[Web Application]

Cardio-Vision

It's Your Way To a Healthier Life

HELWAN UNIVERSITY
Faculty of Computers and Artificial Intelligence
Medical Informatics Program

[Cardio Vision]

A graduation project dissertation by:

Ahmed Ashour Ibrahim	20208016
Abdelrahman Hossam Eldin	20208147
Maryam Ashraf Nasr Eldin	20208218
Omnia Magdy Abdelsalam	20208242
Alaa Atef Elkaradawy	20208266

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computers & Artificial Intelligence, at Medical Informatics Program, the Faculty of Computers & Artificial Intelligence, Helwan University

Supervised by:

Dr. Manal Abdelkader

July 2023

جامعة حلوان
كلية الحاسبات والذكاء الإصطناعي
برنامج المعلوماتية الطبية

[Cardio Vision]

رسالة مشروع تخرج مقدمة من:

احمد عاشور ابراهيم	20208016
عبدالرحمن حسام الدين	20208147
مريم اشرف نصر الدين	20208218
امنية مجدى عبدالسلام	20208042
الاء عاطف القرضاوى	20208066

رسالة مقدمة ضمن متطلبات الحصول على درجة البكالوريوس في الحاسبات والذكاء الإصطناعي ، ببرنامج المعلوماتية الطبية كلية الحاسبات والذكاء الإصطناعي، جامعة حلوان

تحت إشراف:

د. منال عبد القادر

2023 يوليو

Acknowledgements

We express our deepest gratitude to **Dr. Manal Abdel-Kader** for her unwavering support, exceptional guidance, and continuous motivation throughout our graduation project. We are sincerely thankful to **Engineer. Amr Essam** for his exceptional support and assistance, which surpassed our expectations. We are also grateful to **Dr. Samar Mahmoud** for her valuable efforts in assisting us. We extend our profound appreciation to our families, particularly our parents, for their unwavering encouragement, patience, and assistance. The faculty of Computer Science and Artificial Intelligence at Helwan University has provided a conducive educational environment that has shaped us into competent graduates. We acknowledge the contributions of our friends, classmates, and all those who have supported us. Their assistance and encouragement have been invaluable. Thank you to everyone who has played a part in our journey.

Table of Contents

List of Tables

List of Figures

Abstract

Main Idea:

The CardioVision application is a cutting-edge software solution designed to revolutionize cardiovascular health monitoring. This program can predict ischemic heart disease before the patient complains of symptoms so that we can save more lives, as (IHD) is one of the deadliest diseases in the world, and its symptoms are rarely discovered early, resulting in a large number of deaths. So, what are we prepared to do? -> is to train algorithms to predict the occurrence percentage of the disease, as well as where the patient stands in terms of whether his case is mild, moderate, or severe.

Frontend Technologies:

- React: JavaScript library for building user interfaces.
- React Loading: displaying loading spinners and animations.
- Input Validation: for validating user input and preventing errors.
- React Notification: for displaying real-time notifications to user.
- Axios: making HTTP requests to the backend APIs.

Backend Technologies:

- Node.js
- Express.js
- Cloudinary: Image storage.
- JWT: JSON Web Token for user authentication.
- CORS: Handling cross-origin requests.
- Validation: Ensuring data integrity and input validation.
- Node Mailer: Sending email notifications.

Model Technologies:

- Django
- Preprocessing: OneHotEncoder algorithm
- Feature Selection: SelectKBest algorithm
- Models: MLP (Multi-Layer Perceptron) => Used One
- SVM (Support Vector Machine)- LR (Linear Regression)
- CNN (Conventional Neural Network) - GB (Gradient Boosting)

Process:

Register , Login, Forget Password , Dashboard , Profile

- First: you should register by entering your info (name, email, password,…)
- Then you can Login to the application by Email & Password after confirming your account through the link sent to the email.
- If you forget your password, you can request to reset your password a temporary code to use to change your password.
- **Admin** Can View Patients List , search for a patient ,Add Patient , Manage All Users , Set Admin , Delete User , view and edit the profile, delete the account and change the password.
- **User** can fill out the form, get the result ,edit form, view and edit the profile, delete the account and change the password.

Chapter 1: Introduction

1.1: Overview

Our main goal is to predict ischemic heart disease before the patient complains of symptoms so that we can save more lives, as (IHD) is one of the most deadly diseases in the world, and its symptoms are rarely discovered early, resulting in a large number of deaths. So, what are we prepared to do? -> is to train algorithms to predict the occurrence percentage of the disease, as well as where the patient stands in terms of whether his case is mild, moderate, or severe.

1.2: Objectives

1. Develop a machine learning model that can accurately detect the presence of heart disease based on a set of attributes such as age, blood pressure, cholesterol levels, and smoking status. The model should achieve an accuracy of at least 90%, and should be completed within six months.

2. Build a user-friendly web application that allows healthcare professionals to input patient data and receive an immediate risk assessment of heart disease. The application should be able to handle at least 100 users concurrently, and should be completed within six months.

1.3: Purpose

The main purpose of the website is to predict if a person has IHD or not once you enter the required attributes.

1.4 Scope:

Scope or range of the project means the work involved to finish our project; For Example…

1-Planning:

-We planned our project to be divided into subtasks and each task will be assigned to one of team's members to be responsible for.

We will start by initially designing our website using designing tool app (Adobe XD) to have an initial image for our website. Then working will start with front end team (Maryam & Omnia) to convert this design into actual code. After that we will connect with back end team (Ahmed & Alaa) to take our project into action with their functions and methods. Endly we will connect with machine learning with (Abdulrahman) which is responsible for algorithms that will predect the disease and achieve our goal.

2- Designing:

As mentioned before in planning phase we will design our website initially by the help of Adobe XD which is also known as Adobe Experience Design. It is a vector design tool for web and mobile applications, developed and published by Adobe Inc. It is available for macOS and Windows, and there are versions for iOS and Android to help preview the result of work directly on mobile devices.

3- Coding:

We divide our code into four main parts; Front End, Back End, Database & Machine learning.

1- With front end we will work with HTML, CSS, Java Script, Bootstrap & React library

2- With back end we will work with NodeJS

3- We will make Database with non SQL

4- For machine Learning we will work with python language & Django

1.5 General Constraints

Project constraints are the factors that limit your development process. Project constraints can be material, for example money that make up your project budget, or hardware that's needed to write the code. They can also be non-material: anything from time constraints to customer satisfaction.

Constraints that face us during project...

1. Time Constraint:

The time constraints in project management are the deadline by which a project should be completed, and also the development schedule that determines at which point each part of the project should be delivered. What concrete steps can your team take to properly schedule your project and manage the delivery time?

***Plan first...**

Invest your time into planning before the project begins, define the main goals for your team and find all the resources needed to complete each task.

***Schedule each activity...**

Each task should have its duration and delivery time.

***Monitor constantly...**

always make comparisons between the results from the past and the present to analyze what was done correctly and what could be improved. Then, we discuss this with the team and make changes to the next plans.

***Control the process...**

Always maintain control over the development process and identify the elements that have a positive or bad impact on the project.

***Use proper time management methodologies...**

Methodologies like Gantt charts, Critical path analysis, and PERT can help you with planning.

