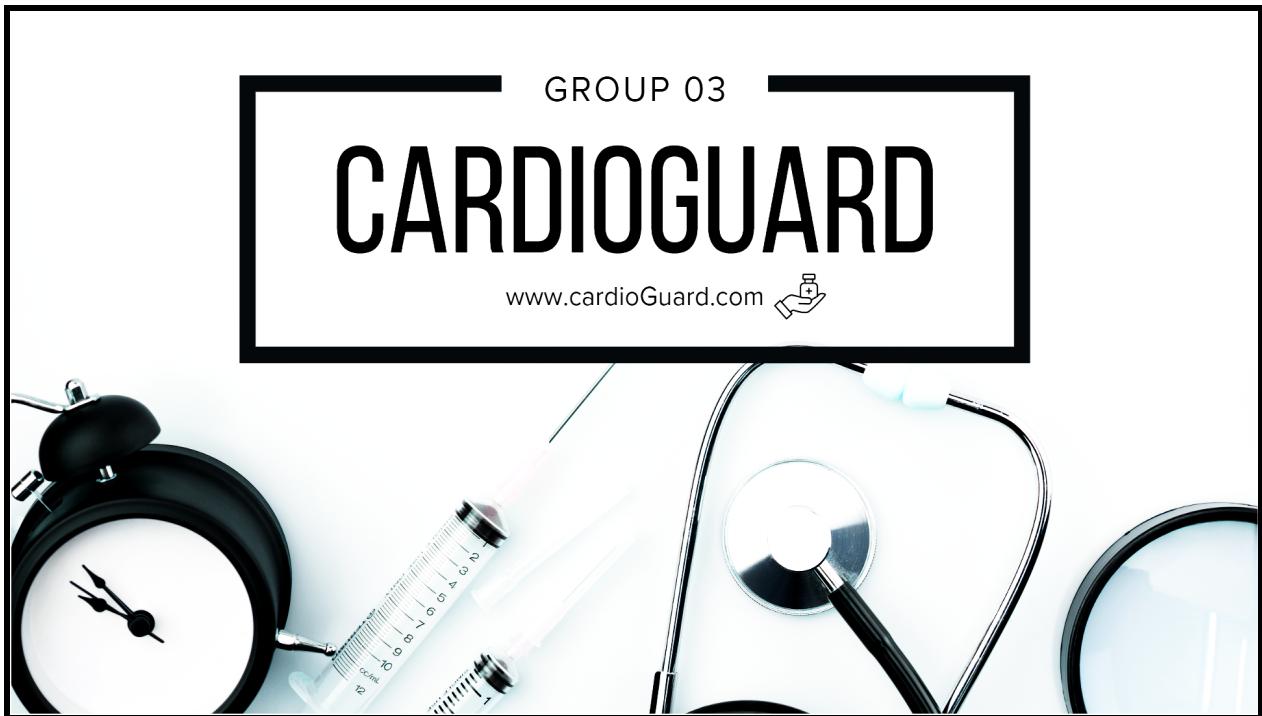


Group 03 - Project Report

CardioGuard

Nov 28, 2024



CO227 Second Year Project (E20)

**E/20/084 Dissanayake P.D.
E/20/254 Methpura S.K.P.
E/20/262 Nanayakkara A.T.L.
E/20/266 Nilupul D.R.P
E/20/397 Thilakasiri P.D.**

TABLE OF CONTENTS

1. Introduction	
a. What is CardioGuard	4
b. Requirements and Users.....	5
c. Objectives.....	6
d. Data collection.....	7
e. Advisors and other contributors.....	13
f. Class Diagram.....	14
g. Sequence Diagrams.....	16, 17
h. UseCase Diagram	18
i. Entity relations diagram.....	19
2. Functional Requirements.....	20
3. Non Functional requirements.....	21
4. Use Case Descriptions.....	22
5. Technology and Implementations	
a. Technology	
i. Java and Spring Boot for backend.....	25
ii. React and Tailwind CSS for front end.....	26
iii. Python for Machine Learning.....	27
6. User Manual & Front end implementation	
i. Content	
1. Home.....	28
2. About us.....	29
3. Contact Us.....	29
4. Services.....	30
5. Users.....	31
ii. DashBoards	
1. Admin DashBoard.....	32
2. Doctor DashBoard.....	33
3. Patient DashBoard.....	37
7. BackEnd development and Machine learning	

i.	Implementation of components	
1.	Controllers and Domains.....	41,42
2.	Entity and Mapper.....	43,44
3.	Exception and Repository.....	45
4.	Security and Services.....	46
ii.	Machine Learning.....	48
8.	Functionality of the system	
i.	Admin Functionalities.....	51
ii.	Doctor Functionalities.....	56
iii.	Patient Functionalities.....	60
9.	Testing and Error checking.....	63
10.	Project Management and version control.....	69
11.	Code Quality and Security.....	71
12.	Feedback.....	77
13.	Challenges.....	78
14.	Work division and individual contribution.....	79
15.	Timeline.....	68
16.	Declaration	83

Introduction

What is CardioGuard?



Figure 1 : CardioGuard logo

CardioGuard

CardioGuard is a cardiac patient database management system embedded with machine learning. System is designed to provide the doctors and patients with an overview of their current medical state. System provides doctors and administration access to patients for easier medical attention. Patients have the ability to see some of their data from their initial admission into a medical sector.

Requirements and Users

With the help of statistics from the World health organization, studies suggest that 64% of the deaths per year are caused due to cardiac arrests also in general known as “Heart Failure”. The system provides a numerical aspect for the doctors and patients to better understand the medical standpoint of the individual for advisory and further continuance in medical attention and care avoiding critical and fatal health problems.

The system , in this form, is a faster and more accurate data provider for cases in ambiguity and non responsive patients in critical situations. Patient information management,diagnostic and treatment,data analysis using a prediction system,prescription management,patient portal accessibility,Administrative features and data backup and recovery are the requirements for this system. Requirements were gained through contact with Doctors and medical students and professors of the university and an Engineer (listed in **Advisors and other contributors**).

CardioGuard is dependent on patient data, supervisory of senior doctors in cardiac diseases sector and further on doctors' correct inputs of data.

Users of this system are identified as:

- ❖ Patients - Cardiac Patients registered to the system
- ❖ Doctors - Doctors registered to the system
- ❖ Non patients - Non patients have the ability to explore the UI
- ❖ Administrator - Administrator of the system

OUR OBJECTIVES

- ❖ Create a platform for the patients and doctors to bridge healthcare and technology.
- ❖ Provide quick access for doctors to get patient data
- ❖ Provide patients a user interface to inspect some of their data
- ❖ Patient information management
- ❖ Easier diagnostic and treatment
- ❖ Data analysis using a prediction system
- ❖ Prescription management
- ❖ Patient portal accessibility
- ❖ Administrative features
- ❖ Data backup for safety



Data Collection and access

- ❖ Initial data will be collected through the admission into a medical center.
- ❖ A doctor or a nurse will be responsible for the data entry into the system
- ❖ There are different types of data collected in the system
 - Symptoms of the patient
 - Details of patient examinations
 - Clinical data
 - Prediction Data

Symptoms of the patient

Symptoms of patient	Description
Bilateral_lower_limb_swelling	Swelling in both legs, often due to fluid accumulation. It can be a symptom of heart failure, kidney disease, or other medical conditions affecting circulation and fluid balance.
Dyspnoea	Also known as shortness of breath. It is a common symptom of heart and lung conditions, including heart failure, where the heart is unable to pump blood efficiently.
Orthopnoea	Shortness of breath that occurs when lying flat, forcing the person to sleep propped up or in a sitting position. It is often associated with heart failure.
Paroxysmal_nocturnal_dyspnoea	Sudden episodes of severe shortness of breath at night, causing the person to wake up and often sit or stand up to relieve the symptoms. It is a sign of worsening heart failure.
Fatigue	A feeling of extreme tiredness or lack of energy that is not relieved by rest. It is a common symptom of heart disease, as the body may not be receiving enough oxygenated blood.

Details of patient examination

Symptoms of patient	Description
Tachycardia_at_rest	An abnormally fast heart rate (usually over 100 beats per minute) when the body is at rest. It can be a sign of various heart conditions, including heart failure or arrhythmias.
Hypotension	Abnormally low blood pressure, which can indicate poor blood flow to organs. It may occur in heart failure or as a side effect of medications.
Raised_jugular_venous_pressure	Increased pressure in the jugular veins, often visible as distension of the neck veins. It is a sign of elevated central venous pressure, commonly seen in heart failure.
Displaced_apex_beat	The apex beat of the heart (normally felt in the left fifth intercostal space) is felt in an abnormal position, indicating an enlarged heart, often due to heart failure.
Right_ventricular_heave	A palpable lifting sensation under the sternum due to the hypertrophy or dilation of the right ventricle, often associated with pulmonary hypertension or right-sided heart failure.
Gallop_rhythm_on_auscultation	An additional heart sound heard during auscultation, often indicative of heart failure. It can include an S3 (ventricular gallop) or S4 (atrial gallop) sound.
Murmurs_associated_with_valvular_heart_disease	Abnormal heart sounds due to turbulent blood flow across a damaged or malfunctioning heart valve, indicating conditions like mitral valve prolapse or aortic stenosis.
Pedal_and_ankle_oedema	Swelling of the feet and ankles due to fluid retention, often seen in heart failure as the heart struggles to pump blood effectively.
Tachypnoea	Rapid breathing rate, which can be a response to reduced oxygen levels in the blood, often seen in heart or lung conditions.

Clinical data are separated into 2 categories.

- ❖ Clinical dataset 1
- ❖ Clinical dataset 2

These categories are given by the supervision from the doctors

Clinical data set 1

Symptoms of patient	Description
Chest_pain_type	Describes the nature of chest pain, which can help differentiate between cardiac and non-cardiac causes. For example, angina (cardiac-related) is typically described as pressure or tightness, while non-cardiac pain might be sharp or stabbing
Resting_blood_pressure	The pressure of blood against the artery walls when the heart is at rest between beats. It's a crucial measure for diagnosing hypertension, which is a risk factor for heart disease.
Serum_cholesterol	The total amount of cholesterol in the blood, including low-density lipoprotein (LDL) and high-density lipoprotein (HDL). High levels of LDL ("bad" cholesterol) and low levels of HDL ("good" cholesterol) increase the risk of heart disease.
Fasting_blood_sugar	The level of glucose in the blood after an overnight fast. Elevated fasting blood sugar can indicate diabetes or prediabetes, both of which are risk factors for heart disease.

Resting_electrocardiographic_results	The results of an ECG taken while the patient is at rest, showing the heart's electrical activity. It can detect arrhythmias, previous heart attacks, and other cardiac abnormalities.
Maximum_heart_rate	The highest heart rate achieved during maximal exercise. It can be used to assess cardiovascular fitness and the heart's response to physical stress. It is often estimated by the formula 220 minus the person's age.
Exercise_induced_angina	Chest pain or discomfort that occurs during physical exertion. It is typically due to reduced blood flow to the heart muscle, often a sign of coronary artery disease.
ST_depression	A finding on an electrocardiogram (ECG) where the ST segment is lower than the baseline. It can indicate myocardial ischemia, which is insufficient blood flow to the heart muscle, often due to coronary artery disease.
Slope_ST_segment	Refers to the direction and angle of the ST segment on an ECG. The slope can provide information about the presence and severity of ischemia. An upsloping, horizontal, or downsloping ST segment can have different clinical implications.
major_coronary_vessels	The main arteries that supply blood to the heart muscle, including the left main coronary artery, left anterior descending artery, circumflex artery, and right coronary artery. Blockages or narrowing in these vessels can lead to heart disease.
Thalassemia	A genetic blood disorder characterized by less hemoglobin and fewer red blood cells than normal. While not directly a heart disease, severe forms can lead to complications such as heart failure due to chronic anemia.

Clinical data set 2

Symptoms of patient	Description
Presence_of_anemia	A condition characterized by a lower than normal number of red blood cells or hemoglobin. Anemia can worsen heart disease by increasing the heart's workload to supply adequate oxygen to tissues.
Creatinine_phosphokinase	An enzyme found in the heart, brain, and skeletal muscles. Elevated levels in the blood can indicate muscle damage, including myocardial infarction (heart attack).
Diabetes	A chronic condition characterized by high blood sugar levels. Diabetes increases the risk of heart disease by contributing to the development of atherosclerosis and damaging blood vessels.
Ejection_fraction	The percentage of blood the left ventricle pumps out with each contraction. It is a key measure of heart function, with normal values typically ranging from 55% to 70%. Lower values indicate heart failure or cardiomyopathy.
High_blood_pressure	A condition where the force of the blood against the artery walls is consistently too high, leading to increased risk of heart disease, stroke, and other cardiovascular problems.

Platelets	Small blood cells that help with clotting. Abnormal platelet counts can be associated with cardiovascular diseases and conditions that affect the blood's ability to clot properly.
Serum_creatinine	A waste product from muscle metabolism measured in the blood. Elevated levels can indicate impaired kidney function, which is a risk factor for heart disease.
Serum_sodium	The level of sodium in the blood. Abnormal levels can indicate issues with fluid balance and kidney function, which can impact heart health.
Smoking	A major risk factor for heart disease. Smoking damages the lining of arteries, leads to the buildup of plaque, and increases the risk of heart attack and stroke.
Follow_up_period_days	The duration of time over which patients are monitored after initial diagnosis or treatment. This period is important for assessing the long-term outcomes and effectiveness of interventions for heart disease.
Death	Whether the patient was taken from the database is alive or not

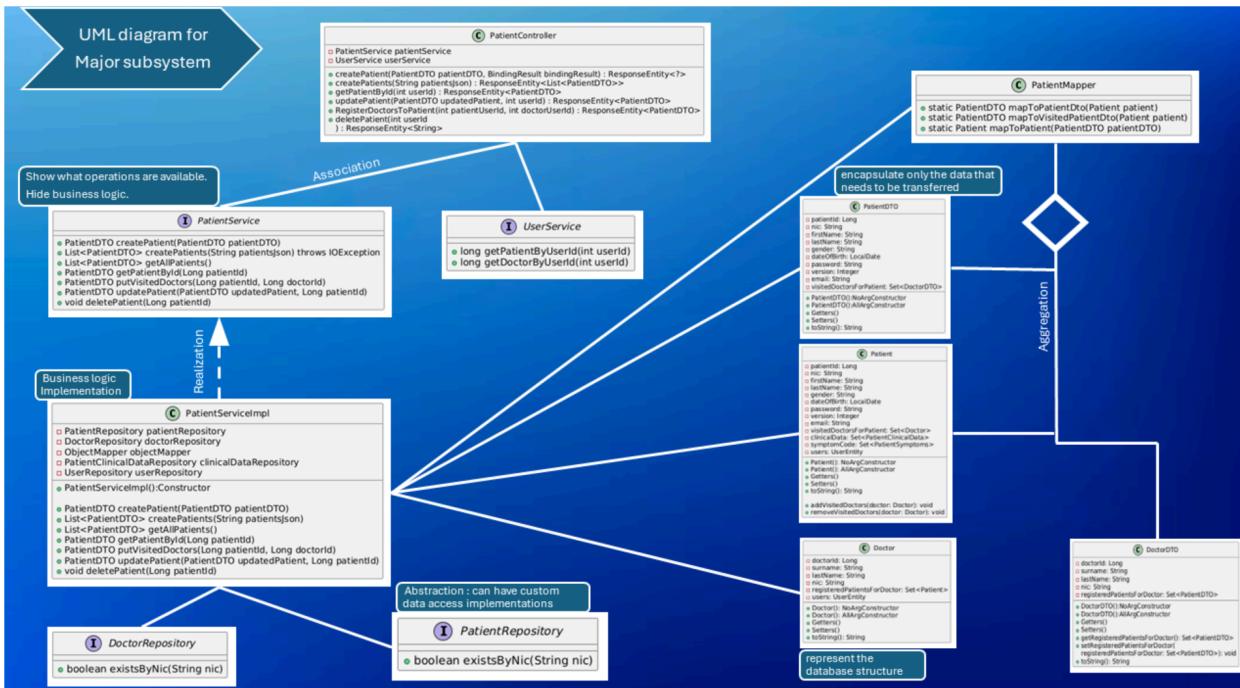
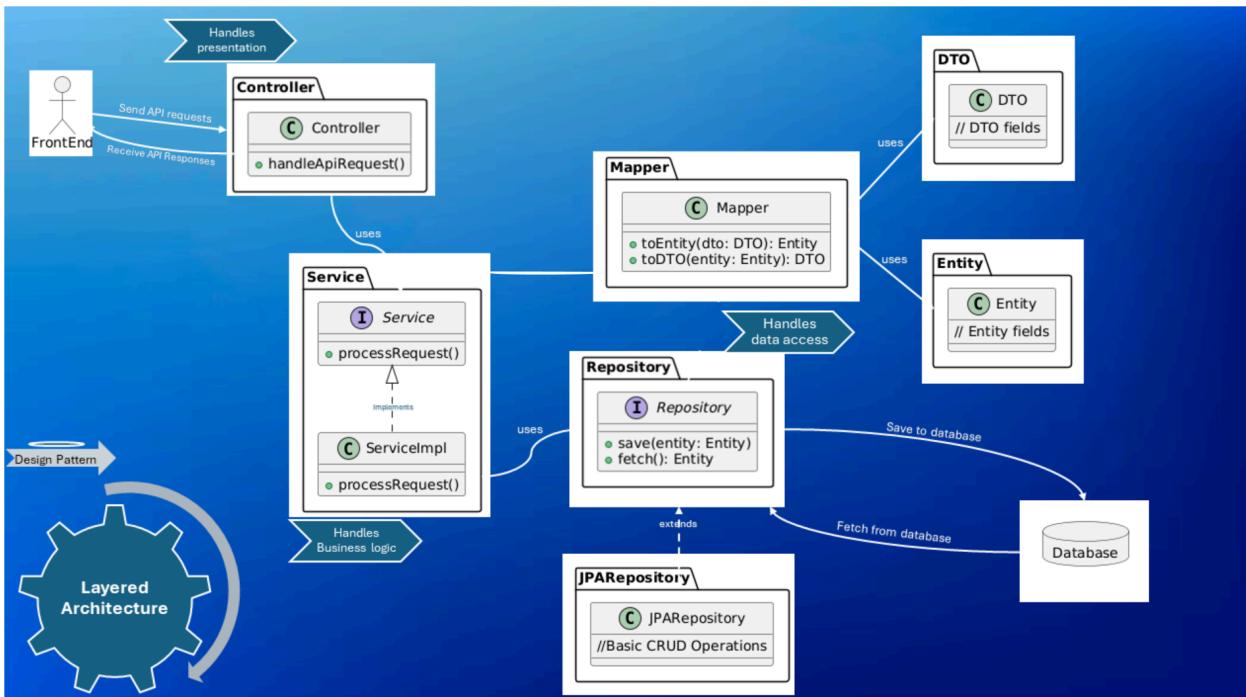
Prediction Data

