Group 23

ECG Analysis Dashboard
Software Architecture Document

Version <1.0>

Prepared By Group 23

Sajeenthiran P – 210553J

Ranaweera HK - 210523T

Rathnayaka W.T - 210536K

Revision History

Date	Version	Description	Author
16/07/24	1.0	Initial version	Group 23

Table of Contents

Software Architecture Document

1. Introduction

1.1 Purpose

This document offers a comprehensive architectural overview of the system, utilizing various architectural perspectives to illustrate its diverse aspects. Its purpose is to capture and communicate the significant architectural choices made for the system.

In order to present a thorough architectural perspective of this system, this Software Architecture Document employs five distinct views following the 4+1 view model: Logical view, Process view, Deployment view, Implementation view, and Use Case view. Additionally, this document delineates the data views of the system. Each view model caters to specific concerns of various stakeholders involved in the system.

This document serves as a valuable reference for developers during system implementation. Developers can utilize the provided class diagrams, activity diagrams, and state diagrams for coding applications. The activity and state diagrams provided can also aid testers in their tasks.

Expected users of this document are given below

- System Designer (to get knowledge regarding application structure and architecture)
- System Developer (to know how they need to implement the system. must understand Class diagrams, activity diagrams, sequence diagrams, and state diagrams)
  Client (to know the components of the systems and see the knowing different aspects of the system)
- Database Designer: (to know regarding data models and application structure)
  System Tester: (to know how they can do the testing process. They need to study activity diagrams and state diagrams.)

1.2 Scope

The Software Architecture Document applies to the ECG Analysis Dashboard, influencing its design, development, and deployment processes

1.3 Definitions, Acronyms, and Abbreviations

- **ECG**: Electrocardiogram
- **HIPAA**: Health Insurance Portability and Accountability Act
- **GDPR**: General Data Protection Regulation
- **UI**: User Interface
- **API**: Application Programming Interface

1.4 References

- IEEE Software Requirements Specification Template

1.5 Overview

The document is organized into sections detailing various architectural views including the use-case view, logical view, process view, deployment view, implementation view, data view, and quality attributes

2. Architectural Representation

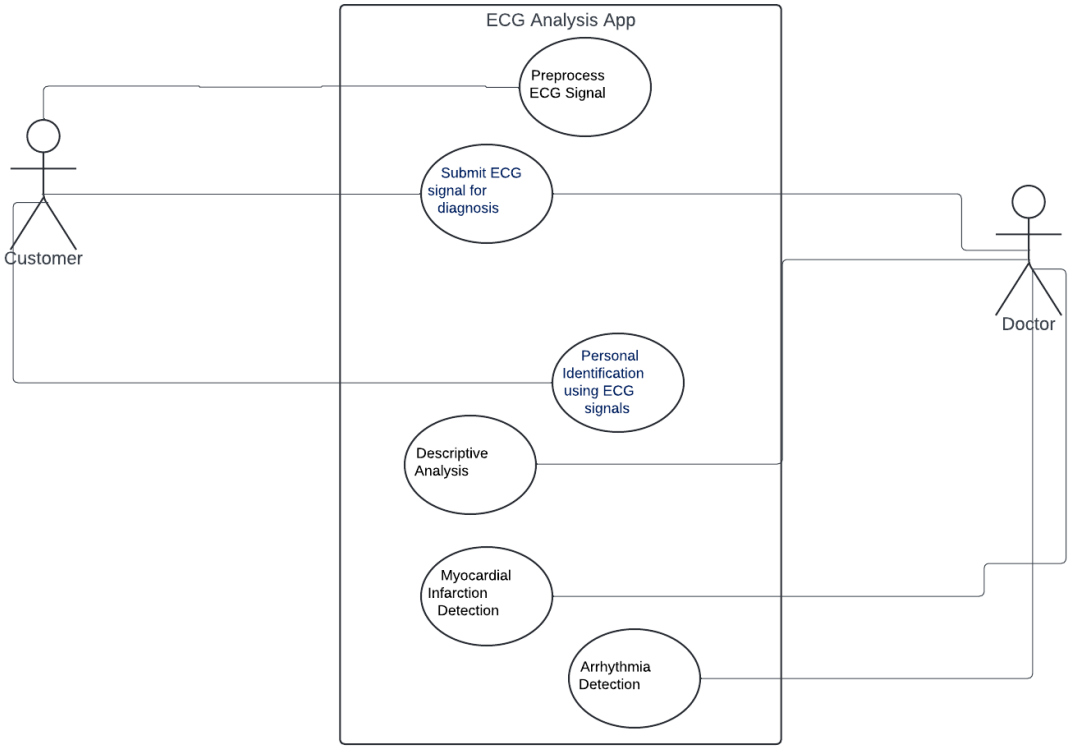
The architecture of the ECG Analysis Dashboard is represented using multiple views:

- **Use-Case View**: Illustrates significant use cases.
- **Logical View**: Describes the system's structure in terms of packages and classes.
- **Process View**: Details the system’s process structure.
- **Deployment View**: Shows the physical deployment of the system.
- **Implementation View**: Outlines the system's layers and components.
- **Data View**: Provides a perspective on data storage and management (optional).

3. Architectural Goals and Constraints

- **Goals**:
  - Provide reliable and accurate ECG analysis.
  - Ensure user data privacy and security.
  - Achieve high performance and responsiveness.
- **Constraints**:
  - Must comply with healthcare regulations (e.g., HIPAA, GDPR).
  - Integration with existing hospital systems.
  - Use of specific machine learning frameworks (e.g., TensorFlow).