2. Scope constraint:

The scope of the project is usually defined during a discovery phase, where a business analyst extracts all the business goals and requirements, and learns about the business processes to determine the best functionality for solving particular issues of both business and its customers.

The steps we can take to manage scope constraints in our project:

***Keep the documentation clear and full…**

To prevent misunderstandings and confusion, you must carefully record every feature and every modification and maintain the whole project scope in one location at all times.

***Set up the change management…**

Making arbitrary modifications to the project can lead to delays.We need to have a certain procedure for making changes in the project, so that every stakeholder knows how exactly the change can be proposed, implemented, reviewed and accepted.

***Communicate frequently**

Always have a scope discussion with team members and stakeholders to ensure that everyone is on the same page.

*** Prioritize the tasks**

Prioritize tasks to optimize the development process. To decide on your next steps based on the hard facts you collect from your initial users after the launch, you may, for instance, determine which features to develop first for your MVP.

3.Cost constraint:

The cost of a project includes all the financial expenses like hardware, materials.

How we managed the cost constraints in our project…

***Use historical data…**

you may estimate the cost of your project, By evaluating it against previous examples of projects that are similar to it

***Estimate the resources…**

Determine the resources you'll need and their cost for each task.

***Compare vendor bids…**

Each of your potential vendors will offer their bids. Average them out to find out the typical cost of projects similar to yours.

The project management constraints triangle's three sides are interdependent on one another…

Here are the project constraints examples:

1. Time and Scope

You must also reduce the project's scope if you want to move the deadline closer and complete everything on schedule.

2.Cost and Scope

Reducing the scope will result in lower costs.

3. Time and Cost

These two factors are also often related. For example, if you want to reduce the time of your project without sacrificing its scope, you may need to expand your development team and this means more expenses.

Other constraints in software project management…

4.Quality

Quality is one of the major software project constraints that is present in any project. It depends heavily on all the parts of the constraint triangle, as it usually requires hefty investments of money and time, and is also influenced by the scope.

for example…

If you want your product to be developed faster, but also don't want to expand your budget, this will affect quality as well.

5.Benefits

The goals and business expectations of the project are its benefits.

6.Risks

Every software development project carries risks, thus it's important to take them into account during the development process.

The most common risk constraints in project management…

·A vendor doesn't deliver what was expected
·An important specialist leaves the project because of illness.
·A competitor launches a similar product

7.Resources

Project management resource constraints are connected with all other constraints, as every step of this process requires certain resources.

Include:

-Equipment -People - Facilities
-Materials -Software

Chapter 2: Planning and analysis

2.1: Project Planning…

2.1.1: Feasibility Study:

Introduction:

Heart disease is one of the leading causes of death worldwide, and early detection can significantly improve a patient's prognosis. Therefore, developing a machine learning model to predict heart disease could have significant benefits for patient care.

Objective:

The objective of this feasibility study is to determine if it is practical to develop a heart disease prediction model using machine learning techniques.

Methods:

To evaluate the feasibility of this project, we will undertake the following steps…

1.Data Collection: The first step is to collect a dataset of patient records containing information such as age, sex, blood pressure, cholesterol level, and other relevant factors. There are several publicly available datasets that we can use, such as the Cleveland Clinic

Foundation (CCF) dataset, the Hungarian Institute of Cardiology dataset, and the University of California, Irvine (UCI) dataset.

2.Data Preparation: Once we have the dataset, we will need to preprocess and clean the data. This step involves dealing with missing data, outliers, and other issues that may affect the accuracy of our model.

3.Feature Selection: We will select the most relevant features for predicting heart disease using various techniques such as correlation analysis, feature importance, and domain expertise.

4.Model Development: Once we have selected the relevant features, we will develop a machine learning model to predict heart disease. We can use various algorithms such as logistic regression, decision trees, random forests, or neural networks.

5.Model Evaluation: We will evaluate the performance of our model using metrics such as accuracy, precision, recall, and F1-score. We will also use techniques such as cross-validation to ensure that our model is robust and can generalize to new data.

6.Deployment: Once we have developed and evaluated our model, we will deploy it in a real-world setting to test its performance in a clinical environment.

Results:

Based on our preliminary analysis, we conclude that it is feasible to develop a heart disease prediction model using machine learning techniques. The availability of publicly available datasets and the variety of machine learning algorithms make it possible to develop accurate and robust models. However, there are still several challenges that we need to overcome, such as data quality, feature selection, and model interpretability. Therefore, we recommend further investigation into these areas before undertaking a full-scale project.

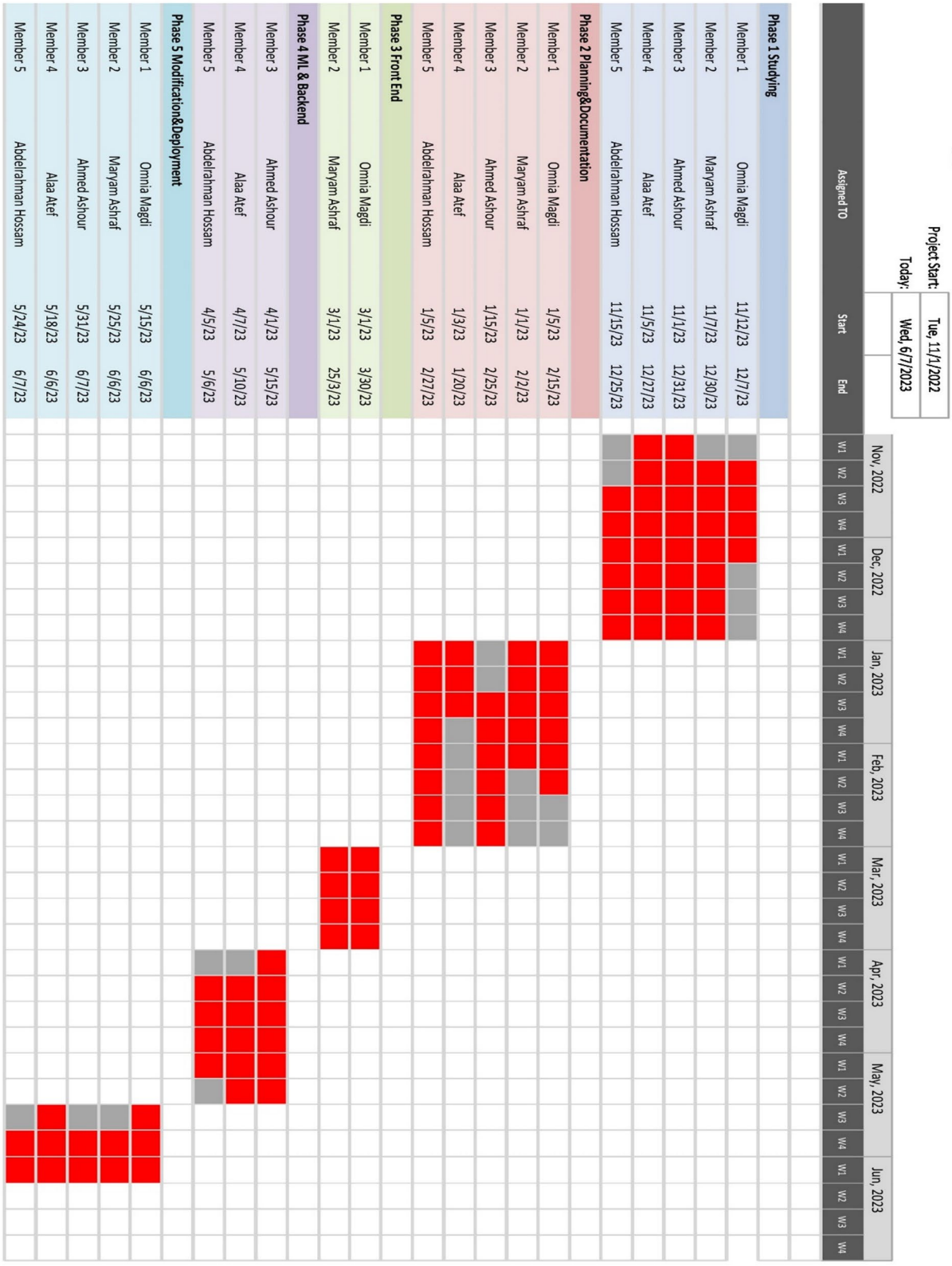
Conclusion:

In conclusion, a feasibility study of a heart disease prediction project suggests that it is practical to develop a machine learning model to predict heart disease. However, several challenges need to be addressed, and further investigation is required to ensure the accuracy, reliability, and interpretability of the model.

2.1.2: Estimated Cost

It depends on the amount of data we'll be using, the hosted server where we'll be running our web service, and how quickly we need the data to be retrieved. which is determined by the healthcare organization with which we are dealing.

2.1.3: Gantt chart



2.2:Analysis and limitations of existing system

We have searched extensively, but we have not found any website that adopts this idea.
However…

We have developed our idea based on a newfangled approach for early detection and prevention of ischemic heart disease using data mining. they proposed a prediction approach for knowing the risk of Ischemic Heart Disease (IHD) by using the earlier patient's dataset of Stroke. They have only conducted research based on numbers and have not implemented the idea in real-life application to benefit any individual.

We have implemented the idea in real-life application, making several changes such as using different datasets, technologies, and utilizing a much larger dataset to improve performance and accuracy. Additionally, we have developed a web application to assist users in determining whether or not they have Ischemic Heart Disease (IHD).

2.3: Need for the new system

The techniques that were used have some drawbacks like :

Decision Tree: One disadvantage of decision trees is that it can create countless trees for the same dataset, which can lead to high complexity. To address this, we can focus on improving the accuracy of predicting various heart-related diseases.

Support Vector Machine: Drawbacks of SVM computation may not be suitable for large datasets, and it may not perform well in cases where the dataset contains a lot of noise, meaning that the target classes are overlapping. To overcome these limitations, ensemble techniques can be considered to achieve higher accuracy in the future.

Naive Bayes: NB has some limitations such as lack of transparency, lengthy processing time, and difficulty in defining classification rules.

To address these limitations, we can develop a novel machine learning technique that can achieve higher accuracy in diagnosing a range of diseases.

Additionally, we will search for a more appropriate technique that has fewer disadvantages. And we will incorporate extra attributes that we believe will improve the accuracy and performance of the results.

2.4 Analysis of the new system…

2.4.1 User requirements:

-Users will enter their data into website and wait for results so requirements needed for them are…

1-Results appear without noticeable system delays.

2-Availability for many users to enter their data.

3-Ease of using and dealing with the system.

4-Correct results.

5-Results being in an understandable and simple form.

2.4.2 System Requirements:

-here we will describe the behavior and features of our system…

1.1 user will log in to the website

1.2 user will be asked to enter some data as name, age, gender and others

1.3 system will accept the data and work on it with some algorithms

1.4 system will generate a report for the user with the risk percentage for this user to have the disease

2.4.3 Domain Requirements:

Domain requirements are expectations related to a particular type of software, purpose, or industry vertical. Domain requirements can be functional or nonfunctional.
Domain requirements typically arise in the military, medical and financial industry sectors, among others.

For example…

A train control system must consider the braking characteristics in various weather scenarios.

How are they defined?

Can be either new functional requirements or constraints on existing requirements.

Why use domain requirements?

· To make the system workable
· If not satisfied, the system may not work

Domain Requirements Problems

1. Understandability

· Requirements are expressed in the language of the application domain.
· Software engineers may not understand domain language.

2. Implicitness

· Domain specialists understand the domain, so they don't make requirements explicit.
· Software engineers may not understand implicit requirements.

2.4.4 Functional Requirements:

What are Functional Requirements?

A functional requirement is a statement of how a system must behave. It defines what the system should do to meet the user's needs or expectations.

Functional requirements are made up of two parts

1. Function

The function is what the system does (e.g., "calculate sales tax").

2. Behavior

The behavior is how the system does it (e.g., "The system shall calculate the sales tax by multiplying the purchase price by the tax rate.").

Types of Functional Requirements…

Here are the most common functional requirement types:

1. Business Regulations:

What do you want your system to do? What are the features you need so you can achieve your goals?
You need to define every system activity for each function within the system and address all functional requirement types. That's why this section will probably be the longest amongst the others as many requirements may fall under this categorization.

2. Certification Requirements

To work on the system, your company may require certifications, such as security certificates.

3. Authorization Levels

These functions determine various system access levels and decide who can CRUD (change, read, update, or delete) information.

1. Authentication functions

They concern the information users share with the system and their authentication level.

2. Audit tracking

The technique of tracking crucial data is known as audit tracking.

3. Searching/reporting requirements

How users can search for and get data is described in this section of the requirements.

4. Historical data

If your database is dynamic, you will see an increase in data, thus you must specify the amount of storage you need to handle this increase.

5. External interfaces

These functions concern the external interface of systems other than the main system.

6. Archiving

The projects must have the ability to archive the data for long-term storage because your system's data may increase faster than your storage capacity can handle.

*Creating Functional Requirements:

When creating functional requirements, it is important to keep in mind that they should be specific, measurable, and achievable. In other words, your functional requirements should:

- **Give the system precise instructions.**
- **Be quantifiable so you can determine whether the system is executing it**
- **Be realistic and stick to the deadline you've set.**
- **Be relevant to your business's objectives**

Examples:

1. A user shall be able to log into the system using their username and password.

In this example, the function is "login", and the behavior is "The system shall allow a user to login using their username and password."

2. The system shall calculate the sales tax for the user's purchase.

In this example, the function is "calculate sales tax" and the behavior is "The system shall calculate the sales tax by multiplying the purchase price by the tax rate."

3. The system shall send a confirmation email to the user after they have successfully placed an order.

In this example, the function is "send confirmation email" and the behavior is "The system shall send a confirmation email to the user after they have successfully placed an order."

Why do Functional Requirements need to be documented?

-The only reliable source for the people involved. Clear documentation of requirements helps to prevent misunderstandings by keeping all

developers, designers, and QA testers on the same page and focused on the same objectives.
-Meetings are held less frequently now. Regular meetings are unnecessary when the team has a written record and a shared understanding.
-The predictability of projects increases. The team can more correctly estimate development time and cost and produce a product that satisfies expectations when given detailed, high-quality requirements.

2.4.5 Non-Functional Requirements:

1-Performance: it is how fast does the system return results.

2-Scalability: it is how much will the performance change with higher workloads.

3-Portability and compatibility: Describes which hardware, operating systems, and browsers, along with their versions does the software run on? Does it conflict with other applications and processes within these environments?