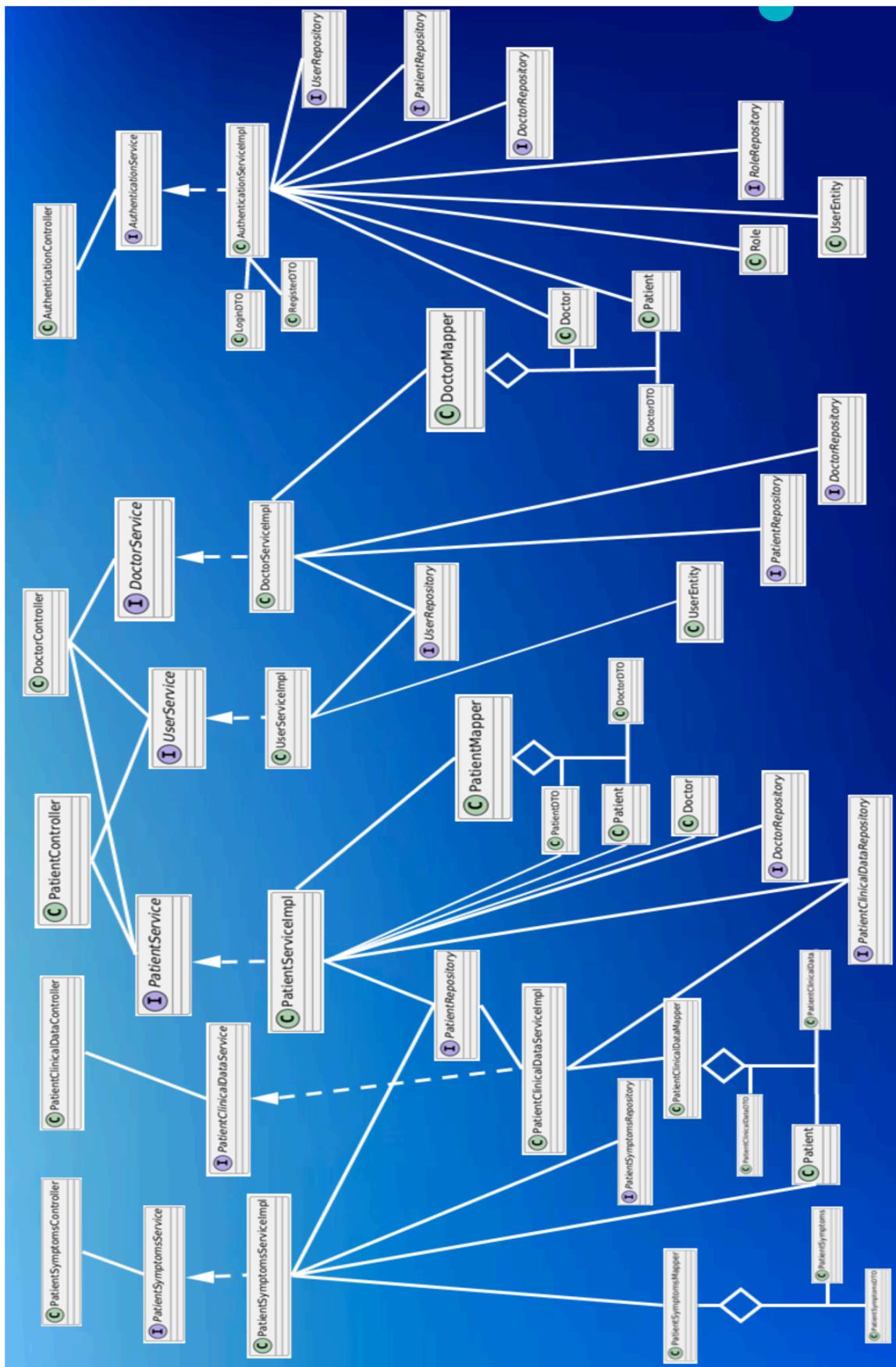
Prediction data are used for the machine learning algorithm used in the system. The description and the parameters are available in [Prediction System].

Advisors and other contributors

- ❖ As advisors, we have the opportunity to get the input from Dr. Damayanthi Herath and Dr Upul Jayasinghe. Dr Upul Jayasinghe was introduced while the project was in progress for further clarifications and further development advises.
- ❖ The requirements of the system was produced by doctors and medical students. Dr Buddhi Dissanayake, also a visiting lecturer in Colombo university, has provided most of the guidance and well as other doctors we inquired. Medical students from the university of Peradeniya and University of Colombo have also provided insight into how the project aspects should be in the medical inputs.
- ❖ Engineer Dushan Atapattu , introduced by Dr Upul, provided us with a comprehensive feedback on how the system is providing us with advise o how the systems should be. Backend implementation was praised by Engineer Dushan with the security approaches and object mapping

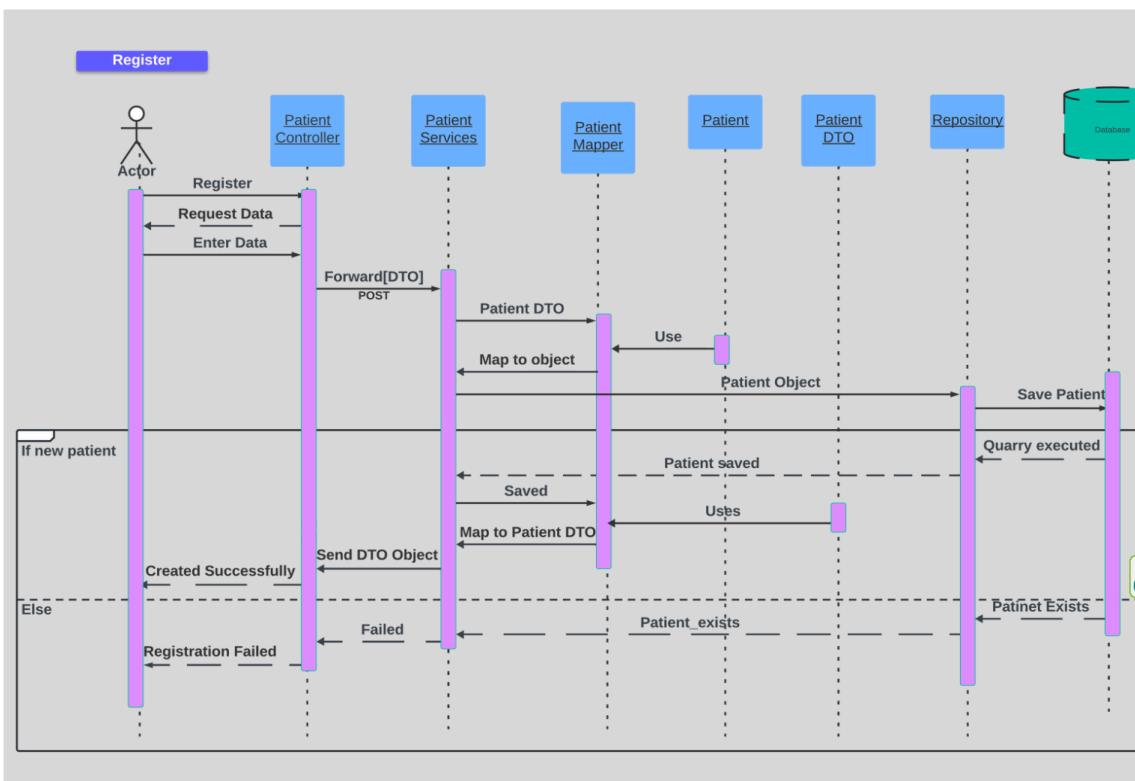
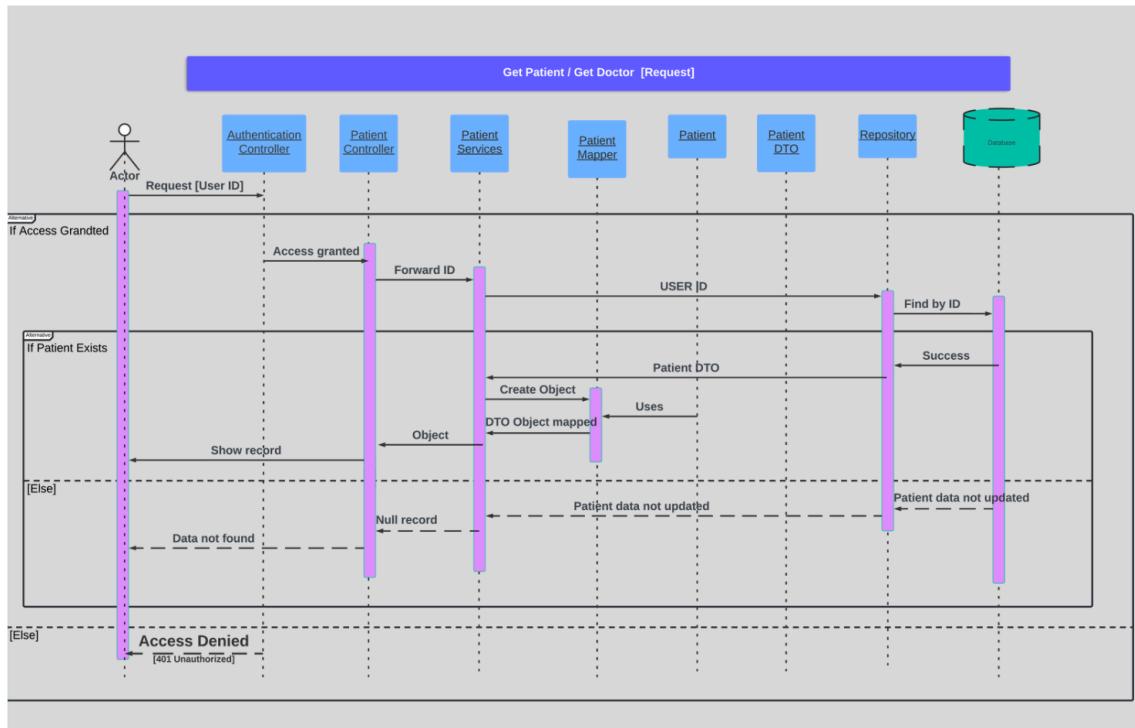
Class Diagram

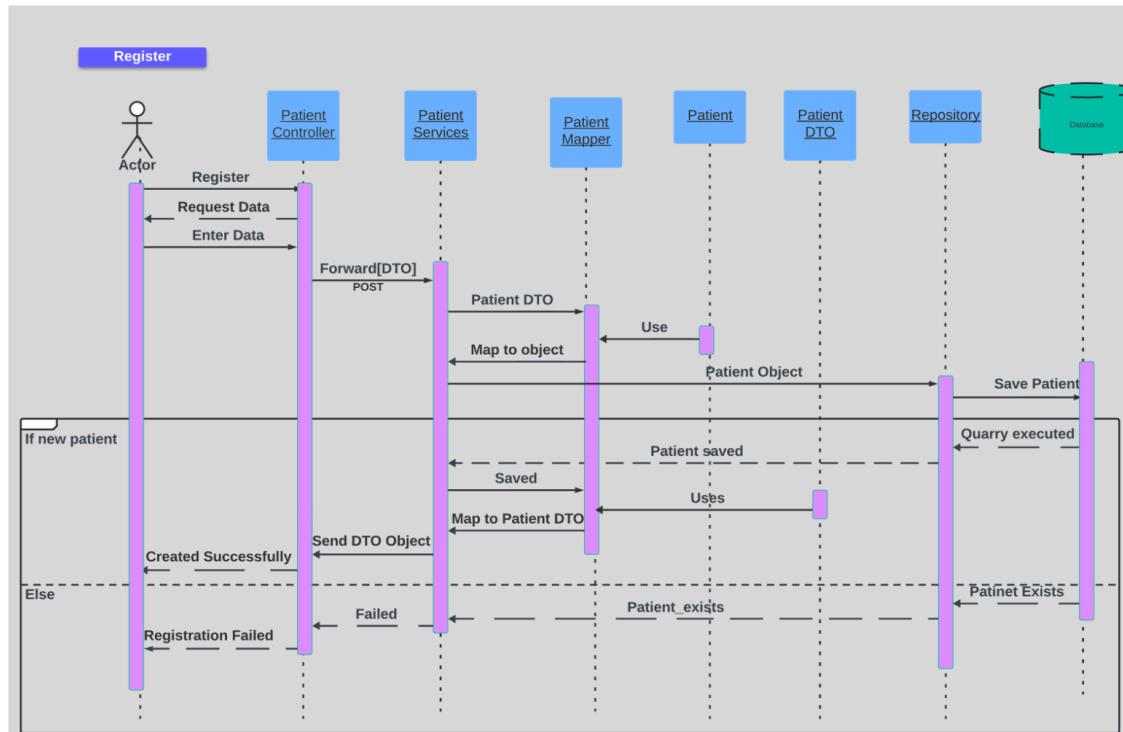
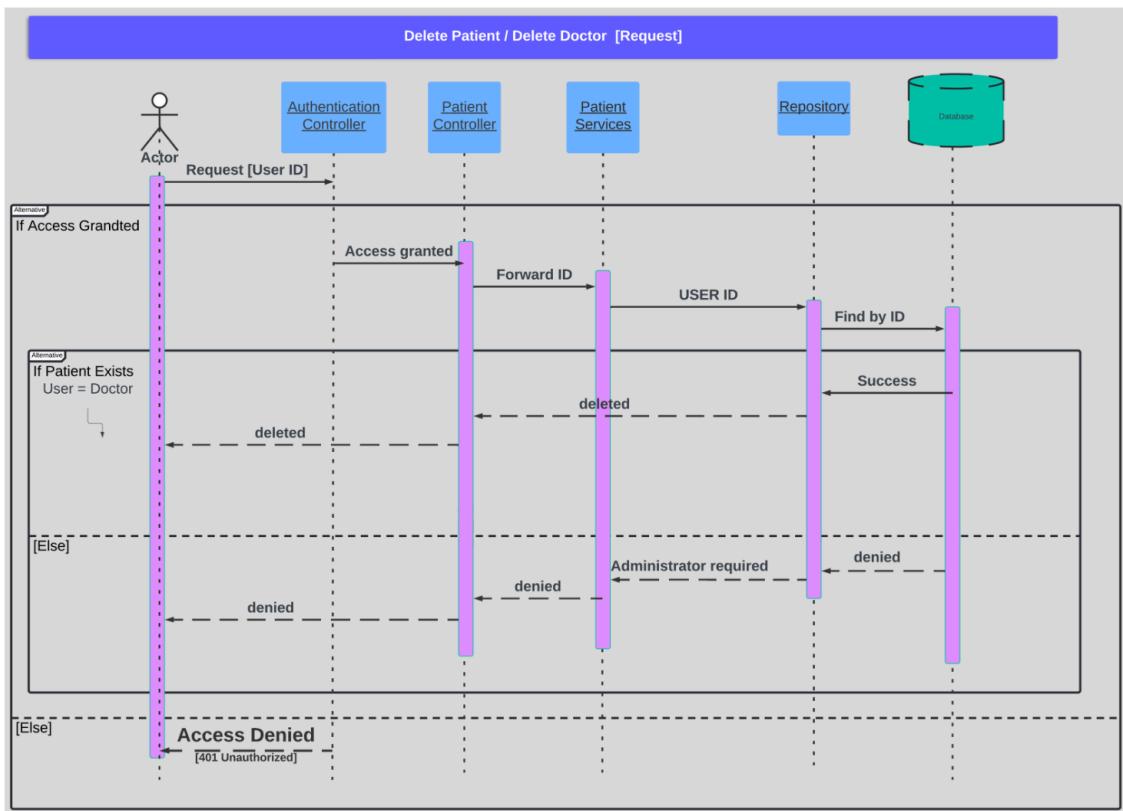




