machine learning model training. Furthermore, to traditional ECG signal processing, our work comprises image-based ECG report examination and signals-based report examination, refining the range of input sources and enhancing prediction potentials. Extending a confident accuracy score is an important success measure for CardioGraph Pro, assuring that predictions are reliable and fruitful for both patients and medical professionals such as doctors and cardiologists. Through the integration of demographic information with ECG signals, we hope to outperform current prediction techniques and provide a more complete early disease detection solution. We use different models to achieve maximum threshold we tarin this data to LSTM, CNN and RNN machine learning models. Team members are assigned specific tasks to ensure effective project management and timely achievement of objectives.

The Gantt chart, which represents the suggested timeframe, offers a project execution roadmap that covers tasks ranging from planning and research to implementation and documentation. The strategy aims to diagnose early disease detection by providing a user-friendly platform that will be helpful for patients and healthcare professionals such as doctors. We are adding a medical chatbot which will improve the user experience even more by providing trustworthy medical advice and real-time result interpretation. In Addition, users have the option to upload images of their ECG measures, which allows for the smoothing of deployed machine learning models traditional image processing methods. CardioGraph Pro is versatile in this function, which fulfils the demands and preferences of a wide range of users. As a team, we made CardioGraph Pro, a new tool or platform that can predict heart problems and give advice on the predicted disease. It is good because it can help patients get better, save money on medical bills, and maybe even save lives and may help doctors to diagnose the patient in less time. We worked hard to make it and we think it will really help in the fight against heart disease.

## Future Work

### Addressing Class Imbalance

While data augmentation techniques have been employed to address class imbalance, further research could explore additional methods such as synthetic data generation or ensemble learning techniques to better balance the distribution of samples across superclass and subclasses, thus reducing bias in predictions.

### Mitigating Overfitting

While switching from LSTM to RNN helped mitigate overfitting, ongoing efforts to prevent overfitting could involve exploring regularization techniques, cross-validation strategies, or leveraging more diverse datasets to train the models effectively.

### Expanding Model Selection and Evaluation

Experimenting with a wider range of machine learning models beyond LSTM, CNN, and RNN could provide insights into which models perform best for specific prediction tasks. Additionally, conducting rigorous evaluation and comparison of different models on diverse datasets can further refine the model selection process.

### Scaling Computational Resources

As the demand for computational resources grows with the expansion of the platform and user base, future work could involve scaling up computational resources, either through cloud-based solutions or optimized hardware infrastructure, to ensure efficient and timely processing of data and predictions.

### Collaboration with Healthcare Institutions

Collaborating with healthcare institutions and clinicians for validation and integration of CardioGraph Pro into clinical workflows could facilitate its adoption and real-world impact, ensuring that it meets the needs and standards of the healthcare industry.

Chapter 06 References

# References

[1] U. R. Acharya, H. Fujita, O. S. Lih, Y. Hagiwara, J. H. Tan, and M. Adam, “Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network,” *Information sciences*, vol. 405, pp. 81–90, 2017.

[2] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng, “Cardiologist-level arrhythmia detection with convolutional neural networks,” *arXiv preprint arXiv:1707.01836*, 2017.

[3] O. Sayadi, M. B. Shamsollahi, and G. D. Clifford, “Robust detection of premature ventricular contractions using a wave-based Bayesian framework,” *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 2, pp. 353–362, 2009.

[4] A. L. Goldberger *et al.*, “PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals,” *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[5] M. G. Tsipouras, D. I. Fotiadis, and D. Sideris, “An arrhythmia classification system based on the RR-interval signal,” *Artificial intelligence in medicine*, vol. 33, no. 3, pp. 237–250, 2005.

[6] J. P. Martínez, R. Almeida, S. Olmos, A. P. Rocha, and P. Laguna, “A wavelet-based ECG delineator: evaluation on standard databases,” *IEEE Transactions on biomedical engineering*, vol. 51, no. 4, pp. 570–581, 2004.

[7] O. Faust, Y. Hagiwara, T. J. Hong, O. S. Lih, and U. R. Acharya, “Deep learning for healthcare applications based on physiological signals: A review,” *Computer methods and programs in biomedicine*, vol. 161, pp. 1–13, 2018.

[8] A. Y. Hannun *et al.*, “Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network,” *Nature medicine*, vol. 25, no. 1, pp. 65–69, 2019.

[9] Z. I. Attia *et al.*, “Screening for cardiac contractile dysfunction using an artificial intelligence–enabled electrocardiogram,” *Nature medicine*, vol. 25, no. 1, pp. 70–74, 2019.

[10] A. H. Ribeiro *et al.*, “Automatic diagnosis of the 12-lead ECG using a deep neural network,” *Nature communications*, vol. 11, no. 1, p. 1760, 2020.

[11] S. Śmigiel, K. Pa\lczyński, and D. Ledziński, “ECG signal classification using deep learning techniques based on the PTB-XL dataset,” *Entropy*, vol. 23, no. 9, p. 1121, 2021.

[12] A. L. Golande and T. Pavankumar, “Optical electrocardiogram based heart disease prediction using hybrid deep learning,” *Journal of Big Data*, vol. 10, no. 1, p. 139, 2023.

[13] J. Zhang, B. Li, K. Xiang, and X. Shi, “Method of diagnosing heart disease based on deep learning ECG signal,” *arXiv preprint arXiv:1907.01514*, 2019.

[14] L. Mhamdi, O. Dammak, F. Cottin, and I. B. Dhaou, “Artificial intelligence for cardiac diseases diagnosis and prediction using ECG images on embedded systems,” *Biomedicines*, vol. 10, no. 8, p. 2013, 2022.