4-Reliability: Describes how often does the system experience critical failures.

5-Maintainability: Describes how much time does it take to fix the issue when it arises.

6-Availability: Describes how is user availability time compared to downtime.

7- Security: Describes how well are the system and its data protected against attacks.

8- Localization: Describes if the system is compatible with local specifics or not.

9- Usability: Describes how easy it is for a customer to use the system.

10- Recoverability: The ability to recover from a crash or a failure in the system and return to full operations.

2.5: Advantages of the new system

1- this website enable user to know if he has IHD or not by only enter some attributes and one click

2- it makes it easier for the patient to know if he has IHD or not without the need for many expensive tests.

3- There is also a section for each doctor to see the results of his patients.

2.6: Risk and Risk Management

What is Risk Management?

Risk management encompasses the identification, analysis, and response to risk factors that form part of the life of a business. Effective risk management means attempting to control, as much as possible, future outcomes by acting proactively rather than reactively. Therefore, effective risk management offers the potential to reduce both the possibility of a risk occurring and its potential impact.

Risk Management Structures:

Risk management structures are tailored to do more than just point out existing risks. A good risk management structure should also calculate the uncertainties and predict their influence on a business. Consequently, the result is a choice between accepting risks or rejecting them. Acceptance or rejection of risks is dependent on the tolerance levels that a business has already defined for itself.

If a business sets up risk management as a disciplined and continuous process for the purpose of identifying and resolving risks, then the risk management structures can be used to support other risk mitigation systems. They include planning, organization, cost control, and budgeting. In such a case, the business will not usually experience many surprises, because the focus is on proactive risk management.

Response to Risks:

Response to risks usually takes one of the following forms:

1.Avoidance: A business strives to eliminate a particular risk by getting rid of its cause.

2.Mitigation: Decreasing the projected financial value associated with a risk by lowering the possibility of the occurrence of the risk.

3.Acceptance: In some cases, a business may be forced to accept a risk. This option is possible if a business entity develops contingencies to mitigate the impact of the risk, should it occur.

When creating contingencies, a business needs to engage in a problem-solving approach. The result is a well-detailed plan that can be executed as soon as the need arises. Such a plan will enable a business organization to handle barriers or blockage to its success because it can deal with risks as soon as they arise.

Importance of Risk Management:

Risk management is an important process because it empowers a business with the necessary tools so that it can adequately identify and deal with potential risks. Once a risk has been identified, it is then easy

to mitigate it. In addition, risk management provides a business with a basis upon which it can undertake sound decision-making.

For a business, assessment and management of risks are the best way to prepare for eventualities that may come in the way of progress and growth. When a business evaluates its plan for handling potential threats and then develops structures to address them, it improves its odds of becoming a successful entity.

In addition, progressive risk management ensures risks of high priority are dealt with as aggressively as possible. Moreover, the management will have the necessary information that they can use to make informed decisions and ensure that the business remains profitable.

Risk Analysis Process:

Risk analysis is a qualitative problem-solving approach that uses various tools of assessment to work out and rank risks for the purpose of assessing and resolving them. Here is the risk analysis process…

1. Identify existing risks

Risk identification mainly involves brainstorming. A business gathers its employees together so that they can review all the various sources of risk. The next step is to arrange all the identified risks in order of priority. Because it is not possible to mitigate all existing risks, prioritization ensures that those risks that can affect a business significantly are dealt with more urgently.

2. Assess the risks

In many cases, problem resolution involves identifying the problem and then finding an appropriate solution. However, prior to figuring out how best to handle risks, a business should locate the cause of the risks by asking the question, “What caused such a risk and how could it influence the business?”

3. Develop an appropriate response

Once a business entity is set on assessing likely remedies to mitigate identified risks and prevent their recurrence, it needs to ask the following questions: What measures can be taken to prevent the identified risk from recurring? In addition, what is the best thing to do if it does recur?

4. Develop preventive mechanisms for identified risks

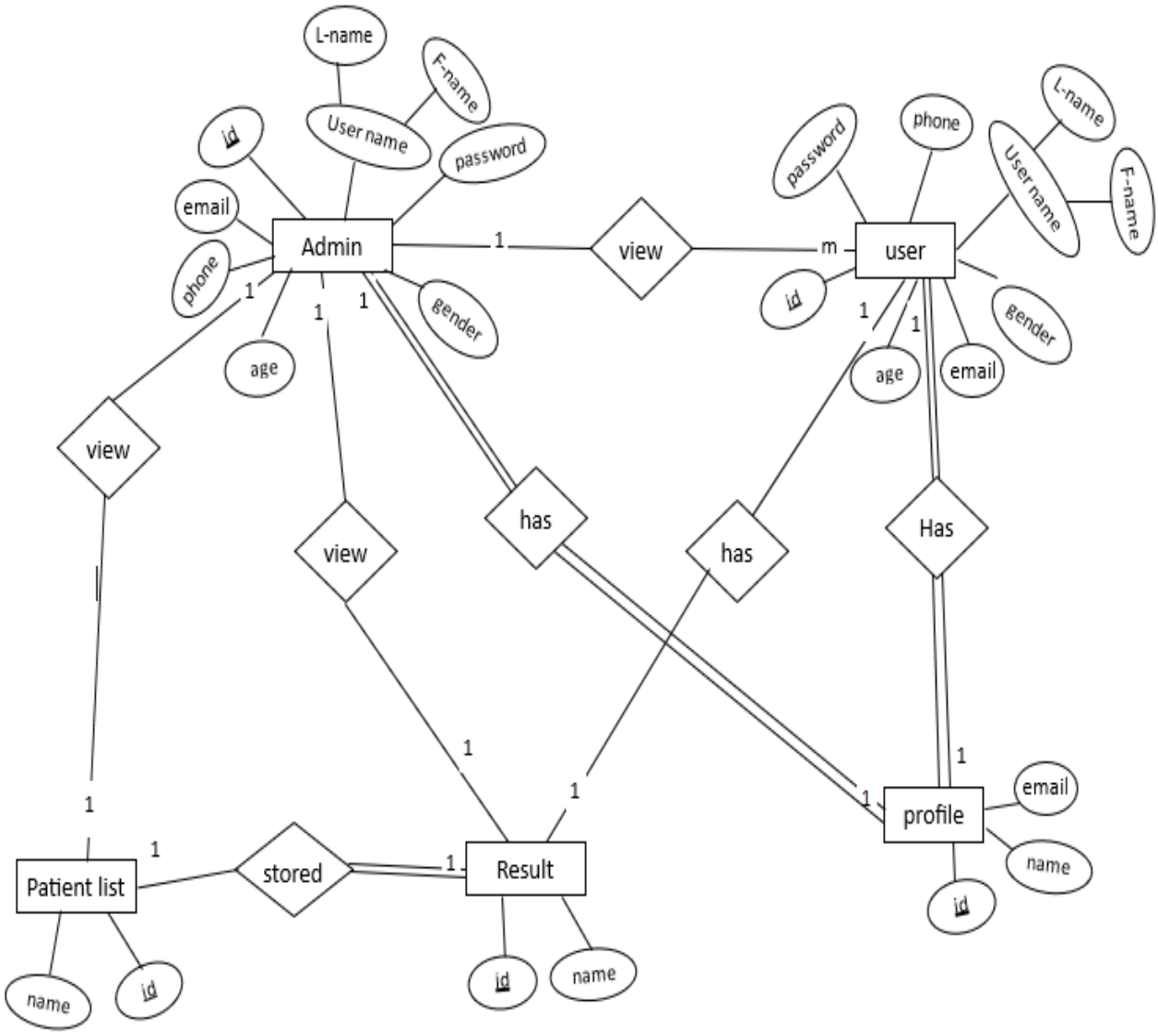
Here, the ideas that were found to be useful in mitigating risks are developed into a number of tasks and then into contingency plans that can be deployed in the future. If risks occur, the plans can be put to action.