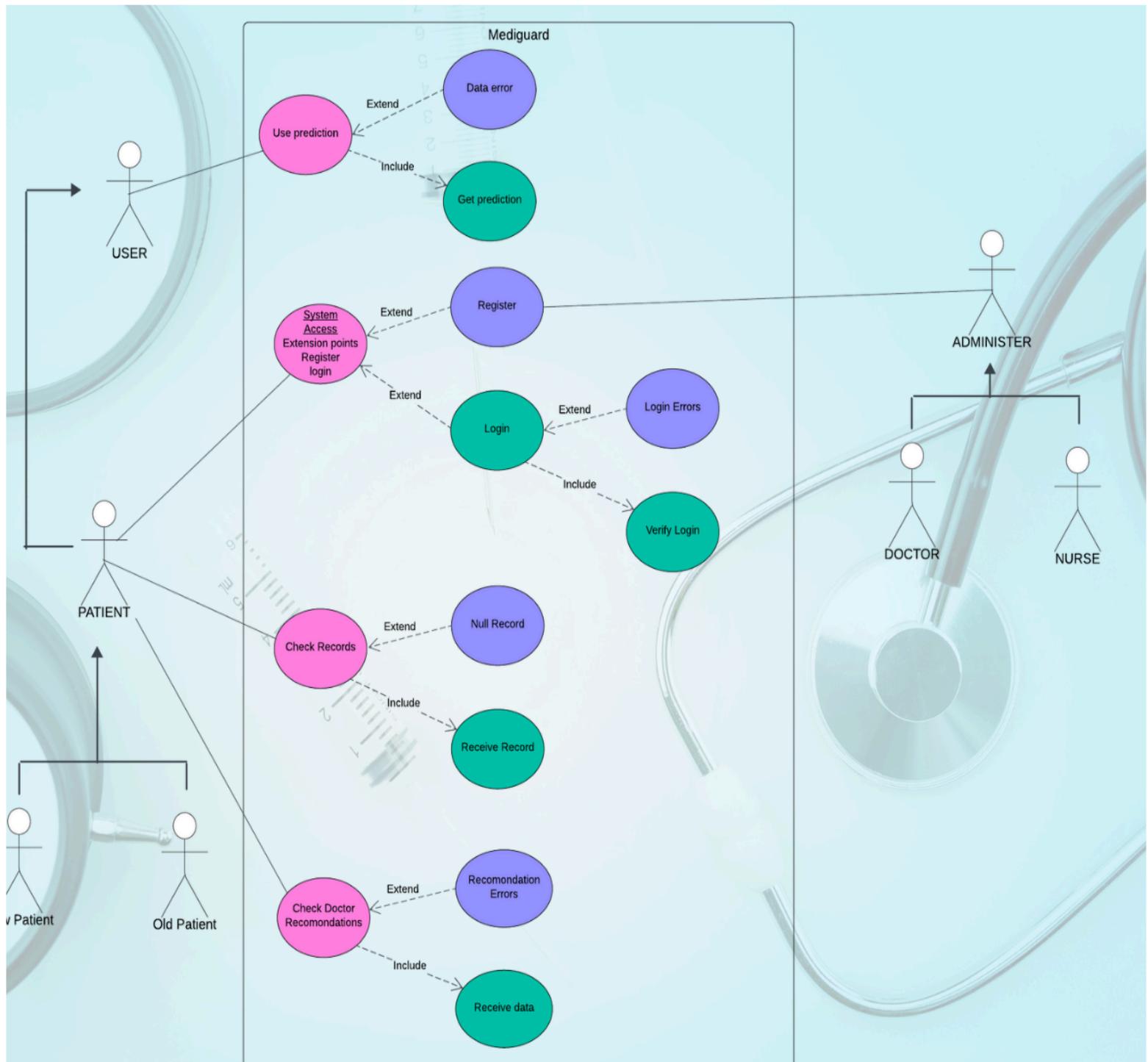
Sequence Diagram

Sequence Diagrams to showcase some of the functionalities in the system

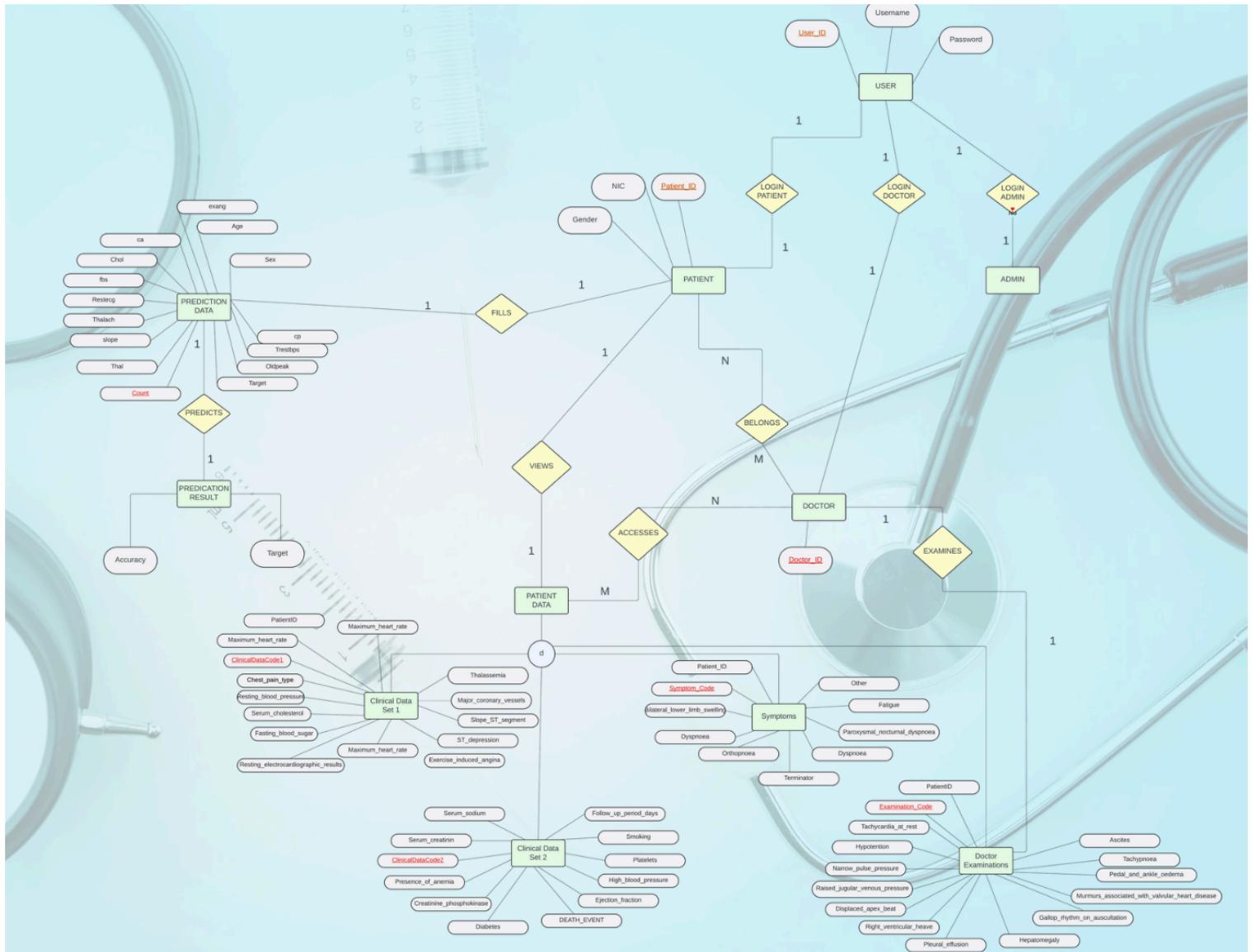




Use Case Diagram



Entity Diagram



Functional Requirements

This system is designed to assist both patients and doctors by providing an efficient and centralized platform to manage patient health records and interactions. Below are the key functional requirements:

System Users:

Patients:

- Patients can register and log in to the system.
- Patients can select a doctor from the available list.
- Patients can view healthy tips and videos to manage their heart health.

Doctors:

- Doctors can access a patient's past medical records.
- If a patient consults a doctor other than their primary doctor, the new doctor can access the patient's historical data for better understanding and treatment continuity.
- Doctors can utilize a machine learning feature integrated into the system to predict heart conditions based on patient data, saving time and enhancing decision-making.

Key Features:

- Centralized Medical Record System:
- Enables seamless sharing of patient data among doctors for better care.
- Machine Learning-Assisted Diagnosis:
- Reduces the time required for analysis and improves diagnostic accuracy.
- Healthy Lifestyle Support:
- Patients receive advice, tips, and videos on preventing heart conditions.

Non-functional Requirements

The system has been designed with a focus on security, performance, and usability.

Security:

- All users must register before accessing the system.
- Doctor accounts are verified by the system admin to ensure authenticity.
- Login requires a valid username and password. Upon successful login, a token is generated to maintain session security. The token is validated during navigation within the site to prevent unauthorized access.

Performance:

- The system is designed to handle multiple users simultaneously without performance degradation.
- Code is modularized into smaller, reusable functions to ensure maintainability and scalability.
- Usability:
- A visually appealing and user-friendly interface ensures smooth navigation for all user types.
- The inclusion of educational content, such as videos and tips, enhances user engagement and provides additional value.

Use Case Descriptions

Use Case 1 – Prediction system

The **Doctor Use Case** focuses on enabling doctors to manage patient health by leveraging a predictive health system. Doctors log in to a secure dashboard where they can upload patients' medical details, such as lab results or health records. The system processes this data to provide a personalized heart attack risk prediction for each patient, displayed as a percentage for clarity. Additionally, doctors receive tailored health recommendations that can be shared with patients and can access patients' health history to monitor progress over time. This use case empowers doctors to make informed decisions about patient care, ensuring proactive health management and improved outcomes.

Use Case 2 – Assign patients to doctors (by admin)

The task of **assigning patients to appropriate doctors** involves matching patients with doctors based on specific criteria, such as the doctor's specialization, workload, and the patient's medical condition or requirements. This process ensures that each patient is assigned to a doctor best equipped to handle their health concerns, thereby improving the quality of care. The task of **assigning patients to doctors** is streamlined by the system's search and filtering capabilities, enabling admins to efficiently match patients with the most suitable doctors. Admins can use the system's search functionality to locate doctors or patients quickly by entering specific details, such as their name, ID etc. Once the admin selects a patient and doctor, the system provides an intuitive interface to assign them, ensuring that the process is quick and error-free. This approach eliminates the need for manual effort in searching through extensive lists and allows admins to manage assignments effectively, even in a large-scale system. This streamlined assignment process helps maintain smooth operations and ensures that patients receive timely and appropriate care.

Use Case 3 – Patient can get overall medical history by clicking specific date

The **Patient Use Case – Accessing Overall Medical History by Date** allows patients to view their complete health records conveniently through the system. By selecting a specific date on the dashboard, patients can retrieve all relevant medical details and activities associated with that date. The process begins with the patient logging into their secure account and navigating to the "Medical History" section. The system then displays a calendar or date-picker interface where the patient can select a specific date of interest. Once the date is selected, the system retrieves and displays all medical details recorded for that day, such as uploaded test results, consultations, prescribed medications, or risk predictions. Patients can view this detailed information in an organized format, which may include doctor notes, recommendations, and follow-up actions. This feature helps patients keep track of their medical journey, identify trends, and stay informed about their health progress over time. It also fosters better patient engagement by making health information easily accessible and actionable.

Use case 4 – Doctors can get overall idea of each patient by looking at graphs showed by the system(eg. How vary cholesterol level each day)

The **Doctor Use Case – Viewing Patient Data Through Graphs** enables doctors to gain a comprehensive understanding of each patient's health trends through graphical representations provided by the system. These graphs display key health metrics, such as daily variations in cholesterol levels, blood pressure, or other monitored parameters, allowing doctors to assess the patient's condition at a glance. Doctors can log into their secure dashboard, select a specific patient, and view intuitive graphs that highlight health trends over time. By analyzing these visualizations, doctors can quickly identify anomalies, track the effectiveness of treatments, and make data-driven decisions. This feature helps doctors provide personalized and proactive care while improving the efficiency of patient monitoring and diagnosis.

Use case 5 – Admin can get overall idea about the system by looking at only dashboard

The **Admin Use Case – Viewing System Overview via Dashboard** allows admins to monitor the overall status of the system through a comprehensive dashboard that displays key metrics in an easily digestible format. The dashboard includes visual elements such as pie charts and other graphs that provide real-time data on the system's performance. Admins can instantly see the total number of patients and doctors, along with demographic breakdowns such as the percentage of male and female patients, and the distribution of age groups (e.g., percentage of elders, children, etc.). These visual insights help admins quickly assess the system's usage, track user demographics, and identify trends or imbalances in the patient-doctor ratio. By providing a quick, at-a-glance overview, the dashboard enables admins to make informed decisions about system management, resource allocation, and overall system efficiency.

In conclusion, the system provides a user-friendly and efficient way for doctors, patients, and admins to manage and access health information. Doctors can gain valuable insights into patient data through intuitive graphs, while patients can easily track their medical history by date. Admins have a clear overview of the system's operations and demographics through comprehensive dashboards. Overall, this system enhances the patient experience, supports informed decision-making by healthcare providers, and ensures smooth and effective management for admins.

Technology and Implementation

Technology

For the project, mainly the following technologies were used,

- ❖ Java and Spring boot
 - The implementation of the backend was requested to be done on Java for the module requirement of CO225
 - Spring boot was used as a framework OF Java
 - Java libraries were used for functional implementations of the system



❖ React and Tailwind Css

- the styling was done using Tailwind CSS which are both industry leading technologies up to date.
- Reusability of components and the easier implementation was the reasons in using React.
- Tailwind CSS greatly complements React



-
- ❖ Python for machine learning
 - Machine learning was used in the project for predicting the see the possibility of patients being in the range of a heart failure using the data in the system.
 - Confusion matrix was constructed to verify the accuracy



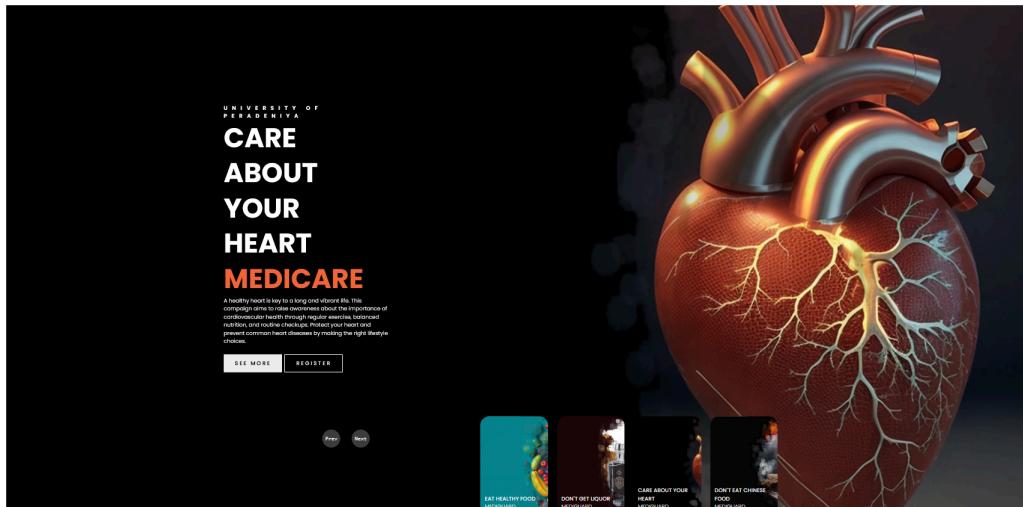
Front End development with user manual

User Manual

This provides how to use the system and also show cases the front end development of the system.

❖ Home Page

- Provides users with an interface with tips and information
- Also provides the useful links for navigation



- Provides useful links for youtube videos with relevance

Let's Talk About Heart Health

Watch on YouTube

Care About Your Heart

Maintaining heart health is vital for overall well-being. This blog highlights the importance of regular cardiovascular exercise, a balanced diet rich in fruits and vegetables, and routine health checkups. Adopting these practices can help prevent heart disease and ensure a longer, healthier life.