4. Use-Case View



4.1 Use-Case Realizations

4.1.1 Submit ECG Signal for diagnosis

Use case name	Submit ECG signal for diagnosis
Actor	Doctor,Customer
Description	The user submits ECG signal through the app
preconditions	User must have valid ECG signal
Main flow	- User selects the option to submit ECG. - The app prompts the window to submit the ECG data file. - The app confirms successful submission and stores the signal for further processing.
Successful end/post condition	ECG signal is captured and stored in the app.

Fail end/post condition	If the app fails to validate the signal, the app displays an error and prompts the user to retry.
Extensions	N/A

4.1.2 Preprocess ECG Signal

Use case name	Preprocess ECG Signal
Actor	Doctor
Description	The captured ECG signal undergoes preprocessing to remove noise and artifacts.
preconditions	ECG signal must be captured and stored.
Main flow	1. The preprocessing module retrieves the stored ECG signal. 2. Noise and artifacts are identified and removed from the signal. 3. The preprocessed signal is stored for further analysis.
Successful end/post condition	The ECG signal is cleaned and ready for analysis.
Fail end/post condition	If preprocessing fails, the system logs the error and notifies the user.
Extensions	N/A

4.1.3 Descriptive Analysis

Use case name	Descriptive Analysis
Actor	Doctor
Description	The app performs descriptive analysis on the preprocessed ECG signal to extract key features and metrics.
preconditions	ECG signal must be preprocessed.
Main flow	- The analysis module retrieves the preprocessed ECG signal. - Key features (e.g., heart rate, QRS duration) are extracted. - Descriptive statistics are calculated.

	- The results are stored and optionally displayed to the user.
Successful end/post condition	Descriptive statistics and key features are available.
Fail end/post condition	If analysis fails, the system logs the error and notifies the user.
Extensions	N/A

4.1.4 Myocardial Infarction Detection

Use case name	Myocardial Infarction Detection
Actor	Doctor
Description	The app analyzes the preprocessed ECG signal to detect signs of myocardial infarction (heart attack).
preconditions	ECG signal must be preprocessed.
Main flow	- The MI detection module retrieves the preprocessed ECG signal. - The module applies algorithms to detect patterns indicative of myocardial infarction. - If detected, the app alerts the user and recommends medical consultation. - The detection result is stored.
Successful end/post condition	Detection result (positive or negative) is available.
Fail end/post condition	If detection fails, the system logs the error and notifies the user.
Extensions	N/A

4.1.5 Arrhythmia Detection

Use case name	Arrhythmia Detection
Actor	Doctor
Description	The app analyzes the preprocessed ECG signal to detect signs of arrhythmia (irregular heartbeat).
preconditions	ECG signal must be preprocessed.

Main flow	- The arrhythmia detection module retrieves the preprocessed ECG signal. - The module applies algorithms to detect patterns indicative of arrhythmias. - If detected, the app alerts the user and provides information on the type of arrhythmia. - The detection result is stored.
Successful end/post condition	Detection result (positive or negative) is available.
Fail end/post condition	If detection fails, the system logs the error and notifies the user.
Extensions	N/A

4.1.6 Personal Identification using ECG signals

Use case name	Personal Identification using ECG signals
Actor	Customer
Description	App Uses biometric features of the ECG signals to determine if two ECG readings belong to the same person.
preconditions	User must have submitted two ECG signals he/she wants to compare
Main flow	- The personal identification module retrieves the preprocessed two ECG signal. - The module applies algorithms to detect the biometric features between two given ECGs. - If the biometric features are similar above a certain threshold, the app alerts the user and provides information on the type of similairty - The result is stored
Successful end/post condition	Identification result (similar or different) is available.
Fail end/post condition	If detection fails, the system logs the error and notifies the user.