Summary

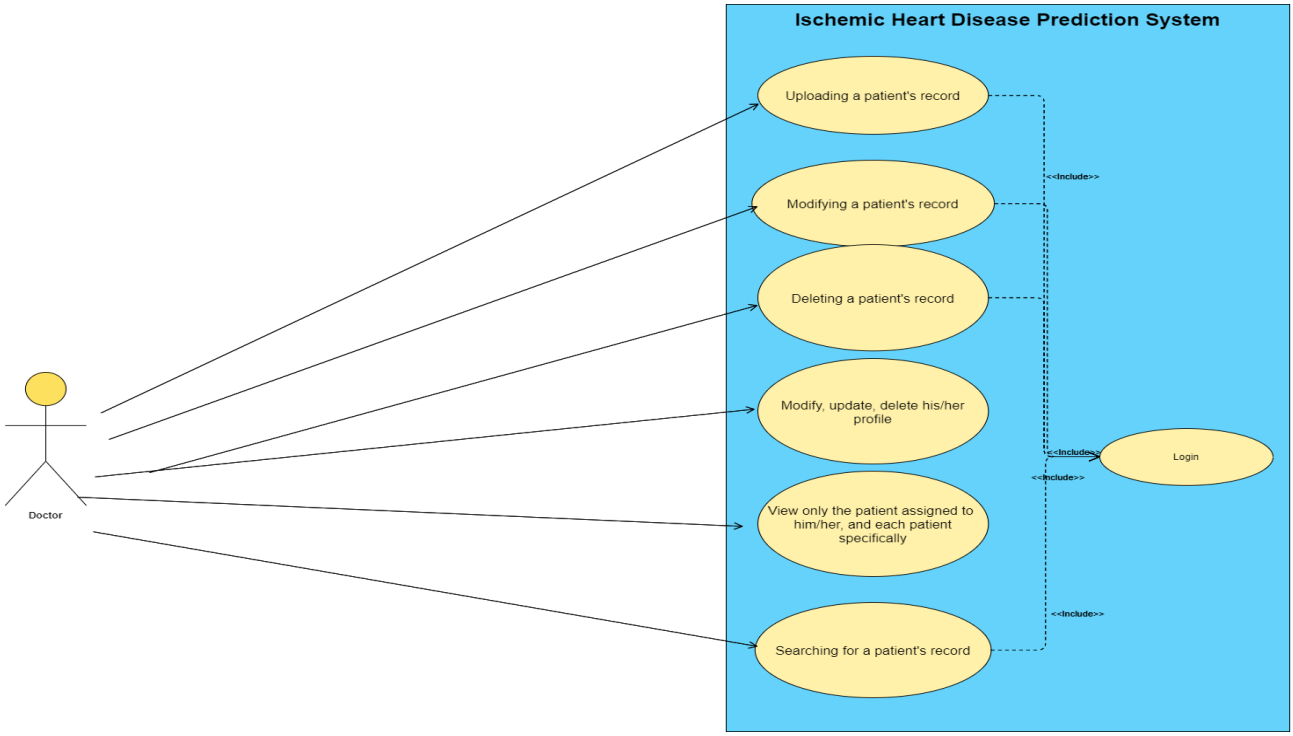
Our business ventures encounter many risks that can affect their survival and growth. As a result, it is important to understand the basic principles of risk management and how they can be used to help mitigate the effects of risks on business entities.

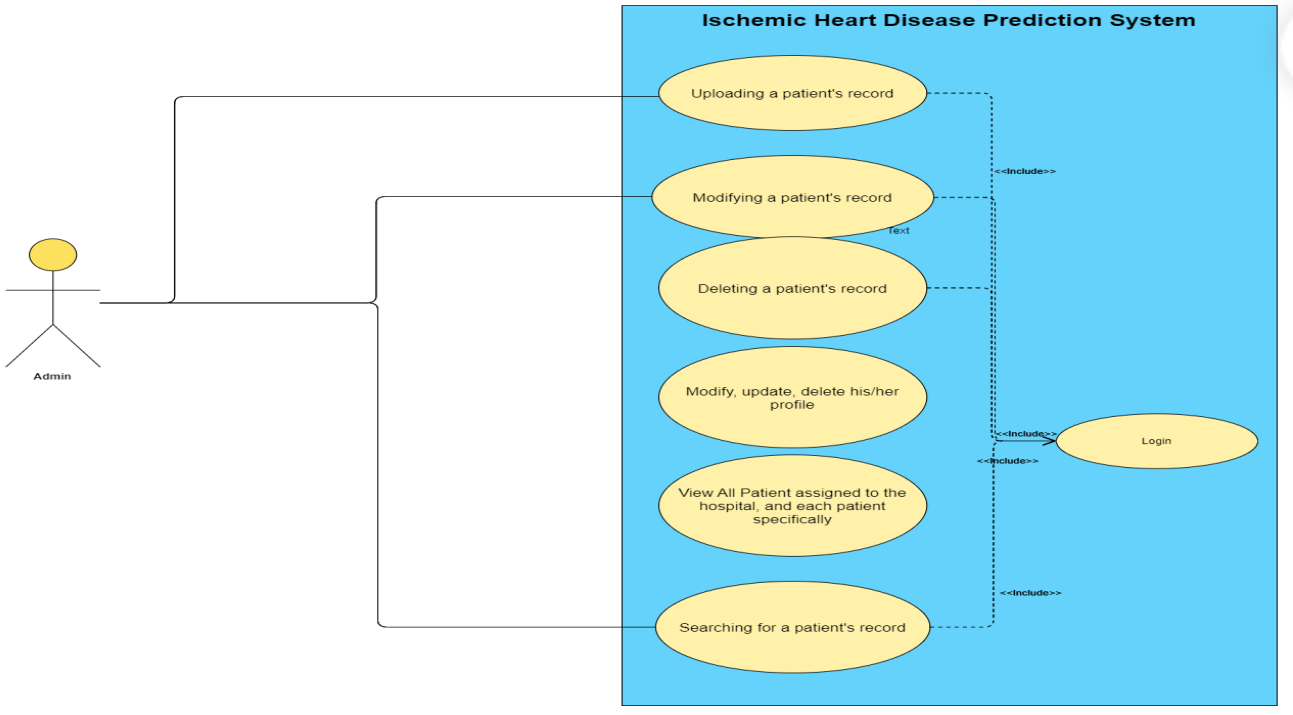
Chapter 3: Software Design

3.1 ERD



3.2 Use-case Diagram





Use Case Description:

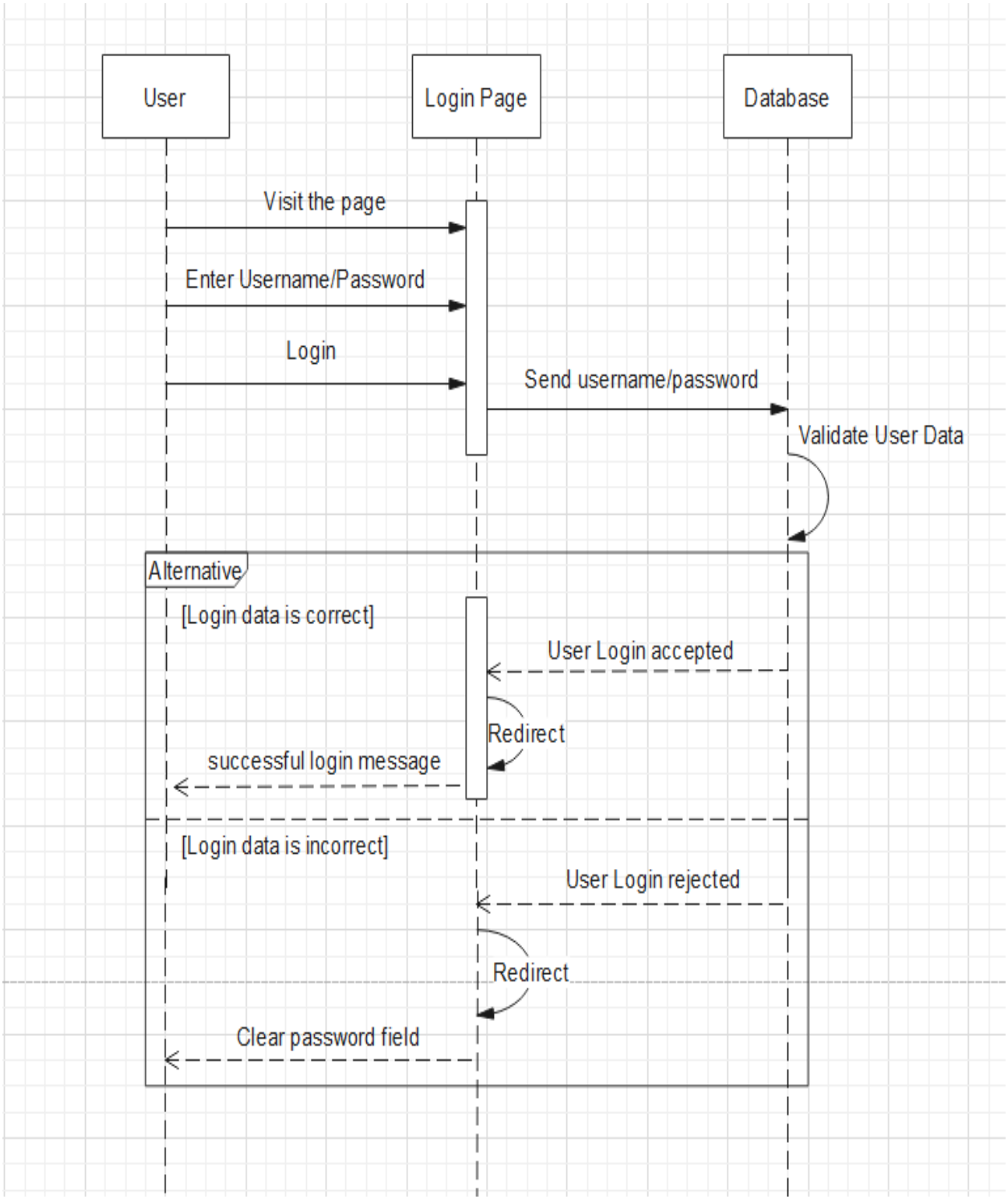
Main Successful Scenario:
The patient data is clear, non-ambiguous, machine-readable, and ready for an algorithm to be applied to it.

Unsuccessful Scenario:
The patient data may be unclear, ambiguous, or not preprocessed enough to be entered into the machine, or it may be insufficient -> not having enough attributes, so the algorithm cannot work on it.
What pre and post-condition(s) you can obtain from the below description of a patient entry data process?
When a clinician, receptionist, or medical secretary enters patient data (EHR) to be evaluated on the system, the data is pre-processed and the prediction assessment percentage is revealed -> so we know the case's proclivity to the system (IHD).

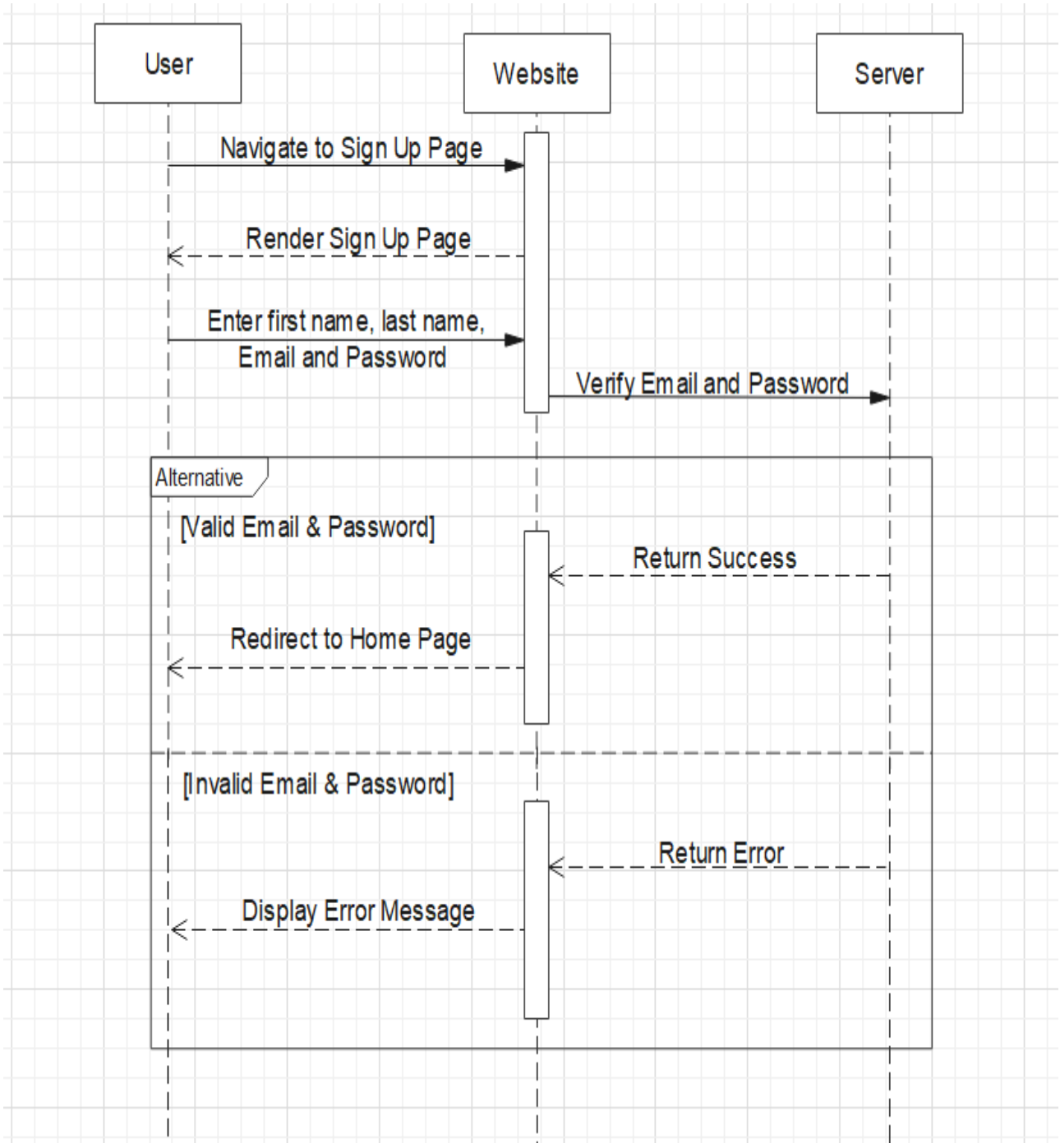
Pre-condition(s): There must be a patient visiting this healthcare facility, and the clinician or medical assistant must be aware of our system and enter his data into our program in order to obtain an appropriate prediction.
Post-condition(s): If necessary, the patient will be directed to the appropriate department and treated.