Don't Eat Chinese Food

Chinese cuisine is often high in sodium and unhealthy fats, which can contribute to heart disease. This blog suggests avoiding excessive consumption of Chinese food and opting for more heart-healthy options like fresh vegetables and lean proteins. Maintaining a balanced diet is key to heart health.

Watch on YouTube

❖ About Page

- Provides information about the system and the legal policy of the system

❖ Contact Us page

- Provides means of contact through the website and also provide information about who to contact.

Get in Touch

Name

Email

Message

Contact Information

Email

Phone

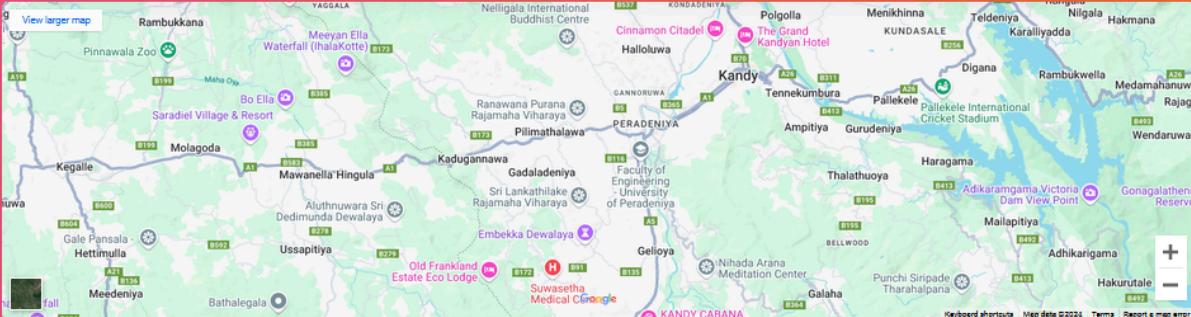
Address

Office Hours

Monday - Friday: 9 AM - 6 PM
Saturday: 10 AM - 4 PM

Follow Us

[f](#) [t](#) [i](#)



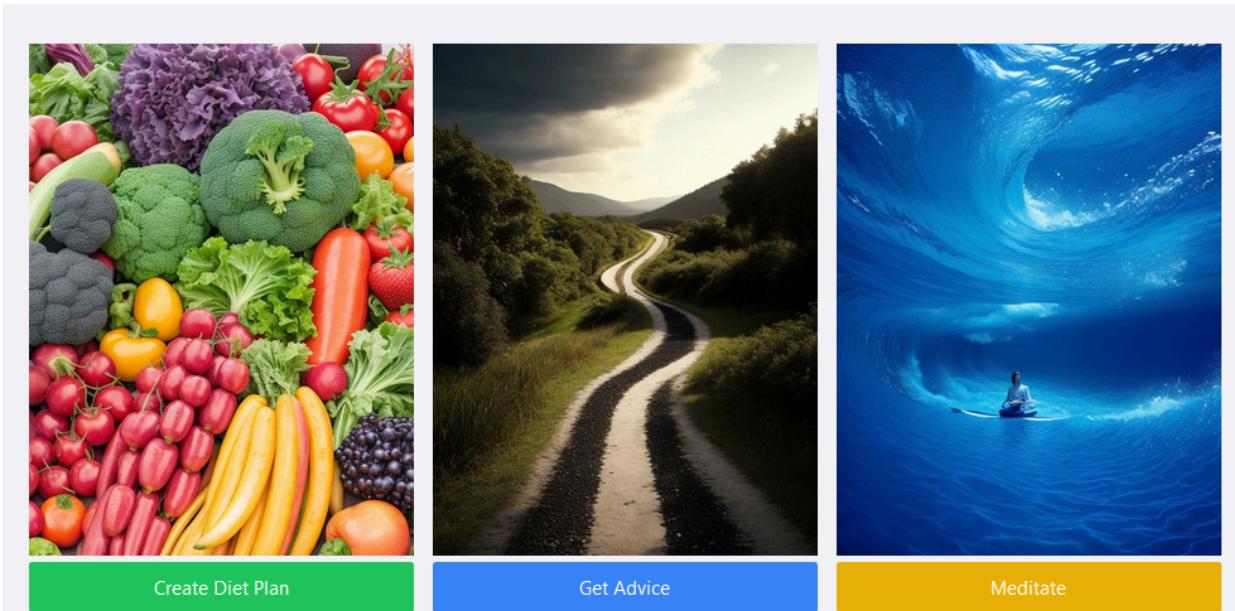
A detailed map of the Kandy area in Sri Lanka, showing major roads like A1, A2, and A3, as well as numerous landmarks such as Rambukkana, Yaggala, Meeyan Ella Waterfall, Nelligala International Buddhist Centre, Cinnamon Citadel, The Grand Kandy Hotel, Pallekelle, and the Pallekelle International Cricket Stadium. The map also includes smaller settlements like Molagoda, Kadugannawa, and Gadaladeniya.

© 2024 CardioGuard. All rights reserved.
[Privacy Policy](#) [Terms of Service](#)

[f](#) [t](#) [i](#)

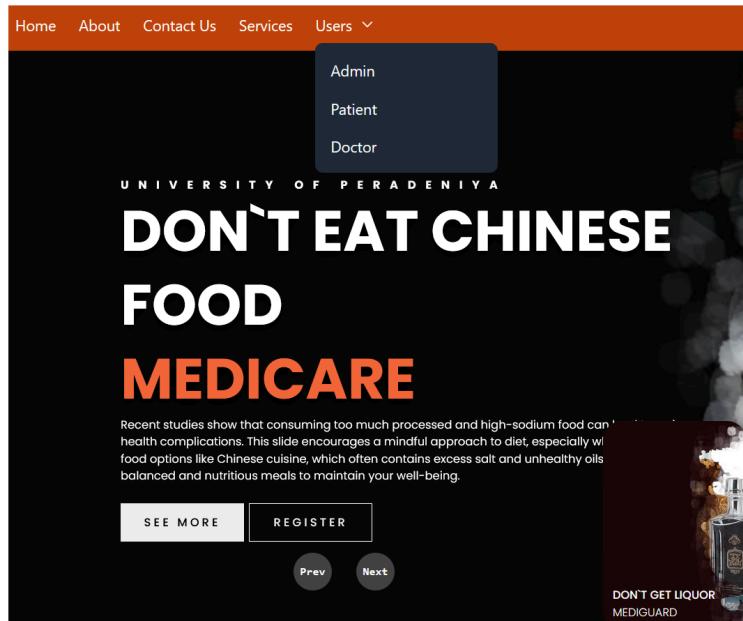
❖ Services page

- Provides quick location fetching methods
- Provides a diet planner for heart patients
- Provides a portal for patients to seek advice
- Shows a map of all the nearest hospitals to the patient



❖ Users Page

- Admin
- Doctors
- Patients



The Users tab is a navigation method that allows users to go to their specific pages with data.

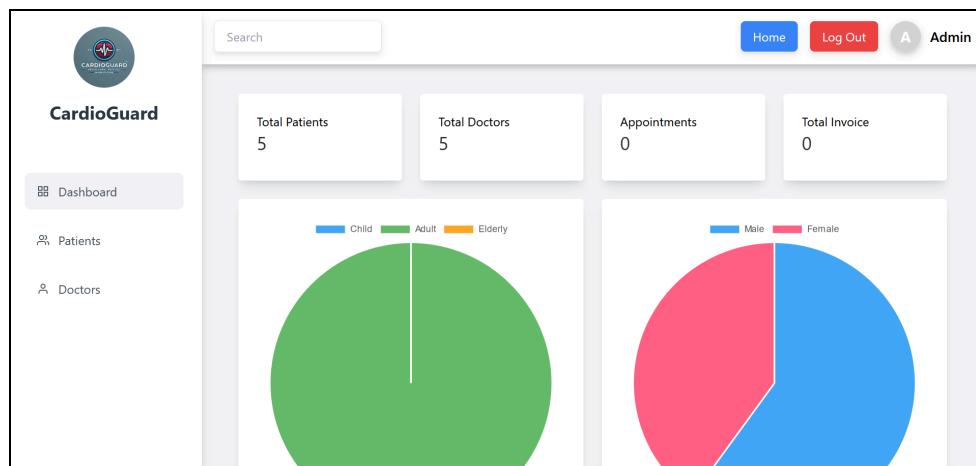
Each type of user has their own type of dashboards.

Dash Boards

❖ Admin Dashboard

Admin dashboard allows administrators to get the data about the system, patients and doctors

❖ 3 aspects ➤ DashBoard



Patients		Doctors	
Search by First Name, Last Name or NIC		Search by Doctor ID, Surname, Last Name, or NIC	
Patient ID	Name	Doctor ID	Name
1	Kumara Perera	1	Dr. Andrew Rodrigue
2	Lakshman Wijesundara	2	Dr. Sidath Samaratunge
3	Samanthi Rajakaruna	3	Dr. Harshika Perera
4	Kusal Mendis	4	Dr. Mukesh Sudarshan
5	Sarala Minoli	5	Dr. Herathi Samanpriya

➤ Patients

- All the patients are available.

The screenshot shows the CardioGuard application interface for managing patients. On the left, there's a sidebar with icons for Dashboard, Patients (which is selected), and Doctors. The main area has a search bar at the top. Below it, a message says "Click the patient to delete or update". A table lists five patients with columns for Patient ID, NIC, First Name, Last Name, Gender, and Date of Birth. At the bottom, there are "Previous" and "Next" buttons, and a note "Page 1 of 1".

Patient ID	NIC	First Name	Last Name	Gender	Date of Birth
1	2002342234587v	Kumara	Perera	Male	1/1/2003
2	200160143466v	Lakshman	Wijesundara	Male	8/7/2001
3	199847154559v	Samanthi	Rajakaruna	Female	4/19/1998
4	199641223445v	Kusal	Mendis	Male	12/3/1996
5	200432457789v	Sarala	Minoli	Female	4/24/2001

➤ Doctors

The screenshot shows the CardioGuard application interface for managing doctors. The sidebar includes icons for Dashboard, Patients, and Doctors. The main area has a search bar and a "Doctors List" section with a table of five doctors. Each doctor entry has "Update" and "Delete" buttons. Below this is a "Register Patient to Doctor" dialog box. It has two dropdown menus: "Select Doctor:" and "Select Patient:". Each menu has a search bar and a list of options. At the bottom is a "Register Patient to Doctor" button.

Doctor ID	NIC	Surname	Last Name	Actions
1	852741968V	Andrew	Rodrigue	Update Delete
2	199410395738V	Sidath	Samarayake	Update Delete
3	199845271324V	Harshika	Perera	Update Delete
4	872423158V	Mukesh	Sudarman	Update Delete
5	8527419856V	Herathi	Samanpriya	Update Delete

Different functionalities are allowed for admin to change the data in system.

❖ Doctor Dashboard

- Doctor dashboard has a similar interface
- 3 aspects
 - Doctor Dashboard

The screenshot shows the 'Patients' section of the CardioGuard Doctor Dashboard. At the top, there is a search bar labeled 'Search'. Below it, a table displays patient information with columns for 'Patient ID', 'Name', and 'NIC'. The data is as follows:

Patient ID	Name	NIC
1	Kumara Perera	2002342234587v
2	Lakshman Wijesundara	200160143466v
3	Samanthi Rajakaruna	199847154559v
4	Kusal Mendis	199641223445v
5	Sarala Minoli	200432457789v

■ Doctor - Patients bar

The screenshot shows the 'Patients Registered to Doctor' section of the CardioGuard Doctor Dashboard. At the top, there is a search bar labeled 'Search'. Below it, a table displays patient information with columns for 'Patient ID', 'First Name', 'Last Name', 'Gender', and 'Date of Birth'. The data is as follows:

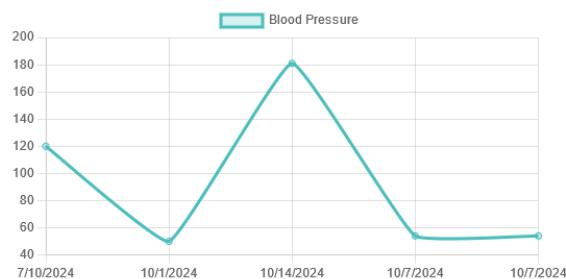
Patient ID	First Name	Last Name	Gender	Date of Birth
5	Sarala	Minoli	Female	4/24/2001
3	Samanthi	Rajakaruna	Female	4/19/1998
4	Kusal	Mendis	Male	12/3/1996
1	Kumara	Perera	Male	1/1/2003
2	Lakshman	Wijesundara	Male	8/7/2001

The screenshot shows the 'Patient Symptom Data' section of the CardioGuard Doctor Dashboard. At the top right, it says 'Patient ID : 2'. Below it is a button labeled 'Add Record'. A table displays symptom data with columns for 'Date(M/D/Y)', 'Details', and 'Doctor Recommendation'. The data is as follows:

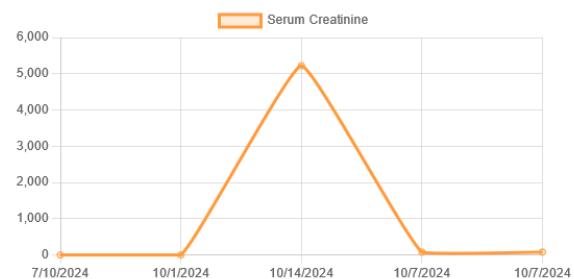
Date(M/D/Y)	Details	Doctor Recommendation
10/18/2024	Show Details	Good
8/15/2024	Show Details	Limit salt intake, elevate legs while resting, and follow up in one week.

Clinical Data Charts

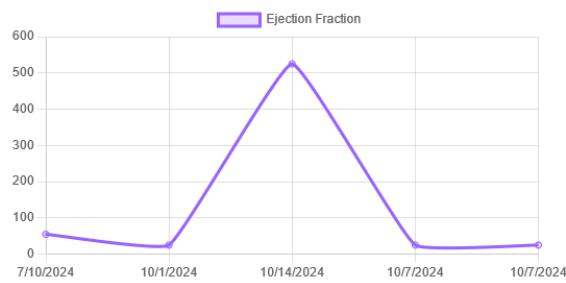
Blood Pressure



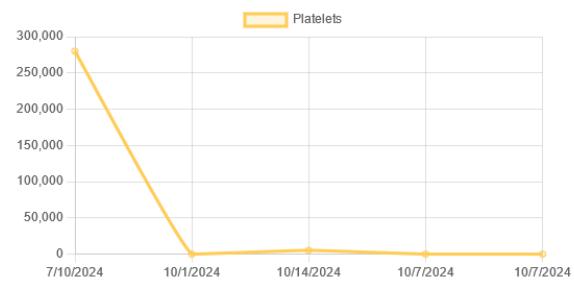
Serum Creatinine



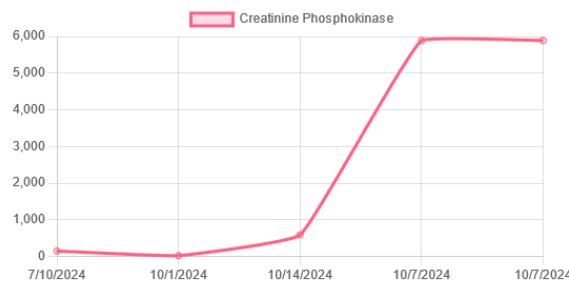
Ejection Fraction



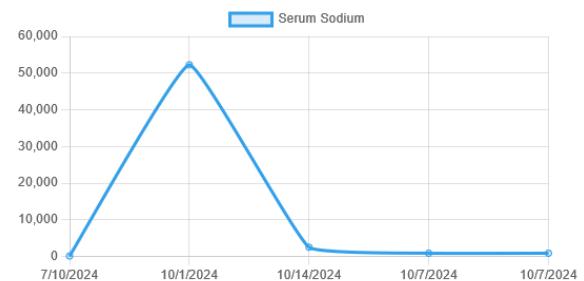
Platelets



Creatinine Phosphokinase



Serum Sodium



- ❖ When a patient is clicked, the person's medical details are displayed to the doctor
- ❖ Comprehensive details are provided through the use of Graphs

■ Doctor - Prediction bar

The screenshot shows the CardioGuard software interface. At the top left is the CardioGuard logo. To its right is a search bar labeled "Search". On the far right are three buttons: "Home" (blue), "Log Out" (red), and a help icon (?). Next to the "Log Out" button is a placeholder for "DoctorName".