Extensions	N/A

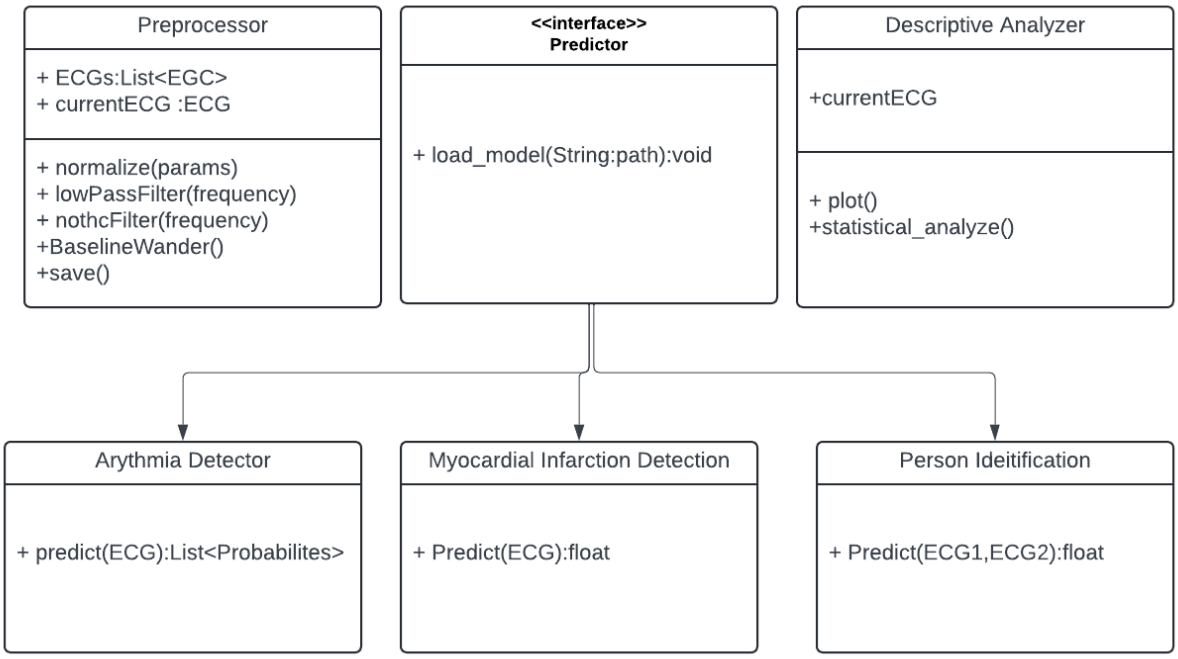
5. Logical View

5.1 Overview

The system is decomposed into several packages, each responsible for a specific functionality:

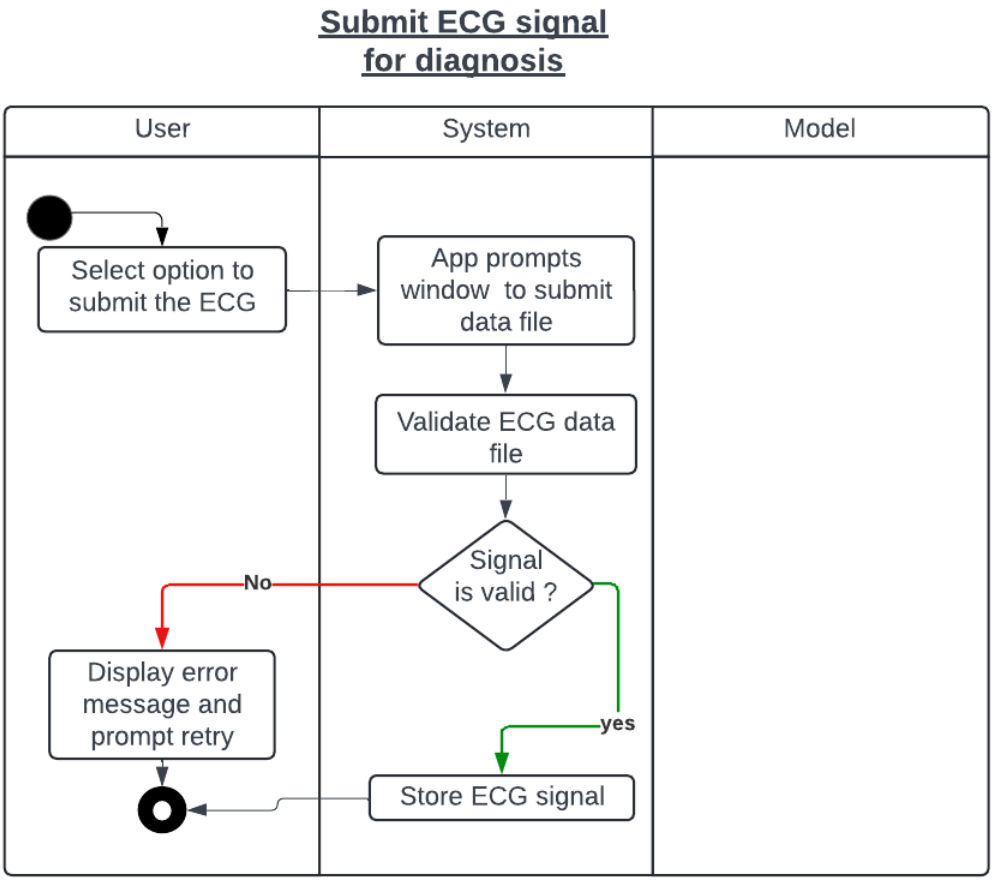
- **Preprocessing Package**: Handles data cleaning and preparation.
- **Analysis Package**: Performs statistical and graphical analysis.
- **Identification Package**: Manages person identification tasks.
- **Detection Package**: Handles arrhythmia and myocardial disease detection.
- **UI Package**: Manages the user interface components.

5.2 Architecturally Significant Design Packages

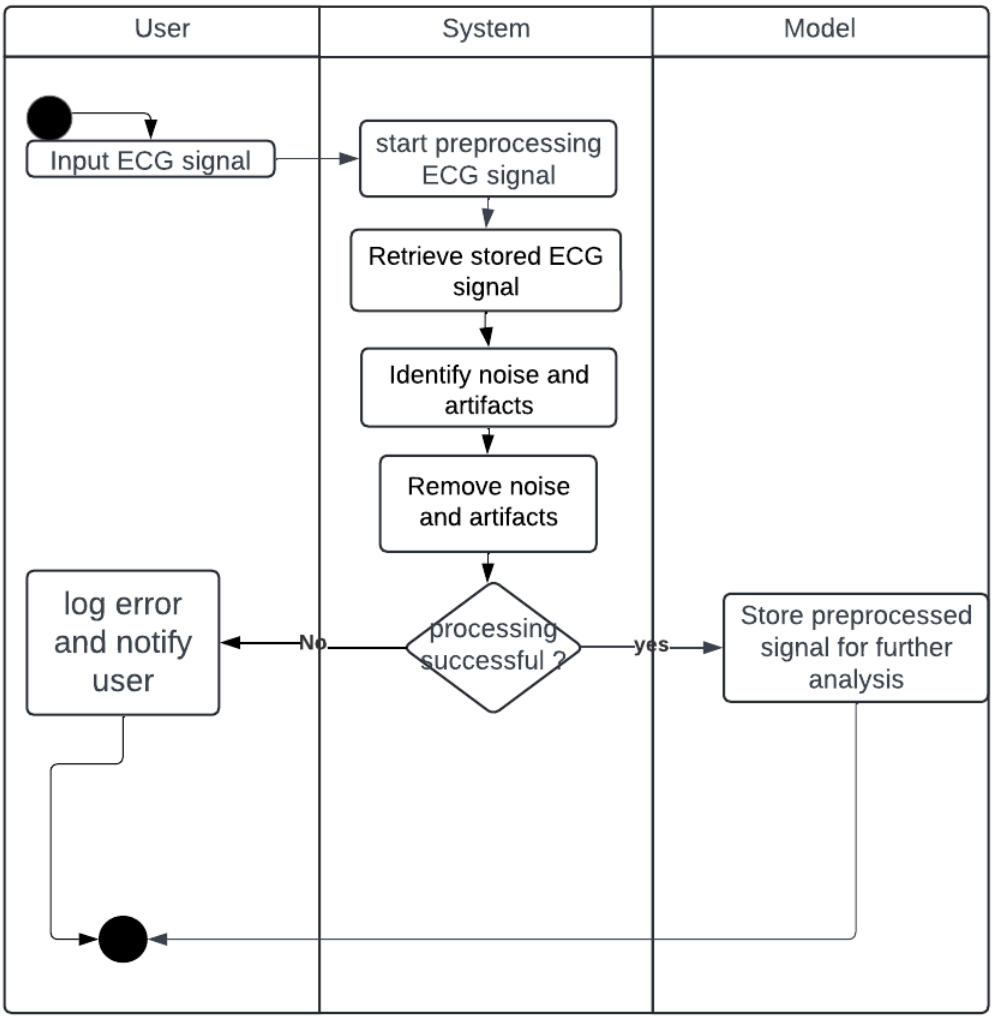


6. Process View

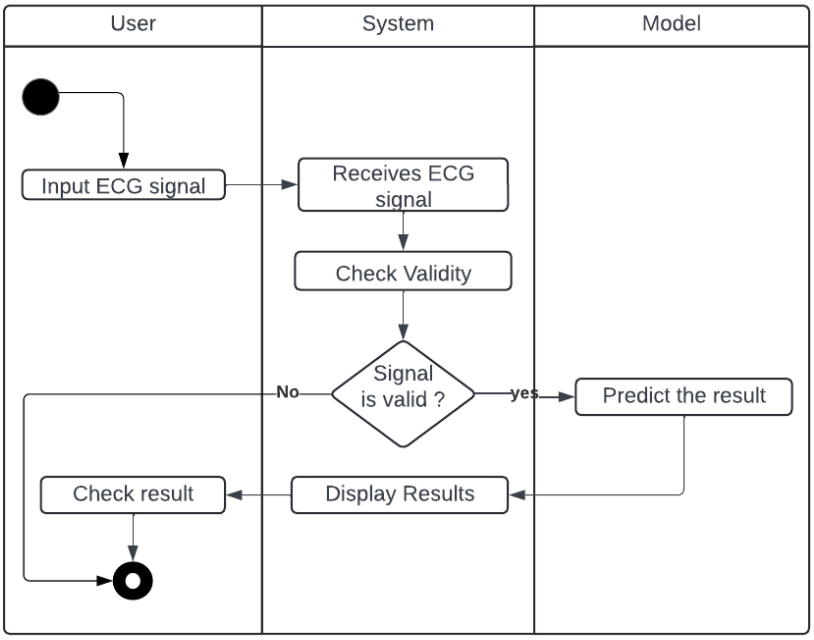
6.1 Activity Diagrams



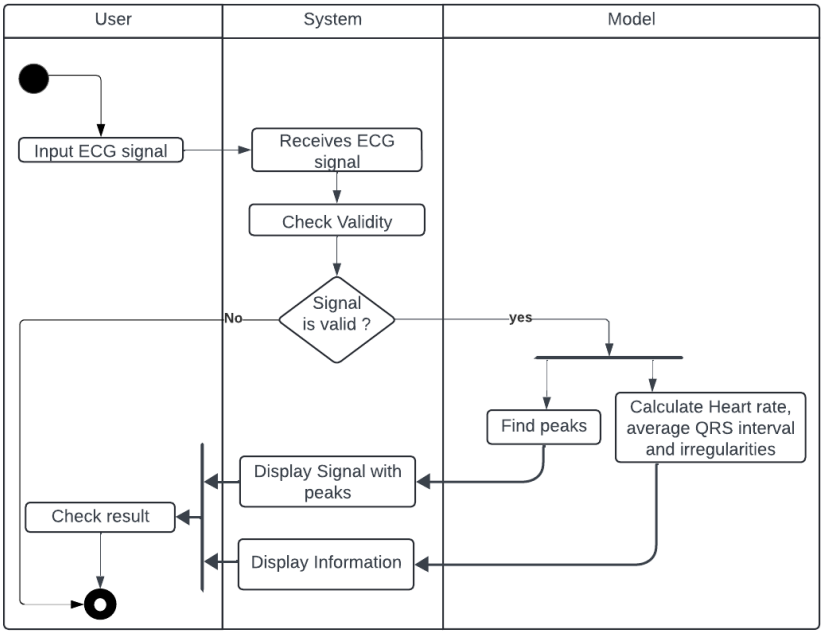
Activity Diagram of Preprocess ECG Signal