The main area is titled "The Prediction System - Evaluation of Patients". It contains several input fields for patient evaluation:

- Enter your age:** Two input fields for "years" (0 or 1) and "Sex (0 = Female, 1 = Male)" (0 or 1).
- Chest Pain Type (0-3):** Two input fields for "type" (0 or 1).
- Serum Cholesterol:** Two input fields for "mg/dL" (0 or 1).
- Resting Blood Pressure:** Two input fields for "mm Hg" (0 or 1).
- Fasting Blood Sugar (1 if > 120 mg/dL):** Two input fields for "0 or 1" (0 or 1).
- Resting ECG Result:** Two input fields for "0-2" (0 or 1).
- Maximum Heart Rate:** Two input fields for "bpm" (0 or 1).
- Exercise-Induced Angina (1 = Yes):** Two input fields for "0 or 1" (0 or 1).
- ST Depression:** Two input fields for "mm" (0 or 1).
- Slope of Peak Exercise ST Segment:** Two input fields for "0-2" (0 or 1).
- Number of Major Vessels Colored by Fluoroscopy:** Two input fields for "0-3" (0 or 1).
- Thalassemia (0-3):** Two input fields for "0-3" (0 or 1).

At the bottom center is an orange "Submit" button.

❖ Patient Dashboard

- Patient dashboard has a similar interface
- 5 aspects

The screenshot shows the CardioGuard Patient Dashboard for Lakshman Wijsundara. The top navigation bar includes a search bar, a user icon, and links for Home, Log Out, and Edit Data. The main content area is divided into several sections:

- Patient Info:** First Name: Lakshman, Last Name: Wijsundara, Patient ID: 2, NIC: 200160143466v, Email: N/A.
- General Info:** Gender: Male, Age: 23 years old, Date of Birth: 2001-08-07, Occupation: Software Engineer, Insurance: HealthPlus.
- Days Until Next Clinic Day:** -11 Days.
- Select a Date:** A calendar for October 2024 showing the current date as Friday, October 11, 2024.

- Patient dashboard

A simplified mockup of the Patient Info and General Info sections from the dashboard. Both sections are contained within a single large box.

Patient Info		General Info	
First Name Lakshman	Last Name Wijsundara	Gender Male	Occupation Software Engineer
NIC 200160143466v	Age 23 years old	Date of Birth 2001-08-07	Insurance HealthPlus
Email N/A			

Select a Date

October 2024						
MON	TUE	WED	THU	FRI	SAT	SUN
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

Selected Date: Fri Oct 11 2024

Days Until Next Clinic Day

-11 Days

➤ Medical Information

Patient Symptoms Info

Bilateral Lower Limb Swelling No	Dyspnoea Yes	Orthopnoea No	Paroxysmal Nocturnal Dyspnoea Yes
Fatigue No	Symptom Date 8/15/2024	Doctor Recommendation Limit salt intake, elevate legs while resting, and follow up in one week.	

Patient Clinical Data

Diagnosis of Heart Disease No	Presence of Anemia Yes	Creatinine Phosphokinase 150	Diabetes No
Ejection Fraction 55%	Blood Pressure 120	Platelets 280000	Serum Creatinine N/A
Serum Sodium 145	Smoking Yes	Follow-Up Period (Days) 45	

Examination Data

Tachycardia at Rest Yes	Hypotension No	Raised Jugular Venous Pressure Yes	Displaced Apex Beat No
Right Ventricular Heave No	Pleural Effusion Yes	Hepatomegaly No	Galloping Rhythm on Auscultation Yes
Pedal and Ankle Oedema Yes	Tachypnoea No	Ascites No	Examination Date 7/1/2024

➤ Doctors

Doctors

Search by Doctor ID, Surname, Last Name, or NIC

Doctor ID	Name
1	Dr. Andrew Rodrigue
2	Dr. Sidath Samaratunge
3	Dr. Harshika Perera
4	Dr. Mukesh Sudarman
5	Dr. Herathi Samanpriya

➤ ReadMore

Patient Clinical Data Overview

Diagnosis of Heart Disease: Yes
Indicates a heart condition diagnosis which impacts overall health management.

Presence of Anemia: No
The patient does not show signs of anemia, which affects oxygen transport in the body.

Diabetes: Yes
The patient has diabetes, a condition that requires careful blood sugar management.

Clinical Marker Ranges

Clinical Marker	Normal Range	Elevated Range	Critical Range	Meaning
Blood Pressure	90-120 mmHg	121-139 mmHg	> 140 mmHg	High BP can increase heart disease risks.
Serum Creatinine (mg/dL)	0.6-1.2 mg/dL	1.3-1.5 mg/dL	> 1.5 mg/dL	Elevated levels indicate possible kidney issues.
Platelets	150,000-450,000 cells/mcL	450,000-500,000 cells/mcL	> 500,000 cells/mcL	High levels can lead to clotting issues.

Clinical Data Trends

Time Point	Blood Pressure (mmHg)	Serum Creatinine (mg/dL)
1	115	110
2	125	120
3	120	115
4	130	125

BackEnd

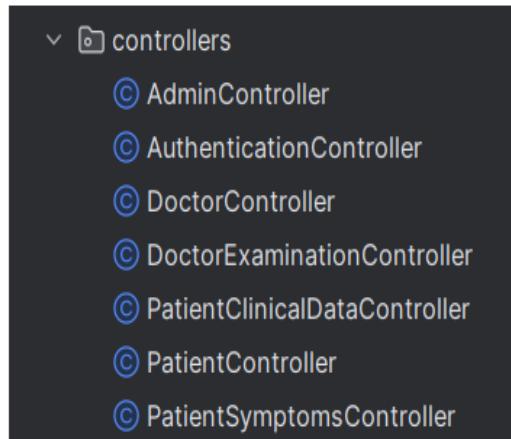
Implementation of Back end Components

- ❖ The back end is created using JAVA
- ❖ There are 8 specific components that contribute to the implementation of the back end
 - Controllers
 - Domains
 - Entity
 - Mapper
 - Exception
 - Repository
 - Security

Controllers and Domains

❖ Controllers

A controller serves as an intermediary between the client and the server. For a Spring Boot application, this is where it interacts with the HTTP requests and then delegates those requests to the appropriate service layers for processing. Controllers act in an intermediary manner, in that they take what the user has done and translate that into an action that the application will undertake. Controllers usually define the routing and map it to a method that will be an action corresponding to a REST operation: GET, POST, PUT, or DELETE. For example, if a user sends a request to get the list of patients, the controller will delegate that request to the service layer, handle the response, and return the proper data. This is modular in that it keeps the logic tidy, separating the request handling from the business logic and data access layers.



In our system , controllers are created for Admin , Authentication , Doctor ,Doctor Examinations,Patient ClinicalData, Patient and Patient symptom classes

❖ Domains

Domains, or domain models, are major business objects and concepts involved in an application. These are abstractions of real-life entities with which the application will deal, and quite often, are closely related to the core business logic. A hospital management system, for example, might have domain models for entities such as Patient, Doctor, or Appointment. Each of the domains encapsulates the respective attributes and behaviors of these entities, concerning their main responsibilities within the system. Domain models help provide structure to an application and ensure that the system is representative of the real-world setting it is aiming or designed to manage, along with the relationships and constraints that represent the business rules.

Our system consists of the following Domains

- © AuthenticationResponseDTO
- © DoctorDTO
- © DoctorExaminationDTO
- © LoginDTO
- © PatientClinicalDataDTO
- © PatientDTO
- © PatientSymptomsDTO
- © PredictionDataDTO
- © RegisterDTO

Entity and Mapper

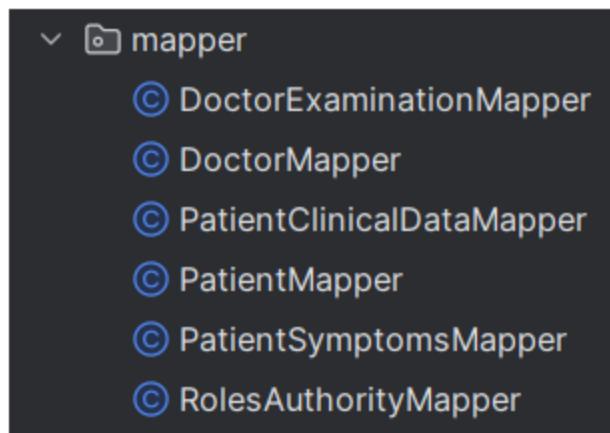
❖ Entity

The entity is an object that represents data stored in a database. An entity in a Spring Boot application is normally annotated with `@Entity` and is designed to map into a database table. Every such class has fields, corresponding to the table columns, representing the attributes of that object, like `id`, `name`, or `createdAt`. Entities are central regarding the data persistence layer since they enable the possibility to easily interact with the database by means of JPA in a straightforward manner. For example, a `User` entity would be equivalent to a `users` table in a database. Any instance of the `User` entity would be one row of such a table. In this way, the encapsulation of the database structure enables the system to remain object-oriented.

- © Admin
- © Doctor
- © DoctorExamination
- © Patient
- © PatientClinicalData
- © PatientSymptoms
- © PredictionData
- © Role
- © UserEntity

❖ Mapper

Mappers are utilities or components responsible for object transformations from one type to another. It is commonly used in the mapping between a domain model and its DTO. The transfers between layers are done by DTOs, which normally happen from client to server and vice-versa. On the other hand, the domain model may represent the very core of the business area. It is the mappers' job to implement these transformations, so that data can be in proper format and so communications within the system are easy. For example, the UserMapper might be used to transform a UserDTO that has been used in the frontend to a User entity for persistence to a database, and vice versa. This separation will keep the internal structure of entities and the external API contracts clean and maintainable.



Exception and Repository

❖ Exception

Exception handling in Spring Boot provides a robust way to manage errors and ensures the application responds elegantly in case of failures. It is possible to define custom exceptions for specific error scenarios, like record not found, invalid input, or access violation. This will be the global exception handling for an application using annotations like `@ControllerAdvice` and `@ExceptionHandler`; hence, the developers can define a unified approach for error handling. For instance, when a client requests a resource that isn't available, it could throw a custom `ResourceNotFoundException`, returning a relevant error message and corresponding HTTP status code to the client, say a 404 status code. In this regard, better user experience also means adequate feedback when something goes wrong.

- © `ConstraintViolationException`
- © `DuplicateException`
- © `GlobalExceptionHandler`
- © `InvalidDataFormatException`
- © `ResourceNotFoundException`
- © `ServiceException`

- ① `DoctorExaminationRepository`
- ① `DoctorRepository`
- ① `PatientClinicalDataRepository`
- ① `PatientRepository`
- ① `PatientSymptomsRepository`
- ① `RoleRepository`
- ① `UserRepository`

❖ Repository

The repositories are one of the key aspects concerning data access in Spring Boot applications. They provide a clean abstraction over the database by exposing methods for querying, saving, updating, or deleting data without having to write complex SQL queries. By extending interfaces like `JpaRepository` or `CrudRepository`, the common database operations are inherited by default in the repository. For example, a `UserRepository` might enable developers to operate on `User` entities to find users by their id, or retrieve all users in the system. That encapsulation will make the interaction with the database much easier and will let the developer be focused only on business logic while the repository takes care about those underlying data operations.

Security and Services

❖ Security

Spring Boot Security handles the protection mechanism of an application from illegal access to the application and other malicious activities. Spring Security is one of the most strong modules in the Spring ecosystem, providing facilities for various principles, such as authentication-who can log in-and authorization-what can be done. It implements roles and permissions for access, allowing different levels of users to access information: ADMIN, DOCTOR, or NURSE, depending on their role. Security configurations have been implemented like password encoding, session management, and token-based authentication-for example, JWT-to ensure sensitive endpoints are reached only by the authenticated user. This approach gives layered security on both the system and the data it manages

- ⌚ JwtAuthenticationEntryPoint
- ⌚ JWTAuthenticationFilter
- ⌚ JWTGenerator
- ⌚ SecurityConfig
- ⌚ SecurityConstants

❖ Services

In the core of a Spring Boot application is hosted core business logic in services. These are responsible for processing the data, implementing business rules, and communicating with the repository for data retrieval or persistence. That is the normal way: a controller calls a service to process the request and carry out the operation that corresponds to it. For instance, a UserService would have methods for user registration, updating user information, or deleting users. It encapsulates the complicated operations of input validation, application of business rules, and ensuring the consistency of data so that the service layer can keep business logic well-organized and permit reusability throughout an application.

```
□ AdminServiceImpl.java  
© CustomUserDetailsServiceimpl  
© DoctorExaminationServiceImpl  
© DoctorServiceImpl  
© PatientClinicalDataServiceImpl  
© PatientServiceImpl  
© PatientSymptomsServiceImpl  
© UserServiceimpl  
① DoctorExaminationService  
① DoctorService  
① PatientClinicalDataService  
① PatientService  
① PatientSymptomsService  
① UserService
```

Machine Learning

The concept of machine learning has been used in this system to identify whether a cardiac patient's data shows the presence of a heart disease or not. The concept is implemented by using a set of parameters and finding whether the parameters diverge towards either True or False states to determine the output of the data.

The Prediction System - Evaluation of Patients

<p>Enter your age</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">years</div>	<p>Sex (0 = Female, 1 = Male)</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">0 or 1</div>
<p>Chest Pain Type (0-3)</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">type</div>	<p>Resting Blood Pressure</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">mm Hg</div>
<p>Serum Cholesterol</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">mg/dL</div>	<p>Fasting Blood Sugar (1 if > 120 mg/dL)</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">0 or 1</div>
<p>Resting ECG Result</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">0-2</div>	<p>Maximum Heart Rate</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">bpm</div>
<p>Exercise-Induced Angina (1 = Yes)</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">0 or 1</div>	<p>ST Depression</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">mm</div>
<p>Slope of Peak Exercise ST Segment</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">0-2</div>	<p>Number of Major Vessels Colored by Fluoroscopy</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">0-3</div>
<p>Thalassemia (0-3)</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">0-3</div>	
Submit	

The parameters used are;

Age:

- The age of the patient. Age is a significant risk factor for heart disease, with risk increasing as people get older.

Sex:

- The biological gender of the patient (male or female). Men are generally at higher risk for heart disease at an earlier age compared to women.

CP (Chest Pain Type):

- The type of chest pain experienced by the patient. Common categories include typical angina, atypical angina, non-anginal pain, and asymptomatic. This helps in assessing the likelihood of coronary artery disease.

Trestbps (Resting Blood Pressure):

- The blood pressure is measured while the patient is at rest. High resting blood pressure (hypertension) is a major risk factor for heart disease.

Chol (Serum Cholesterol):

- The total cholesterol level in the blood, including both LDL (bad cholesterol) and HDL (good cholesterol). High cholesterol levels can lead to plaque buildup in arteries, increasing heart disease risk.

FBS (Fasting Blood Sugar):

- The blood glucose level after fasting for at least 8 hours. A value above 126 mg/dL indicates diabetes, which is a risk factor for heart disease.

Restecg (Resting Electrocardiographic Results):

- The results of an electrocardiogram taken while the patient is at rest. This can show abnormalities such as arrhythmias, past heart attacks, or other heart conditions.

Thalach (Maximum Heart Rate Achieved):

- The highest heart rate achieved during a stress test or exercise. This helps assess cardiovascular fitness and the heart's response to physical stress.

Exang (Exercise-Induced Angina):

- Indicates whether the patient experiences angina (chest pain) induced by exercise. It's a sign of coronary artery disease.

Oldpeak:

- ST depression on an ECG during peak exercise relative to rest. This can indicate the severity of ischemia (reduced blood flow to the heart muscle).

Slope:

- The slope of the peak exercise ST segment on an ECG, which can be upsloping, flat, or downsloping. It provides information on the presence and severity of ischemia.

CA (Number of Major Vessels):

- The number of major coronary arteries (0-3) that are colored by fluoroscopy. Higher numbers can indicate more severe coronary artery disease.

Thal (Thalassemia):

- A genetic blood disorder affecting hemoglobin production. In the context of heart disease, it can be used to indicate the presence of normal blood flow, fixed defect, or reversible defect as assessed by certain tests like thallium stress test.

When the data is sent, the results can be seen in the forms of



The system is based on Logistic Regression.

Accuracy is 84%

Test Train ratio is 0.3

Python has been used for the machine learning code implementation.