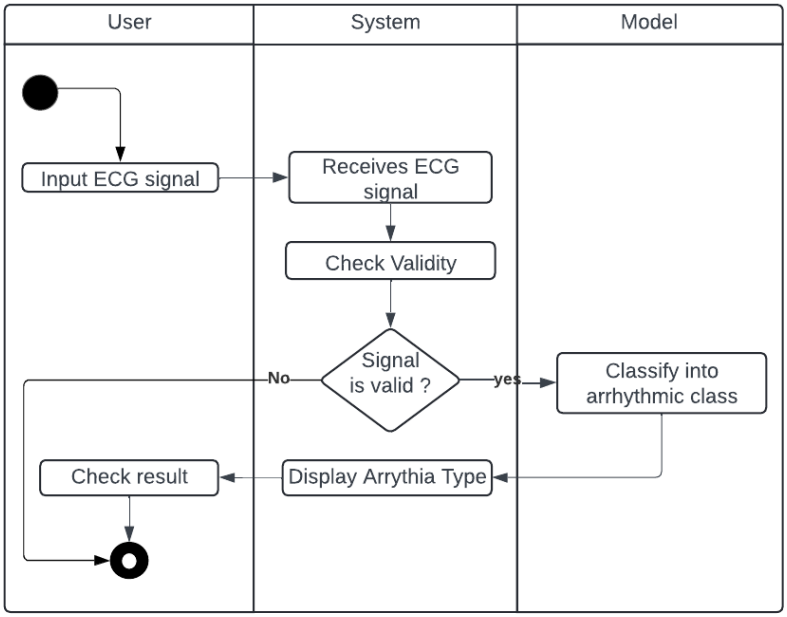
Activity Diagram of MI detection process



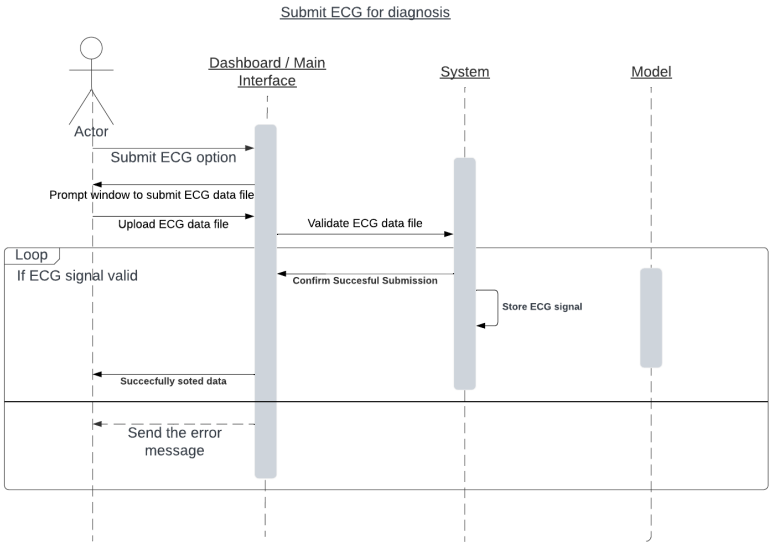
Activity Diagram of Descriptive Analysis

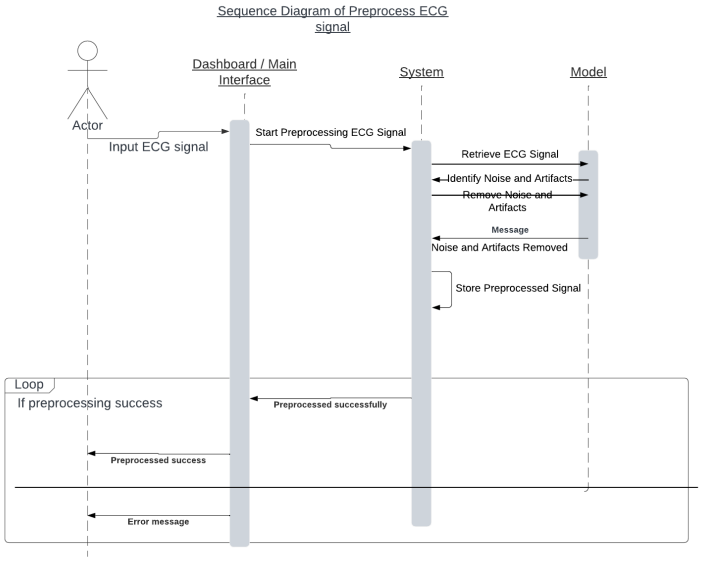


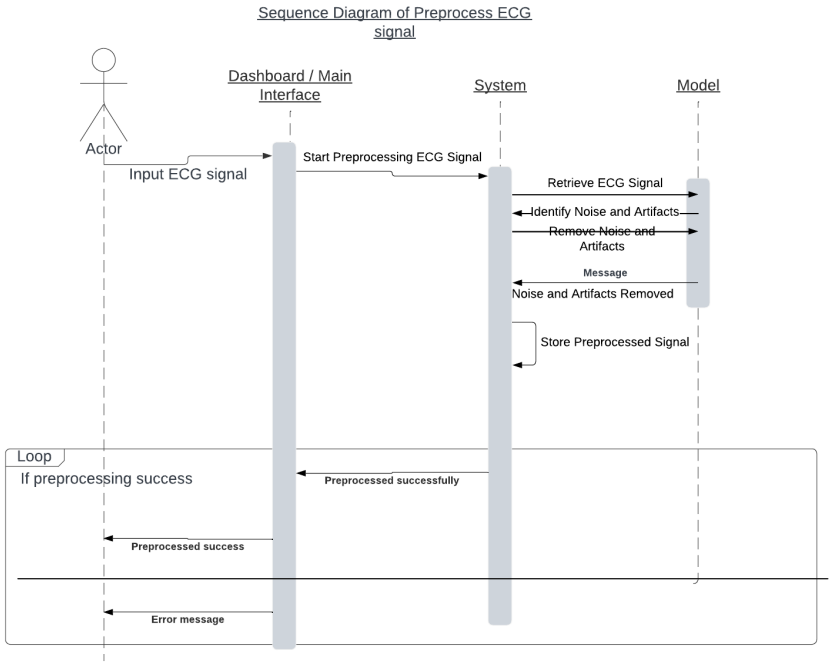
Activity Diagram of Arrhythmia detection process