Functionalities of the system

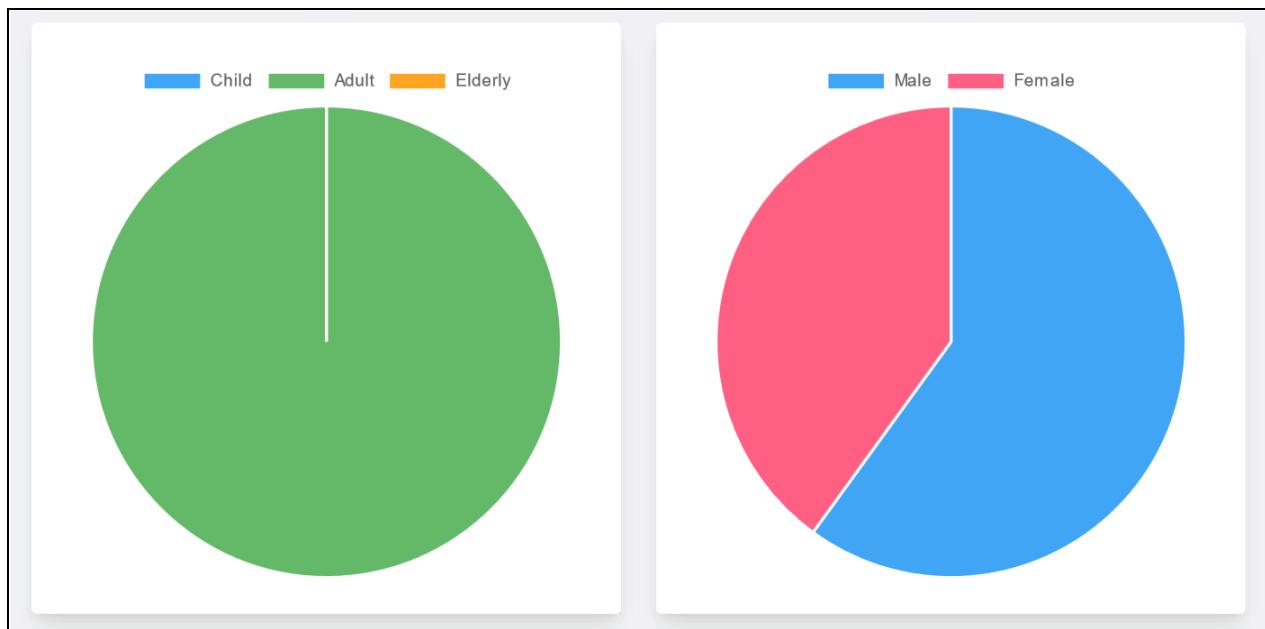
The system has different functionalities for each user.

❖ Admin Functionalities

- Admin can see how many patients , Doctors,Appointments and Total invoices are there within the system.

Total Patients 5	Total Doctors 5	Appointments 0	Total Invoice 0
---------------------	--------------------	-------------------	--------------------

- Admin is able to view the age groups and gender groups of patients



- Admin is able to get the details of patients with their name,NIC or their PatientID

Patients		
Search by First Name, Last Name or NIC		
Patient ID	Name	NIC
1	Kumara Perera	2002342234587v
2	Lakshman Wijesundara	200160143466v
3	Samanthi Rajakaruna	199847154559v
4	Kusal Mendis	199641223445v
5	Sarala Minoli	200432457789v

- Admin is able to get the details of Doctors with their name,NIC or their DoctorID

Doctors	
Search by Doctor ID, Surname, Last Name, or NIC	
Doctor ID	Name
1	Dr. Anderew Rodrigue
2	Dr. Sidath Samaratunge
3	Dr. Harshika Perera
4	Dr. Mukesh Sudarman
5	Dr. Herathi Samanpriya

- Admin is able to register a patient to a certain doctor

Register Patient to Doctor

<p>Select Doctor:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">Doctors</div> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>Search by Doctor ID, Surname, Last Name, or NIC</p> <table border="1" style="width: 100%; border-collapse: collapse; font-size: 0.8em;"> <thead> <tr> <th>Doctor ID</th> <th>Name</th> </tr> </thead> <tbody> <tr><td>1</td><td>Dr. Andrew Rodrigue</td></tr> <tr><td>2</td><td>Dr. Sidath Samarayake</td></tr> <tr><td>3</td><td>Dr. Harshika Perera</td></tr> <tr><td>4</td><td>Dr. Mukesh Sudarman</td></tr> <tr><td>5</td><td>Dr. Herathi Samanpriya</td></tr> </tbody> </table> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">Selected Doctor ID</div>	Doctor ID	Name	1	Dr. Andrew Rodrigue	2	Dr. Sidath Samarayake	3	Dr. Harshika Perera	4	Dr. Mukesh Sudarman	5	Dr. Herathi Samanpriya	<p>Select Patient:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;">Patients</div> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>Search by First Name, Last Name or NIC</p> <table border="1" style="width: 100%; border-collapse: collapse; font-size: 0.8em;"> <thead> <tr> <th>Patient ID</th> <th>Name</th> <th>NIC</th> </tr> </thead> <tbody> <tr><td>1</td><td>Kumara Perera</td><td>2002342234587v</td></tr> <tr><td>2</td><td>Lakshman Wijesundara</td><td>200160143466v</td></tr> <tr><td>3</td><td>Samanthi Rajakaruna</td><td>199847154559v</td></tr> <tr><td>4</td><td>Kusal Mendis</td><td>199641223445v</td></tr> <tr><td>5</td><td>Sarala Minoli</td><td>200432457789v</td></tr> </tbody> </table> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">Selected Patient ID</div>	Patient ID	Name	NIC	1	Kumara Perera	2002342234587v	2	Lakshman Wijesundara	200160143466v	3	Samanthi Rajakaruna	199847154559v	4	Kusal Mendis	199641223445v	5	Sarala Minoli	200432457789v
Doctor ID	Name																														
1	Dr. Andrew Rodrigue																														
2	Dr. Sidath Samarayake																														
3	Dr. Harshika Perera																														
4	Dr. Mukesh Sudarman																														
5	Dr. Herathi Samanpriya																														
Patient ID	Name	NIC																													
1	Kumara Perera	2002342234587v																													
2	Lakshman Wijesundara	200160143466v																													
3	Samanthi Rajakaruna	199847154559v																													
4	Kusal Mendis	199641223445v																													
5	Sarala Minoli	200432457789v																													
<input style="background-color: #007bff; color: white; border: none; padding: 5px 10px; border-radius: 5px; width: fit-content;" type="button" value="Register Patient to Doctor"/>																															

- Admin can add , update or delete a patient record

Click the patient to delete or update

Patient ID	NIC	First Name	Last Name	Gender	Date of Birth
1	2002342234587v	Kumara	Perera	Male	1/1/2003
2	200160143466v	Lakshman	Wijesundara	Male	8/7/2001
3	199847154559v	Samanthi	Rajakaruna	Female	4/19/1998
4	199641223445v	Kusal	Mendis	Male	12/3/1996
5	200432457789v	Sarala	Minoli	Female	4/24/2001

Previous
Page 1 of 1
Next

Patient Information

<p>Patient ID: 1 NIC: 2002342234587v Name: Kumara Perera</p>	<p>Gender: Male Date of Birth: 1/1/2003 Email: N/A</p>
<input style="background-color: #007bff; color: white; border: none; padding: 5px 10px; border-radius: 5px; width: fit-content;" type="button" value="Update Details"/>	<input style="background-color: #dc3545; color: white; border: none; padding: 5px 10px; border-radius: 5px; width: fit-content;" type="button" value="Delete Patient"/>

Patient Information

Update Patient Information

First Name Kumara	Last Name Perera
NIC 2002342234587v	Gender Male
Date of Birth 01 / 01 / 2003	Email

Update Patient **Cancel**

Patient Information

Patient ID: 1
NIC: 2002342234587v
Name: Kumara Perera

Gender: Male
Date of Birth: 1/1/2003
Email: N/A

Update Details **Delete Patient**

localhost:5173

Are you sure you want to delete this patient?

OK **Cancel**

➤ Admin is able to view patient's medical details

Patient Symptoms Info

Bilateral Lower Limb Swelling Yes	Dyspnoea No	Orthopnoea Yes
Paroxysmal Nocturnal Dyspnoea No	Fatigue Yes	Doctor Recommendation Increase fluid intake and monitor symptoms.
Symptom Date 7/10/2024		

Patient Clinical Data

Diagnosis of Heart Disease Yes	Presence of Anemia No	Creatinine Phosphokinase 200
Diabetes Yes	Ejection Fraction 60%	Blood Pressure 110
Platelets 260000	Serum Creatinine 1	Serum Sodium 250
Smoking No	Follow-Up Period (Days) 30	

Examination Data		
Tachycardia at Rest No	Hypotension Yes	Raised Jugular Venous Pressure Yes
Displaced Apex Beat No	Right Ventricular Heave Yes	Pleural Effusion No
Hepatomegaly Yes	Gallop Rhythm on Auscultation No	Pedal and Ankle Oedema No
Tachypnoea Yes	Ascites No	Examination Date 8/15/2024

- Admin is able to add, update or delete a doctor's record

Doctors List					Add Doctor
Doctor ID	NIC	Surname	Last Name	Actions	
1	852741968V	Anderew	Rodrigue	<button>Update</button>	<button>Delete</button>
2	199418395738V	Sidath	Samarayake	<button>Update</button>	<button>Delete</button>
3	198945271324V	Harshika	Perera	<button>Update</button>	<button>Delete</button>
4	872423158V	Mukesh	Sudarman	<button>Update</button>	<button>Delete</button>
5	8527419856V	Herathi	Samanpriya	<button>Update</button>	<button>Delete</button>

Previous Page 1 of 1 Next

❖ Doctor Functionalities

- Doctors can view all the patients but cannot access all the patient data.

Patients

Search by First Name, Last Name or NIC

Patient ID	Name	NIC
1	Kumara Perera	2002342234587v
2	Lakshman Wijesundara	200160143466v
3	Samanthi Rajakaruna	199847154559v
4	Kusal Mendis	199641223445v
5	Sarala Minoli	200432457789v

- Doctors can access only patients registered under their ID

Patients Registered to Doctor

Patient ID	First Name	Last Name	Gender	Date of Birth
2	Lakshman	Wijesundara	Male	8/7/2001
5	Sarala	Minoli	Female	4/24/2001
4	Kusal	Mendis	Male	12/3/1996
1	Kumara	Perera	Male	1/1/2003
3	Samanthi	Rajakaruna	Female	4/19/1998

- Doctors can access the data of the patients , add new records, change patient data and delete records. Also, doctors can access previous interactions through the systems

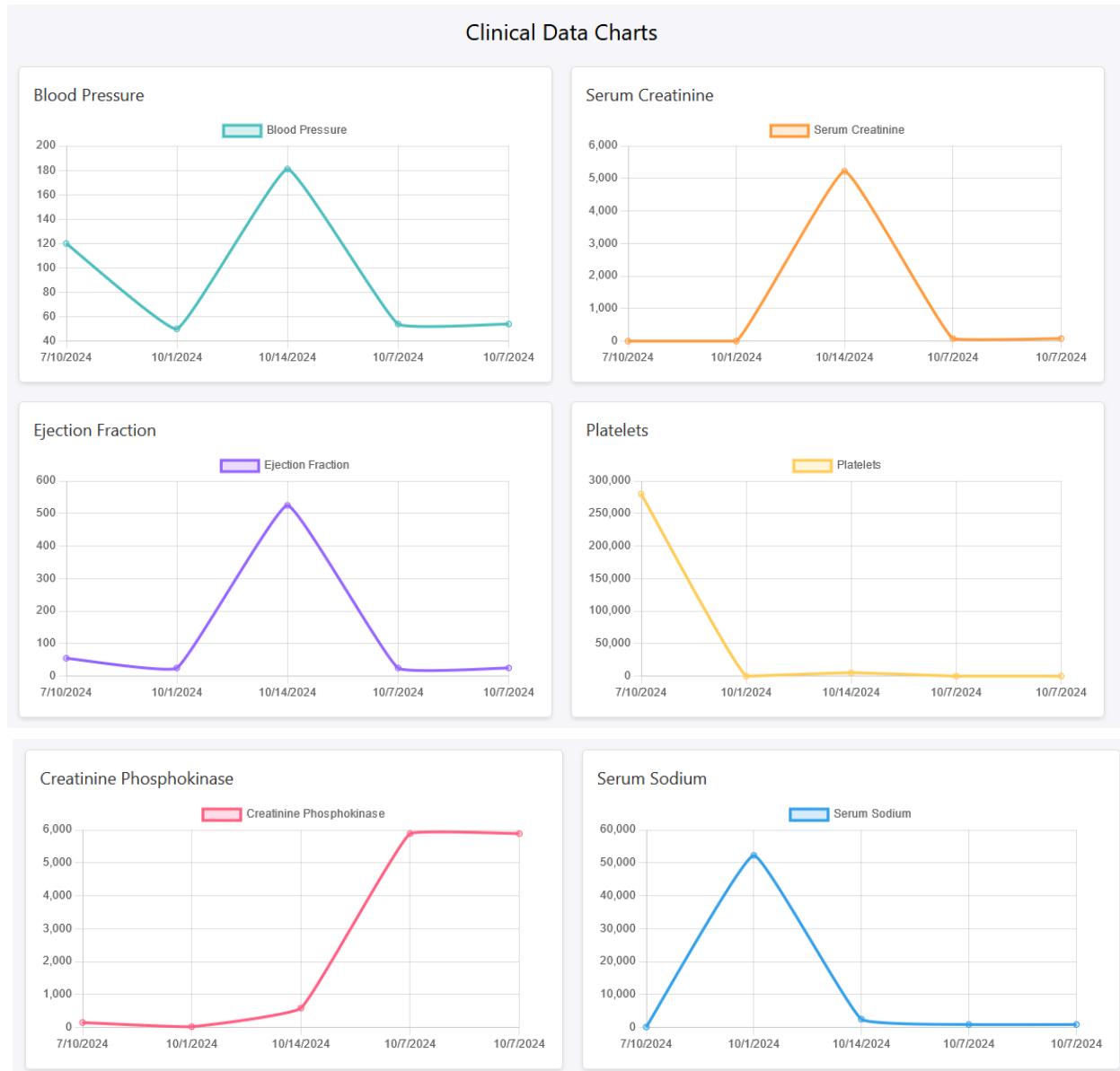
Patient Symptom Data

Patient ID : 2

Add Record

Date(M/D/Y)	Details	Doctor Recommendation
10/18/2024	Show Details	Good
8/15/2024	Show Details	Limit salt intake, elevate legs while resting, and follow up in one week.

➤ Doctors can see visual data representation through charts and graphs



- Doctors can access past Clinical data and admissions information

Patient Clinical Data		Patient ID : 2
Date(M/D/Y)	Details	
10/14/2024	Hide Details	
Diagnosis of Heart Disease: Yes	Blood Pressure: 181	
Presence of Anemia: Yes	Platelets: 5252	
Creatinine Phosphokinase: 589	Serum Creatinine: 5221	
Diabetes: No	Serum Sodium: 2523	
Ejection Fraction: 525%	Smoking: No	
	Follow-Up Period (Days): 25	
	Update Delete	
10/7/2024	Show Details	
10/7/2024	Show Details	
10/1/2024	Show Details	
7/10/2024	Show Details	

- Doctors can access past Examination Data and Add new records

Patient Examination Data		Patient ID : 2
Date (M/D/Y)	Details	
10/30/2024	Hide Details	
Tachycardia at Rest: No	Pedal and Ankle Oedema: Yes	
Hypotension: No	Gallop Rhythm: No	
Raised Jugular Venous Pressure: Yes	Tachypnoea: Yes	
Displaced Apex Beat: No	Ascites: No	
	Update Delete	
10/14/2024	Show Details	
7/1/2024	Show Details	

➤ Doctors can update the data through forms

Patient Details Submission Form

Diagnosis of Heart Disease

Presence of Anemia

Creatinine Phosphokinase:

Diabetes

Ejection Fraction:

Blood Pressure:

Platelets:

Serum Creatinine:

Serum Sodium:

Doctor Examination Submission Form

Tachycardia at Rest

Hypotension

Raised Jugular Venous Pressure

Displaced Apex Beat

Pedal and Ankle Oedema

Gallop Rhythm

Tachypnoea

Ascites

Examination Date:

➤ Doctors can use prediction system to enhance their medical opinions

The Prediction System - Evaluation of Patients

Enter your age	Sex (0 = Female, 1 = Male)
<input type="text" value="years"/> years	<input type="text" value="0 or 1"/> 0 or 1
<input type="text" value="years"/> years	<input type="text" value="0 or 1"/> 0 or 1
Chest Pain Type (0-3)	
<input type="text" value="type"/> type	<input type="text" value="type"/> type
Resting Blood Pressure	
<input type="text" value="mm Hg"/> mm Hg	<input type="text" value="mm Hg"/> mm Hg
Serum Cholesterol	
<input type="text" value="mg/dL"/> mg/dL	<input type="text" value="mg/dL"/> mg/dL
Fasting Blood Sugar (1 if > 120 mg/dL)	
<input type="text" value="0 or 1"/> 0 or 1	<input type="text" value="0 or 1"/> 0 or 1
Resting ECG Result	
<input type="text" value="0-2"/> 0-2	<input type="text" value="0-2"/> 0-2
Maximum Heart Rate	
<input type="text" value="bpm"/> bpm	<input type="text" value="bpm"/> bpm
Exercise-Induced Angina (1 = Yes)	
<input type="text" value="0 or 1"/> 0 or 1	<input type="text" value="0 or 1"/> 0 or 1
ST Depression	
<input type="text" value="mm"/> mm	<input type="text" value="mm"/> mm
Slope of Peak Exercise ST Segment	
<input type="text" value="0-2"/> 0-2	<input type="text" value="0-2"/> 0-2
Number of Major Vessels Colored by Fluoroscopy	
<input type="text" value="0-3"/> 0-3	<input type="text" value="0-3"/> 0-3
Thalassemia (0-3)	
<input type="text" value="0-3"/> 0-3	<input type="text" value="0-3"/> 0-3

❖ Patient Functionalities

- Patient Can edit some of his data [ex: Email , Phone number]



- Patients can view their personal data

A screenshot of a patient profile page displaying detailed personal information. The page is divided into two main sections: "Patient Info" and "General Info".