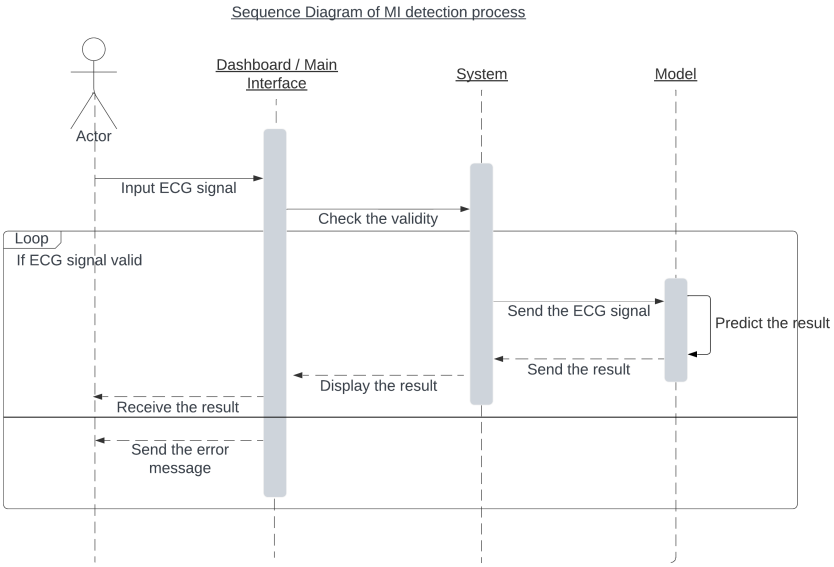


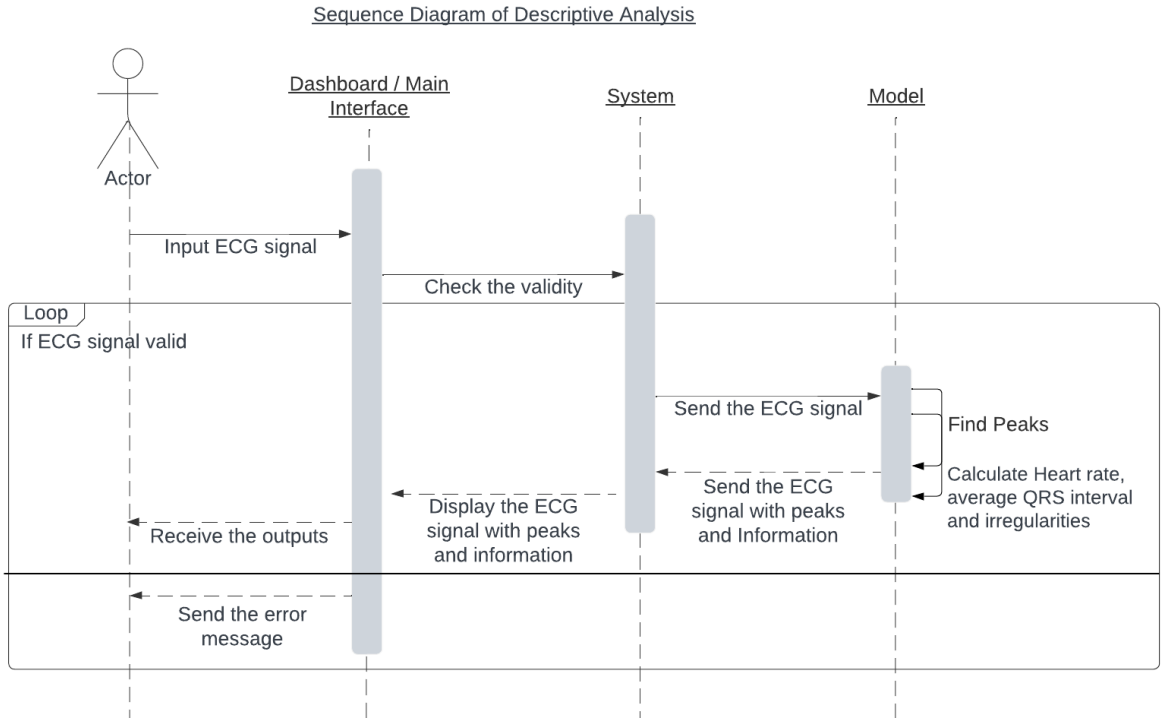
6.2 Sequence Diagrams

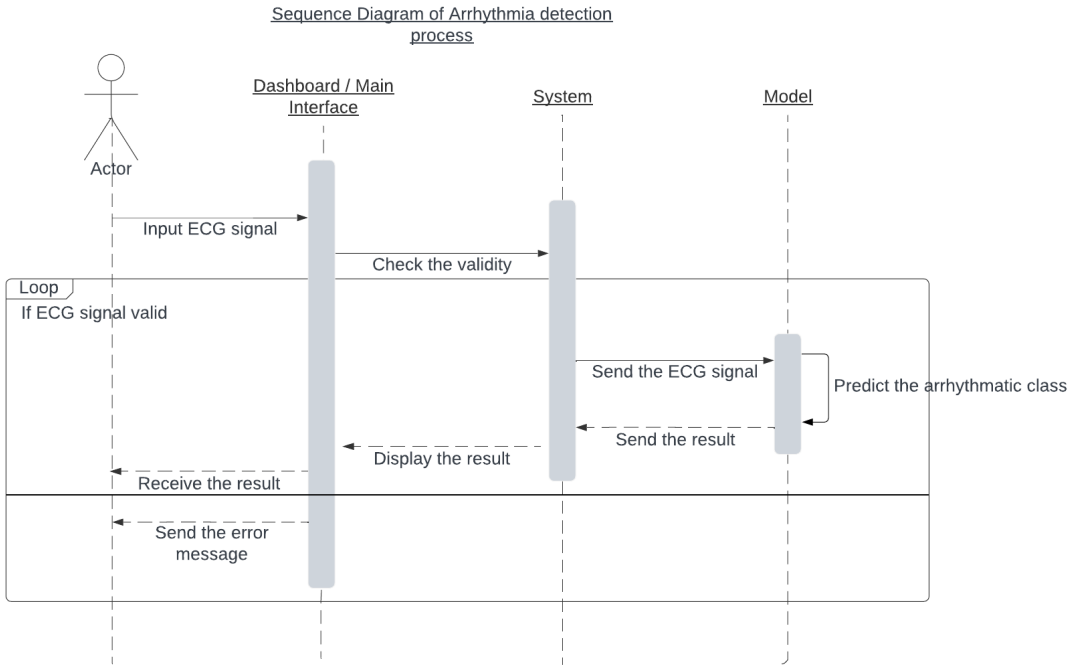




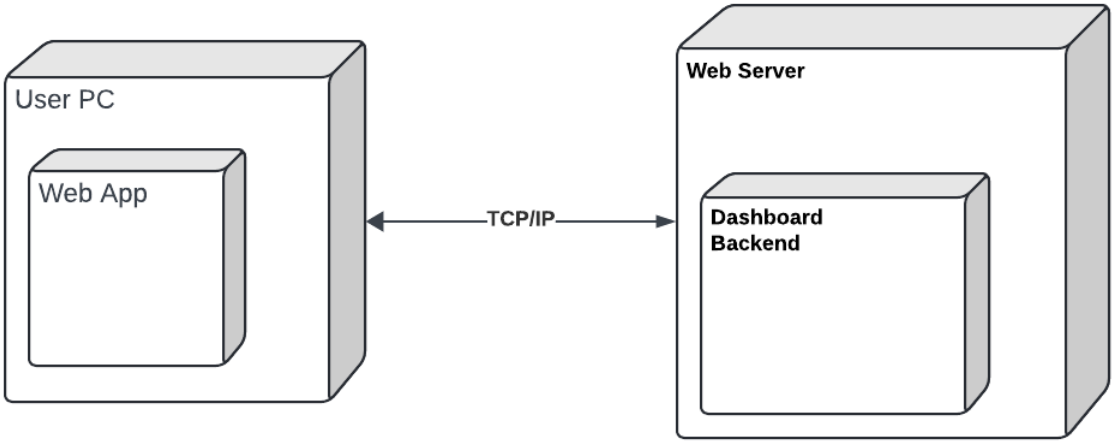

Sequence Diagram of Preprocess ECG signal
Actor
Dashboard / Main Interface
System
Model
Input ECG signal
Start Preprocessing ECG Signal
Retrieve ECG Signal
Identify Noise and Artifacts
Remove Noise and Artifacts
Message
Noise and Artifacts Removed
Store Preprocessed Signal
Loop
If preprocessing success
Preprocessed successfully
Preprocessed success
Error message


Sequence Diagram of MI detection process
Actor
Dashboard / Main Interface
System
Model
Input ECG signal
Check the validity
Loop
If ECG signal valid
Send the ECG signal
Predict the result
Send the result
Display the result
Receive the result
Send the error message


Sequence Diagram of Descriptive Analysis
Actor
Dashboard / Main Interface
System
Model
Input ECG signal
Check the validity
Loop
If ECG signal valid
Send the ECG signal
Find Peaks
Calculate Heart rate, average QRS interval and irregularities
Send the ECG signal with peaks and Information
Display the ECG signal with peaks and information
Receive the outputs
Send the error message


Sequence Diagram of Arrhythmia detection process
Actor
Dashboard / Main Interface
System
Model
Input ECG signal
Check the validity
Loop
If ECG signal valid
Send the ECG signal
Predict the arrhythmatic class
Send the result
Display the result
Receive the result
Send the error message

6. Deployment View



7. Implementation View

7.1 Overview

UI Interface Layer

This layer is the front-end part of the system that users interact with. It consists of five sections:

Preprocessing: Handles the initial steps of cleaning and normalizing the raw ECG data.