Patient Info	General Info
First Name Lakshman	Gender Male
Last Name Wijesundara	Age 23 years old
NIC 200160143466v	Date of Birth 2001-08-07
Email N/A	Occupation Software Engineer
	Insurance HealthPlus

- Patient are provided their next appointment / Clinic day

A screenshot of a patient interface. On the left, a box displays "Days Until Next Clinic Day" with the text "-11 Days" in green. On the right, a larger box titled "Select a Date" shows a calendar for October 2024. The calendar highlights the current date as Friday, October 11, 2024. The text "Selected Date: Fri Oct 11 2024" is displayed at the bottom of the calendar box.

Select a Date						
		October 2024				
		MON	TUE	WED	THU	FRI
«		30	1	2	3	4
<		7	8	9	10	11
>		14	15	16	17	18
»		21	22	23	24	25
		28	29	30	31	1
					2	3

Selected Date: Fri Oct 11 2024

➤ Patients can view their medical data

Patient Symptoms Info

Bilateral Lower Limb Swelling No	Dyspnoea Yes	Orthopnoea No	Paroxysmal Nocturnal Dyspnoea Yes
Fatigue No	Symptom Date 8/15/2024	Doctor Recommendation Limit salt intake, elevate legs while resting, and follow up in one week.	

Patient Clinical Data

Diagnosis of Heart Disease No	Presence of Anemia Yes	Creatinine Phosphokinase 150	Diabetes No
Ejection Fraction 55%	Blood Pressure 120	Platelets 280000	Serum Creatinine N/A
Serum Sodium 145	Smoking Yes	Follow-Up Period (Days) 45	

Examination Data

Tachycardia at Rest Yes	Hypotension No	Raised Jugular Venous Pressure Yes	Displaced Apex Beat No
Right Ventricular Heave No	Pleural Effusion Yes	Hepatomegaly No	Gallop Rhythm on Auscultation Yes
Pedal and Ankle Oedema Yes	Tachypnoea No	Ascites No	Examination Date 7/1/2024

➤ Clinical data overview is provided

Patient Clinical Data Overview

Diagnosis of Heart Disease:
Yes
Indicates a heart condition diagnosis which impacts overall health management.

Presence of Anemia:
No
The patient does not show signs of anemia, which affects oxygen transport in the body.

Diabetes:
Yes
The patient has diabetes, a condition that requires careful blood sugar management.

Clinical Marker	Normal Range	Elevated Range	Critical Range	Meaning
Blood Pressure	90-120 mmHg	121-139 mmHg	> 140 mmHg	High BP can increase heart disease risks.
Serum Creatinine (mg/dL)	0.6-1.2 mg/dL	1.3-1.5 mg/dL	> 1.5 mg/dL	Elevated levels indicate possible kidney issues.
Platelets	150,000-450,000 cells/mcL	450,000-500,000 cells/mcL	> 500,000 cells/mcL	High levels can lead to clotting issues.

Clinical Data Trends

- Patients are given their scheduled appointment days if they are inclined. They can reach out through the services page if required.

Patient Symptoms Info

Symptom Date	Details
7/10/2024	Show Details
6/20/2024	Show Details
8/5/2024	Show Details
5/25/2024	Show Details
5/25/2024	Show Details

Patient Clinical Data

Date	Details
8/10/2024	Show Details
6/15/2024	Show Details
8/1/2024	Show Details

Patient Dashboard

Nimal Silva
Active
 Patient ID: 1

[Edit Data](#)

Patient Info

First Name	Nimal
Last Name	Silva
NIC	1987432165890v
Email	e20266@eng.pdn.ac.lk

General Info

Gender	Male
Age	26 years old
Date of Birth	1998-05-12
Occupation	Software Engineer
Insurance	HealthPlus

Testing and Error Checking

The testing on the system has been done in different phases.

❖ Unit Testing

- Unit testing has been carried out for every component
 - Using VS code component Preview for React
 - Postman for API and data sending
 - Functional testing on separate functions

A screenshot of the Postman application interface. The main window shows a collection of API requests. One request is selected: a GET request to `http://localhost:8080/api/admin/doctor/1`. The 'Params' tab is active, showing a single parameter 'Key' with value 'Value'. Below the request details, a test runner window is open, displaying a list of test cases for 'PatientRepositoryTests' from 'com.example.MediguardBack'. All tests are marked as successful (green checkmarks) with execution times: 436 ms, 409 ms, 14 ms, 8 ms, and 5 ms respectively.

Two screenshots of a mobile application interface. The left screen is titled 'Doctor Data' and contains a text input field labeled 'Enter Doctor ID' and a blue button labeled 'Get Doctor Data'. Below the button is the message 'No doctor data available.' The right screen is titled 'Patient Symptoms Data' and contains a text input field labeled 'Enter Patient ID' and a blue button labeled 'Get Patient Data'. Below the button is the message 'No patient data available.'

❖ Integration Testing

- After unit testing, the components and working status are tested as a whole unit.
- The functional tests are also done for integration testing.

❖ Regression testing

- The system is checked when new code is inserted for correct implementation

❖ Confusion Matrix

- The machine learning code is expected to be tested for better performance
- Therefore, the confusion matrix is used as method to find the accuracy of the machine learning model

Measure	Value	Derivations	
Sensitivity	0.8261	$TPR = TP / (TP + FN)$	
Specificity	0.8537	$SPC = TN / (FP + TN)$	
Precision	0.8261	$PPV = TP / (TP + FP)$	
Negative Predictive Value	0.8537	$NPV = TN / (TN + FN)$	
False Positive Rate	0.1463	$FPR = FP / (FP + TN)$	
False Discovery Rate	0.1739	$FDR = FP / (FP + TP)$	
False Negative Rate	0.1739	$FNR = FN / (FN + TP)$	
Accuracy	0.8411	$ACC = (TP + TN) / (P + N)$	
F1 Score	0.8261	$F1 = 2TP / (2TP + FP + FN)$	
Matthews Correlation Coefficient	0.6797	$T P \cdot T N - F P \cdot F N / \sqrt{((T P + F P) \cdot (T P + F N) \cdot (T N + F P) \cdot (T N + F N))}$	

	Training Set			
	TARGET	Class0	Class1	SUM
OUTPUT				
Sensitivity Sensitivity or True Positive Rate (TPR)	57	12	69	
Specificity Specificity (SPC) or True Negative Rate (TNR)	37.75%	7.95%	82.61% 17.39%	
Precision Precision or Positive Predictive Value (PPV)				
Negative Predictive Value Negative Predictive Value (NPV)				
False Positive Rate Fall-out or False Positive Rate (FPR)	Class0	37.75%	7.95%	82.61% 17.39%
False Discovery Rate False Discovery Rate (FDR)	Class1	12	70	82
False Negative Rate Miss Rate or False Negative Rate (FNR)		7.95%	46.36%	85.37% 14.63%
Accuracy Accuracy (ACC)	SUM	69	82	127 / 151
F1 Score F1 Score (F1)		82.61% 17.39%	85.37% 14.63%	84.11% 15.89%
Matthews Correlation Coefficient Matthews Correlation Coefficient (MCC)				

- ❖ Try and Catch blocks for exceptions
 - Typical implementation of Try and catch blocks to avoid system failures in a state of error



- ❖ Automated testing for logins and validations

A screenshot of a "Register" user interface form. The form fields and their validation errors are:

- First Name: Pathum (highlighted with a red border)
- Last Name: dissanayake (highlighted with a red border)
- National ID: ABCDEFG (highlighted with a red border)
- Date of Birth: mm / dd / yyyy (highlighted with a red border)
- I want to receive inspiration, marketing promotions, and updates via email: (checkbox highlighted with a red border)

The "Next" button at the bottom is blue.

- ❖ Using Doctest in Python to test the functionality of the created functions
- ❖ Also, final test phases have been implemented for every component
- ❖ Mocks , stabs and fakes have been used to check the functionality of system as well
- ❖ Glass Box testing and Black box testing methods have been used to verify the outputs when different Inputs are presented

✓	✓	PatientServiceTest (com.example.MediguardBackEnd. 966 ms
✓		testCreatePatient_Success() 939 ms
✓		testCreatePatient_DuplicateException() 9 ms
✓		testUpdatePatient_ResourceNotFoundException() 5 ms
✓		testGetPatientById_ResourceNotFoundException() 4 ms
✓		testUpdatePatient_Success() 3 ms
✓		testGetAllPatients() 4 ms
✓		testGetPatientById_Success() 2 ms

✓	✓	PatientControllerTest (com.example.MediguardBa 1 sec 187 ms
✓		testCreatePatient_Success() 1 sec 165 ms
✓		testCreatePatient_DuplicateException() 6 ms
✓		testUpdatePatient_Success() 5 ms
✓		testGetPatientById_Success() 3 ms
✓		testRegisterDoctorsToPatient_Success() 2 ms
✓		testCreatePatient_BindingResultErrors() 3 ms
✓		testDeletePatient_Success() 3 ms

✓	✓	PatientRepositoryTests (com.example.MediguardBack 436 ms
✓		PatientRepository_UpdatePatient_ReturnUpdatedPa 409 ms
✓		PatientRepository_DeletePatient_ReturnVoid() 14 ms
✓		PatientRepository_FindById_ReturnPatient() 8 ms
✓		PatientRepository_SaveAll_ReturnSavePatient() 5 ms

- ❖ In the development of **CardioGuard**, we conducted comprehensive testing to ensure system reliability and accuracy.
- ❖ **Unit testing** was performed on individual components to validate their correctness. Integration testing was carried out to confirm that components interacted seamlessly, ensuring a robust flow of data across the system. Additionally, manual testing was executed using **Postman** and **Postman Flows**, enabling us to simulate user operations and identify potential issues in API interactions.
- ❖ This combination of automated and manual testing ensured that the application met both functional and non-functional requirements.

PatientControllerTests

Test Method	Time
testCreatePatient_Success()	4 sec 431 ms
testCreatePatient_DuplicateException()	30 ms
testUpdatePatient_Success()	42 ms
testGetPatientById_Success()	28 ms
testRegisterDoctorsToPatient_Success()	24 ms
testCreatePatient_BindingResultErrors()	17 ms
testDeletePatient_Success()	14 ms

PatientServiceTests

Test Method	Time
testCreatePatient_Success()	1 sec 864 ms
testCreatePatient_DuplicateException()	19 ms
testUpdatePatient_ResourceNotFoundException()	9 ms
testGetPatientById_ResourceNotFoundException()	5 ms
testUpdatePatient_Success()	4 ms
testGetAllPatients()	7 ms
testGetPatientById_Success()	4 ms

PatientRepositoryTests

Test Method	Time
PatientRepository_UpdatePatient_ReturnUpdatedPatient()	738 ms
PatientRepository_DeletePatient_ReturnVoid()	31 ms
PatientRepository_FindById_ReturnPatient()	17 ms
PatientRepository_SaveAll_ReturnSavePatient()	10 ms

Tests passed: 4 of 4 tests – 796 ms

```

primary key (patient_id, doctor_id)
)
Hibernate:
    create table patient (
        patient_id bigint not null,
        date_of_birth DATE not null,
        email TEXT,
        first_name TEXT not null,
        gender TEXT not null,
        last_name TEXT not null,
        nic TEXT not null,
        user_id integer,
        primary key (patient_id)
    )

```



```
© PatientController.java © PatientControllerTest.java × © PatientRepositoryTests.java © AdminConfig.java ▾
1 package com.example.MediguardBackEnd.Controller; ⚠ 19 ^
2
3 > import ...
28
29 © public class PatientControllerTest { drpnilupul
30
31     @Mock 6 usages
32     private PatientService patientService;
33
34     @Mock 5 usages
35     private UserService userService;
36
37     @Mock 4 usages
38     private BindingResult bindingResult;
39
40     @InjectMocks 6 usages
41     private PatientController patientController;
42     @InjectMocks 1 usage
43     private AdminController adminController;
44
45     private ObjectMapper objectMapper; 1 usage
46
47     @BeforeEach drpnilupul
48     public void setUp() {
49         MockitoAnnotations.openMocks( testClass: this);
50         objectMapper = new ObjectMapper();
51     }
52 }
```

Project Management and Version Control

We employed **GitHub** as the central platform for project management and version control throughout the development of **CardioGuard**.

GitHub allowed us to manage source code collaboratively, track changes, and document our progress efficiently. Key functionalities, such as branching and pull requests, facilitated seamless teamwork and conflict resolution. Screenshots and links to the GitHub repository showcase our systematic approach to version control, while GitHub's issue tracking and commit history demonstrate adherence to best practices in project management.

The screenshot shows the GitHub 'Branches' page with a dark theme. At the top right is a green 'New branch' button. Below it is a navigation bar with tabs: 'Overview' (which is selected), 'Yours', 'Active', 'Stale', and 'All'. A search bar with placeholder text 'Search branches...' is positioned below the tabs. The page is divided into three main sections:

- Default:** Shows one branch, 'main', which was updated 'last month'. It has a 'Default' status indicator, a trash icon, and a more options icon.
- Your branches:** Lists four branches: 'Pradeep_20_10_24_PD', 'Pradeep_20_10_24', 'Pradeep_09_10_24', and 'Service-page-tabs'. Each branch entry includes the last update time ('last month'), a check status icon, a 'Behind | Ahead' status bar, a pull request link (#17, #13, #6), and a trash icon.
- Active branches:** Lists five branches: 'Tharindu/Oct/24', 'terms-and-condition', 'error-show-when-enter-username-password-in-resiteration', 'login-error-showing', and 'Pradeep_20_10_24_PD'. Similar to the 'Your branches' section, each entry shows the last update time, check status, behind/ahead status, pull request link, and a trash icon.

At the bottom center of the page is a 'View more branches >' link.

Branches						New branch
Overview	Yours	Active	Stale	All		
<input type="text"/> Search branches...						
Branch		Updated	Check status	Behind	Ahead	Pull request
Tharindu/Oct/24	View	last month		11	0	#21
terms-and-condition	View	last month		14	0	#20
error-show-when-enter-username-password-in-resiteration	View	last month		16	0	#19
login-error-showing	View	last month		18	0	#18
Pradeep_20_10_24_PD	View	last month		20	0	#17
Pradeep_20_10_24	View	last month		23	0	#16
20/Oct/Tharindu	View	last month		22	0	#16
Prediction-to-database	View	last month		32	0	#15

Code Quality and Security

❖ Code Quality

- For better code quality maintenance and better coding practice, we have implemented many techniques in this project

■ Readability

```
)  
@Column(name="patient_id",updateable = false)  
private Long patientId;  
6 usages  
@Column(name = "nic",nullable = false,columnDefinition = "  
private String nic;  
6 usages  
@Column(name = "first_name",nullable = false,columnDefinition = "  
private String firstName;  
6 usages  
@Column(name = "last_name",nullable = false,columnDefinition = "  
private String lastName;  
6 usages  
@Column(name = "gender",nullable = false,columnDefinition = "  
private String gender;  
6 usages
```

```
app = Flask(__name__)  
  
@app.route('/predict', methods=['POST'])  
def predict():  
    if request.method == 'POST':  
        data = request.get_json(force=True)  
        input_data = list(data['data'].values())  
        dataset = testing_prediction(input_data)  
        return jsonify(dataset)  
  
if __name__ == '__main__':  
    app.run(port=8000, debug=True)
```

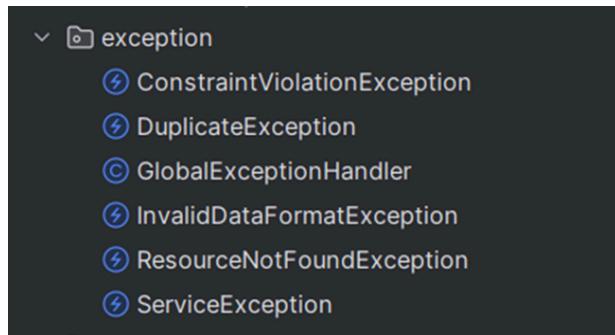
The coding has been done using extensions that allow better coding with appropriate indentation that allows excessive readability

■ Consistency

```
2 usages ▲ drpnilupul  
@Data  
public class LoginDTO {  
    1 usage  
    @NotBlank(message = "Username is required")  
    @Size(min = 4, message = "Username must be at least 4 ch")  
    private String username;  
  
    1 usage  
    @NotBlank(message = "Password is required")  
    @Size(min = 4, message = "Password must be at least 4 ch")  
    private String password;  
  
    no usages ▲ drpnilupul  
    public LoginDTO() {  
    }  
  
    no usages ▲ drpnilupul  
    public LoginDTO(String username, String password) {  
        this.username = username;  
        this.password = password;  
    }
```

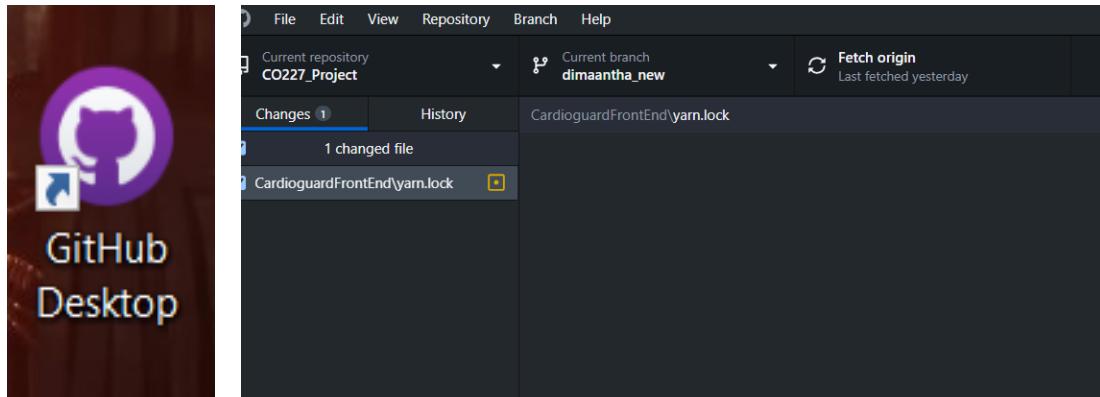
Codes are created in consistent manner implementing classes and using coding patterns

■ Error Handling



Errors are handled using exception handling and also input validations. Therefore, the system continues even if errors occur handling them without causing errors in the output.