Descriptive Analysis: Provides detailed analysis and visualization of the ECG signals, offering statistical summaries and visual representations.

Myocardial Infarction Detection: Displays results from the myocardial infarction detection model, indicating the presence of a heart attack.

Arrhythmia Detection: Shows the output of the arrhythmia detection model, identifying any irregular heart rhythms.

Person Identification: Uses biometric features of the ECG signals to determine if two ECG readings belong to the same person.

Models Layer

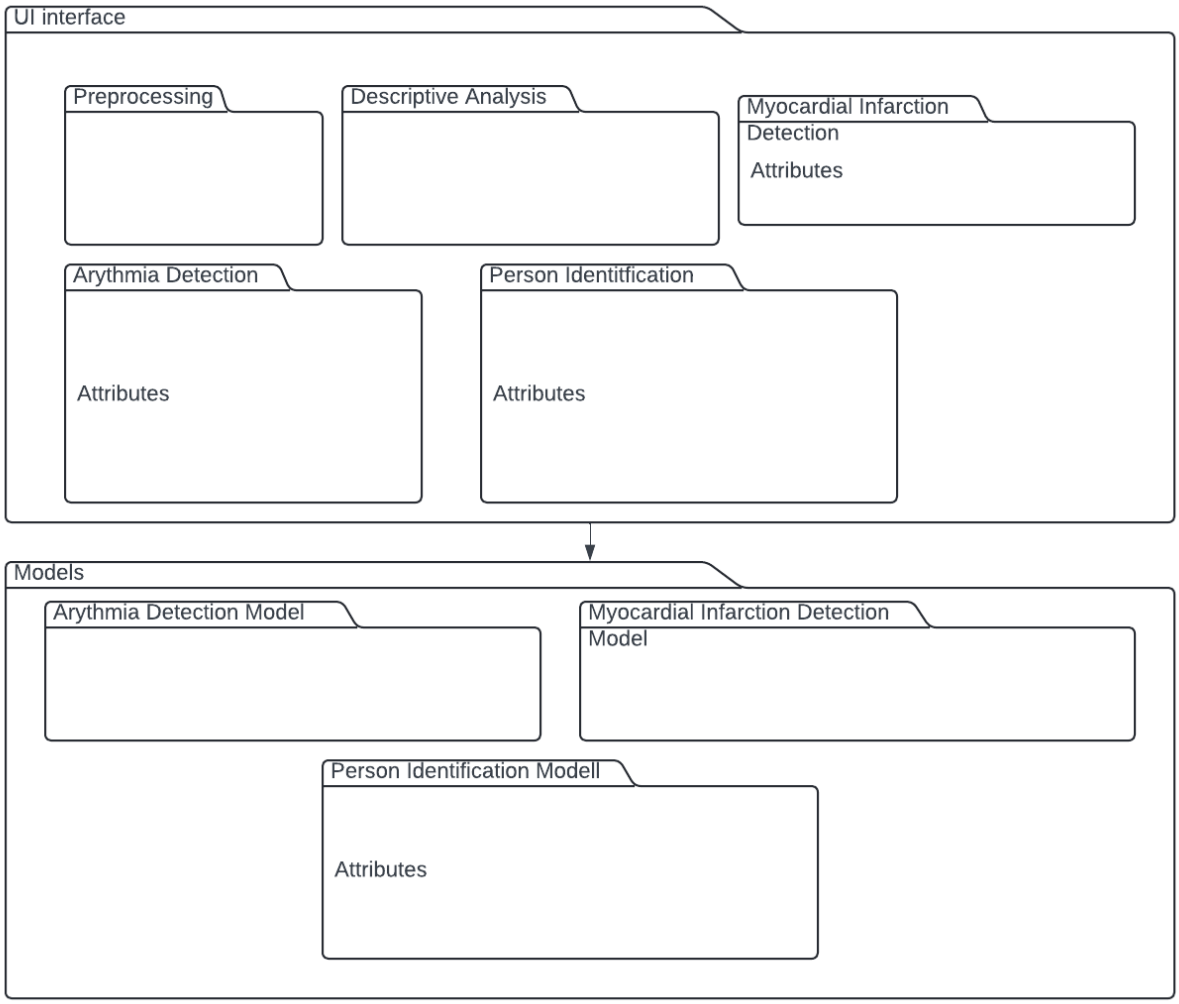
This layer consists of the backend components where the actual data processing and analysis occur:

Arrhythmia Detection Model: Analyzes ECG signals to detect abnormal heart rhythms.

Myocardial Infarction Detection Model: Identifies signs of myocardial infarction in the ECG data.

Person Identification Model: Uses machine learning techniques to match ECG signals to individuals for identification purposes.

7.2 Layers



8. Size and Performance

- **Size**: The system must be capable of handling up to 1 TB of ECG data.
- **Performance**: The system should process an ECG signal within 2 seconds and support up to 100 concurrent users

9. Quality

- **Extensibility**: Modular design to allow easy addition of new features.
- **Reliability**: Ensure 99.9% uptime.
- **Portability**: Compatible with Windows, macOS, and Linux.
- **Security**: Compliant with HIPAA and GDPR regulations.

12. References

[1] IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

[2] “Streamlit.” Internet: https://docs.streamlit.io/[July. 16, 2024]

-