■ Refactoring



The work has been done in GitHub where it shows conflicts when merging code snippets allowing more careful coding and providing exceptions beforehand.

■ Comments

```
# Initialize Flask app
app = Flask(__name__)

# Enable CORS
CORS(app)

# Load and preprocess the dataset
# Ensure that the file path is correct and uses double backslashes for Windows or forward slashes.
heart_data = pd.read_csv('Prediction System/heart disease data.csv')
heart_data = heart_data.dropna().drop_duplicates()

# Separate features and target
X = heart_data.drop(columns='target', axis=1)
Y = heart_data['target']

# Split the data into training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, t3

    },
    {
        path: "/patient",
        element: <PatientMainLayout />, // Patient layout with sidebar and topbar
        children: [
            {
                path: "", // Patient dashboard
                element: <PatientDashboard />,
            },
            {
                path: "patients", // Patient medical info page
                element: <PatientMedicalInfo />,
            },
            // Add more patient routes here if necessary
        ],
    },
    {
        path: "/doctor",
        element: <DoctorMainLayout />, // Doctor layout with sidebar and topbar
        children: [
            {
                path: "patients", // Doctor medical info page
                element: <DoctorMedicalInfo />,
            },
            // Add more doctor routes here if necessary
        ],
    }
}
```

Commenting codes of importance allows for greater readability and allows for checking and reviewing codes easily

❖ Security

- For security of the system, many techniques have been implemented.

■ Automatic Validations

The screenshot shows a registration form with the following fields and validation errors:

- First Name: Pathum (highlighted in red)
- Last Name: dissanayake (highlighted in red)
- National ID: ABCDEFG (highlighted in red)
- Gender: Male, Female, Other (radio buttons)
- Date of Birth: mm / dd / yyyy (highlighted in red)
- I want to receive inspiration, marketing promotions, and updates via email:

Validation messages:
First Name is required.
Last Name is required.
National ID is required.
Gender is required.
Date of Birth is required.

Buttons: Next, Already have an account? Log In

■ Encoding

user_id	password	username
1	\$2a\$10\$Tzf87f8X529XPYPtRgOmh3ZaZ/CB/M3bgH8lrHIJ...	admin
2	\$2a\$10\$5oq0r8aCDjES3xCOJkdawuWYB82aX8SGZAfCIQsCt.u...	Kumara
3	\$2a\$10\$Ls6NWeluaYxl/3uX23ne/Ouv2ERc5TY7hhmgMPAi7G...	Lakshman
4	\$2a\$10\$J8v15rA6ESUsKOn8/GLHDOsyzGF0KjFeHN5G2Y0AFEd...	Samantha
5	\$2a\$10\$OQxS8DteZVKX9jONegOE0ITNuezm.eEeEGaMPNXRtl...	Andrew
6	\$2a\$10\$68.dNs97zOISaBLyRpq/rONm7xI1LenaQvlfXuobNXH...	Sidath
7	\$2a\$10\$Dg85j9OUG2XkFKicozXhXen1VAybjO7k2RbPCE0PnZk...	Kusal
8	\$2a\$10\$BKvtU/xRtAs6k7TsOs74POpYsmpQQAIx3X4uJt.D/aj...	Sarala
9	\$2a\$10\$hgih7dLw9u8NYjY.5LhfBuKRnK6vFTd1i0fcfJJdb8q...	Harshika
10	\$2a\$10\$pWidWDdfQdxwYqdW53P4ruLkRvLmGLrHGRYDGvv1hst...	smith
11	\$2a\$10\$EILHIfm3pdkBWbtMUh0dLONvmsngl5TKhljQTqF3qHA...	Herathi

■ Security measures within BackEnd implementation

▼ security

- © JwtAuthenticationEntryPoint
- © JWTAuthenticationFilter
- © JWTGenerator
- © SecurityConfig
- © SecurityConstants

- ❖ **JWT Authentication Entry Point:** The JWT Authentication Entry Point is the part of the web application that responds to unauthorized requests. It blocks all the requests that do not support proper JWT tokens and displays an error message, most frequently a 401 Unauthorized. You can have only authenticated users accessing the restricted resources.
- ❖ **JWT Authentication Filter:** The JWT Authentication Filter is a piece of software placed in between the HTTP address of a web service and the URL address itself and is used to examine each HTTP request for a valid JWT through the header Authorization. After a successful authorization process, it searches the token for the authentication information associated with the user, and it modifies the current session context accordingly to provide access to secured resources.
- ❖ **JWT Generator:** The JWT Generator is an accessory that allows generating JSON Web Tokens (JWT) for authenticated users. It stores user details and other incidents in the form of a token, which is encrypted with a secret key or a combination of public and private keys. Afterwards, this token can be used to authenticate the user and help protect certain resources within the web application.

```

@Configuration
@EnableWebSecurity
public class SecurityConfig {

    2 usages
    private final CustomUserDetailsService userDetailsService;
    2 usages
    private final JwtAuthenticationEntryPoint jwtAuthenticationEntryPoint;

    no usages drpnilupul
    @Autowired
    public SecurityConfig(CustomUserDetailsService customUserDetailsService, JwtAuthenticati
        this.userDetailsService = customUserDetailsService;
        this.jwtAuthenticationEntryPoint = jwtAuthenticationEntryPoint;
    }
}

```

These types of authentication have been used for security purposes.

■ Post Method usage

The screenshot shows two separate Postman requests:

- Request 1: POST /api/patients**
- Request 2: POST /api/patients/1/symptoms**

Request 1: POST /api/patients

Body (raw JSON):

```

1 {
2     "nic": "2002342234587V",
3     "firstName": "Kumaria",
4     "lastName": "Perera",
5     "gender": "Male",
6     "dateOfBirth": "2003-01-01"
7 }

```

Request 2: POST /api/patients/1/symptoms

Body (raw JSON):

```

1 {
2     "bilateralLowerLimbSwelling": true,
3     "dyspnoea": false,
4     "orthopnoea": true,
5     "paroxysmalNocturnalDyspnoea": false,
6     "fatigue": true,
7     "doctorRecommendation": "Increase fluid intake and monitor symptoms.",
8     "symptomDate": "2024-07-10T10:00:00Z"
9 }
10

```

Post method to hide data through the URLs

FeedBack

During the development of **CardioGuard**, the following suggestions and feedback were received from supervisors to enhance the overall system functionality and user experience:

1. UI Improvements:

The supervisors suggested revisiting the user interface to ensure a more user-friendly and visually appealing design that enhances the usability of the application.

2. Security Enhancements:

Emphasis was placed on strengthening the system's security measures, particularly for handling sensitive patient data and ensuring robust protection against unauthorized access.

3. Patient Data Accessibility:

It was recommended to improve the flow and accessibility of patient data for authorized users, ensuring that the system is intuitive and efficient for medical professionals.

4. Integration of Google Login:

The addition of Google login functionality was proposed to simplify the authentication process and provide users with a convenient and secure way to access the system.

5. Email Notifications:

Supervisors advised implementing an email notification system to inform users about important events, such as appointment confirmations or updates to their medical records.

6. Password and Username Validation:

Strengthening the validation process for usernames and passwords was suggested to ensure better user security and compliance with best practices.

7. Doctor-Specific Functionality:

Suggestions were made to fine-tune the features specifically designed for doctors, such as patient record management, to enhance their workflow and ease of use.

Challenges

During the project, the team encountered several significant challenges, which required strategic approaches to resolve:

- **Integration Issues:** Ensuring smooth integration between the machine learning models, backend systems, and frontend interface was particularly challenging. Discrepancies in data formats, communication protocols, and system dependencies led to delays. The team overcame these issues by conducting regular debugging sessions, maintaining clear documentation, and implementing iterative testing to identify and resolve compatibility problems.
- **Time Management:** Balancing the demands of academic schedules and project deadlines proved to be a common challenge for the team members. To address this, they created a detailed project timeline, divided tasks based on individual strengths, and regularly tracked progress to stay on schedule.
- **Technical Complexities:** Developing and fine-tuning machine learning algorithms to achieve the desired accuracy and efficiency required extensive research and testing. Similarly, optimizing backend processes for performance and scalability presented significant hurdles. These challenges were tackled by consulting academic resources, attending workshops, and engaging in peer discussions to find effective solutions.
- **Cross-functional Coordination:** Since each team member specialized in different aspects of the project, ensuring clear and consistent communication was critical. Initial misunderstandings about roles and deliverables created delays. The team adopted collaborative tools, such as project management software and shared documentation platforms, and scheduled regular status meetings to improve coordination and ensure alignment.
- **Adapting to Technology:** The technology vs practise came into picture as we were determined to use new technologies instead of what we were used to. Due to that, it was challenging to find out some technology that would correspond to the requirements in the system.

By addressing these challenges proactively and fostering a collaborative team environment, the group was able to complete the project successfully, ensuring both technical and operational goals were met.

Work division and individual contribution

Work Task Division Among Project Members

The project was systematically divided into tasks to ensure streamlined progress and effective collaboration. Each member or subgroup was assigned responsibilities based on their expertise, ensuring balanced workload distribution and efficient completion of project milestones. The **machine learning component** of the project was primarily developed by Pathum Dissanayake (E/20/084). The **backend development** was handled by Tharindu Lakshan (E/20/262) and Pradeep Nilupul (E/20/266), while the **frontend responsibilities** were managed by Kavindu Methpura (E/20/254) and Dimantha Thilakasiri (E/20/397).

The following breakdown outlines the key areas of responsibility:

1. Pathum Dissanayake (E/20/084): Machine Learning Leads

- o Took the primary responsibility for the machine learning component of the project.
- o Implemented and optimized machine learning algorithms, ensuring accurate visual data processing and integration with the overall system.
- o Facilitated the incorporation of machine learning outputs into the front-end for enhanced user interaction.

2. Tharindu Lakshan (E/20/262) and Pradeep Nilupul (E/20/266): Backend Development Leads

- o Designed and developed the backend architecture to ensure robust system functionality.
- o Integrated authentication mechanisms and ensured secure data handling across the system.
- o Finalized the backend functionalities and addressed technical feedback from the engineering team and supervisors.
- o Worked on integrating pre-built components into the system and maintaining backend compatibility with other project modules.

3. Kavindu Methpura (E/20/254) and Dimantha Thilakasiri (E/20/397): Front-End Development Leads

- o Led the development of the user interface, focusing on creating visually appealing and intuitive dashboards.
- o Added dynamic components to pages and implemented navigation features using API keys.
- o Incorporated articles and other visual elements into the front end to enhance user experience.
- o Conducted thorough testing and ensured seamless integration of frontend and backend components.

This structured division of labor and specialization allowed the team to leverage individual strengths, ensuring a high-quality, cohesive final product.

While fulfilling their respective responsibilities, everyone also contributed to other tasks within the project.

additionally,

Pathum Dissanayake

- **APIs and Integration:** Delivered a fully functional API layer, forming the backbone of system interaction.
- **Front-End Development:** Developed front-end components in React and worked on concurrent system execution.

Pradeep Nilupul and Tharindu Nanayakkara

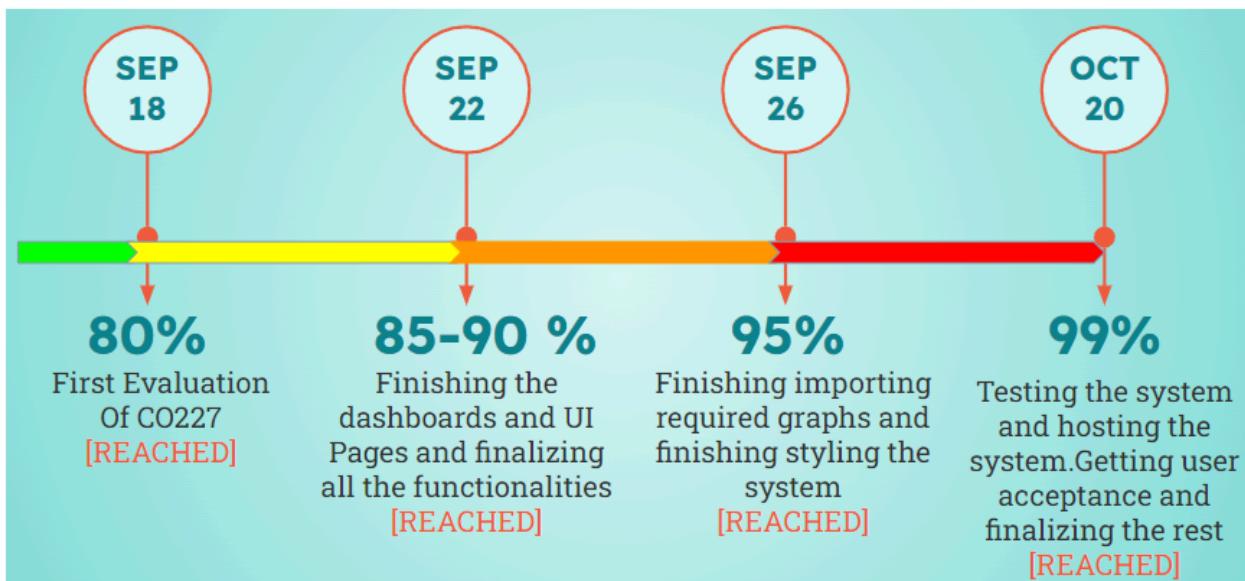
- **User Dashboards:** Built user-specific dashboards and added authentication for secure access.
- **Styling and Documentation:** Styled the front end and created comprehensive system documentation.
- **Adaptation:** Adjusted the system features based on expert advice from doctors and supervisors.

Kavindu Methpura and Dimantha

- **Testing and Hosting:** Worked on testing and hosting strategies for deployment readiness.
- **Service Pages:** Focused on developing service pages and documenting contributions.
- **Help to develop machine learning part as well.**

TimeLine

The strategic timeline and milestones were created amongst the members to ensure the completion of the project.



We have fully created the system implementation for the 2nd year project.

Project was completed by dividing the work among ourselves. Each member of the group contributed for the accomplishment of the project.

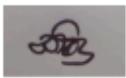
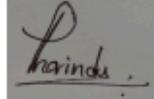
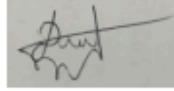
Around 10 meetings were held with the advisors and daily meetings were taken to discuss the workloads and shifts in the project within the group members.

Declaration

We , members of the Grroup 03 of the CO227 second year project ,

- E/20/084 - Dissanayake P.D.
- E/20/254 - Methpura S.K.P.
- E/20/262 - Nanayakkara A.T.L.
- E/20/266 - Nilupul D.R.P
- E/20/397 - Thilakasiri P.D.

hereby declare that all the information in the report is accurate to the best of your knowledge.

E.No	Name	Signature
E/20/084	Dissanayake P.D	
E/20/254	Methpura S.K.P	
E/20/262	Nanayakkara A.T.L	
E/20/266	Nilupul D.R.P	
E/20/397	Thilakasiri P.D	

CARDIO GUARD

Your health, our priority. Compassionate care, innovative solutions—because your well-being is the heart of our mission.

CO227 @ 2024

www.CardioGuardUOP.com

CONTACT US
+94 71 075 8542
MediGuardUOP@gmail.com

CardioGuard
“Where Technology meets Care”