3.4 Sequence diagram:

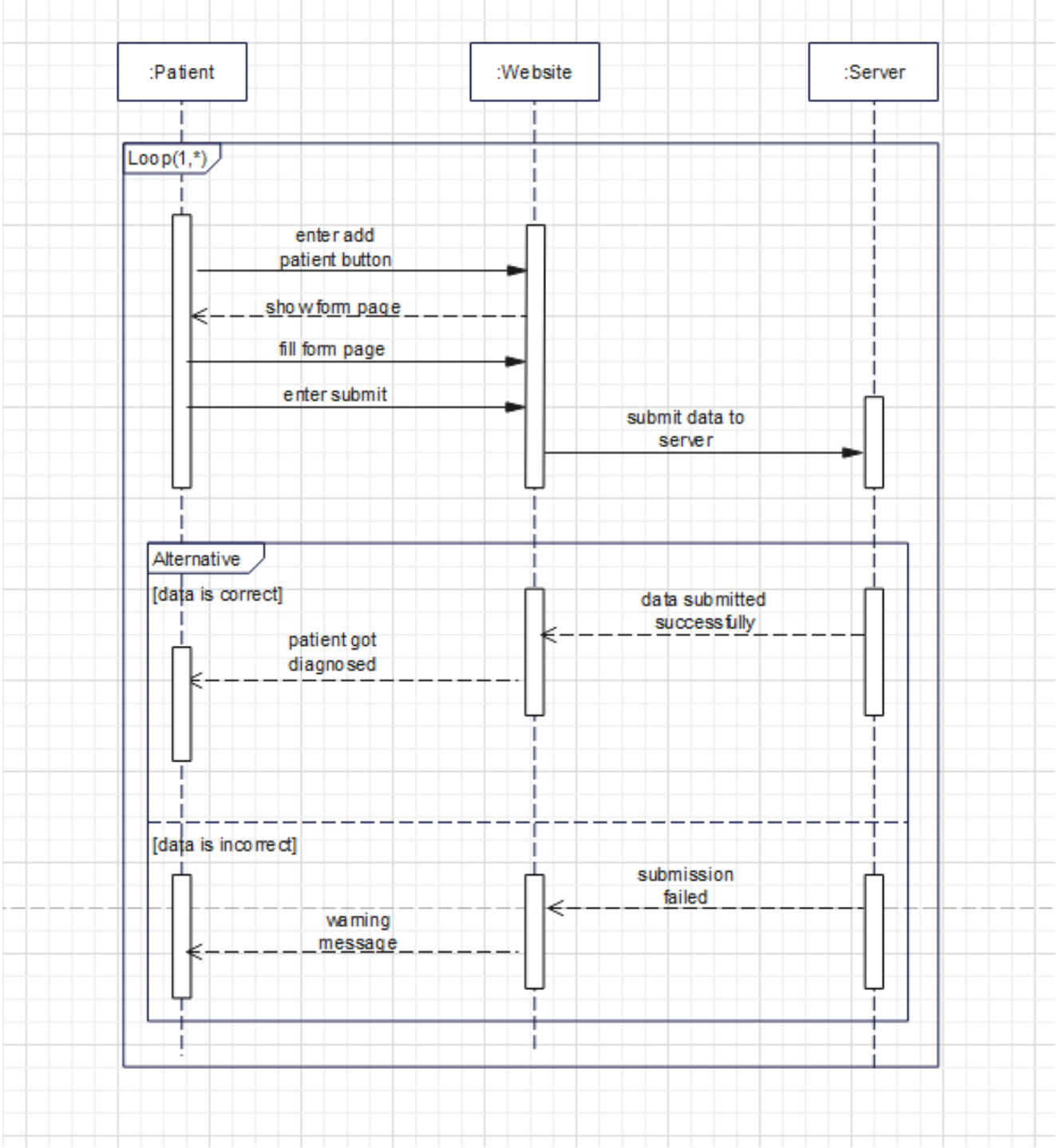
3.4.1: Login Sequence Diagram:



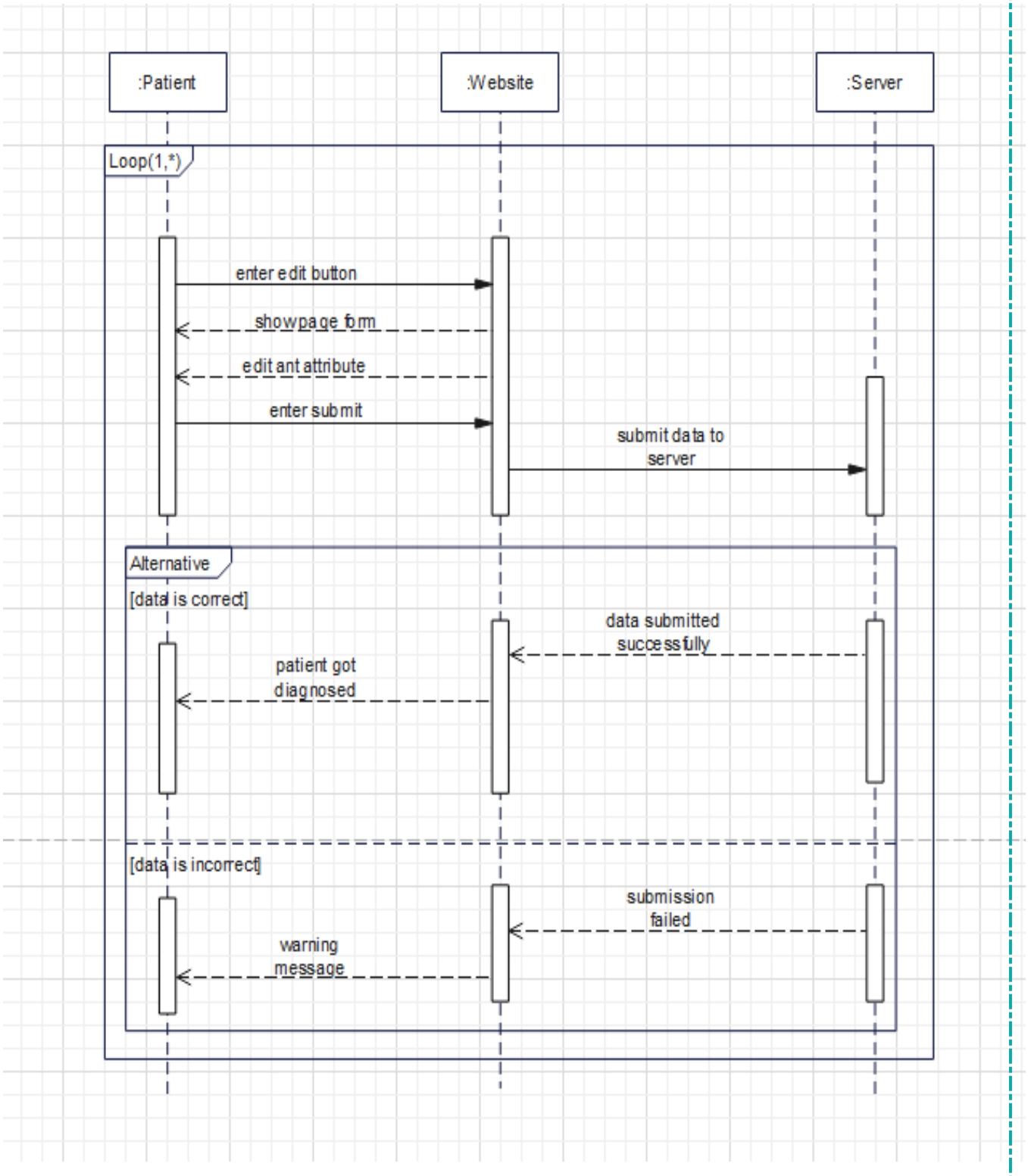
3.4.2: Sign Up Sequence Diagram:



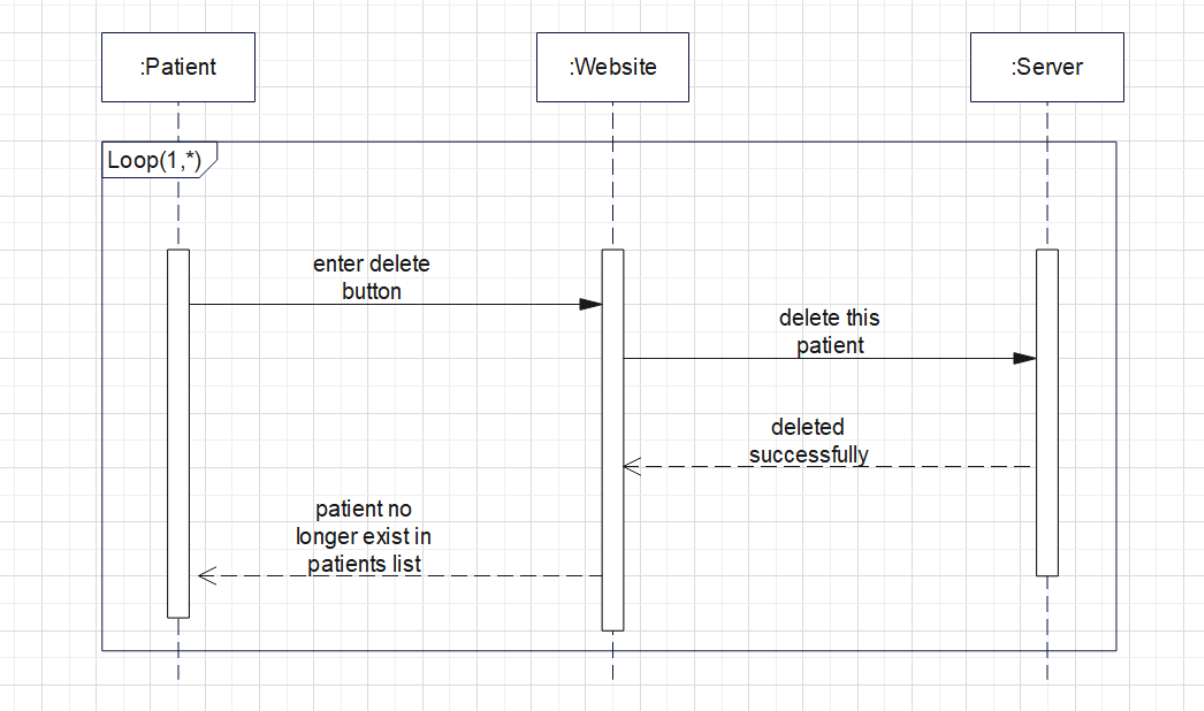
3.4.3: Add patient Sequence Diagram:



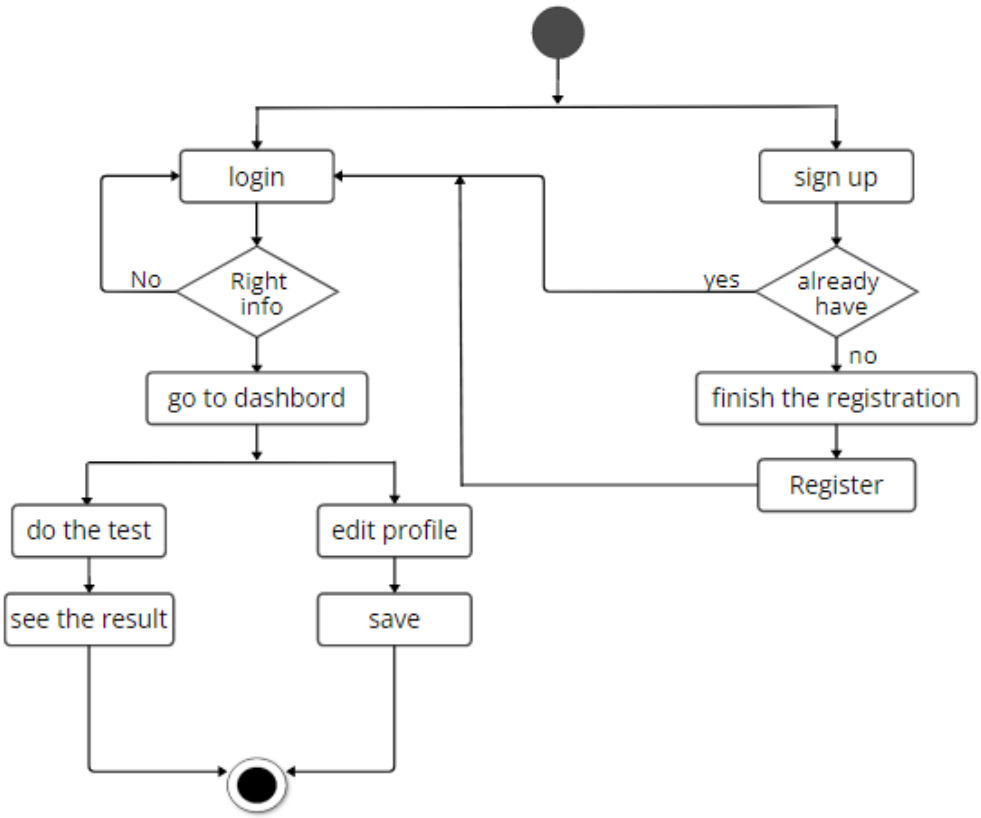
3.4.4: Edit patient sequence diagram:

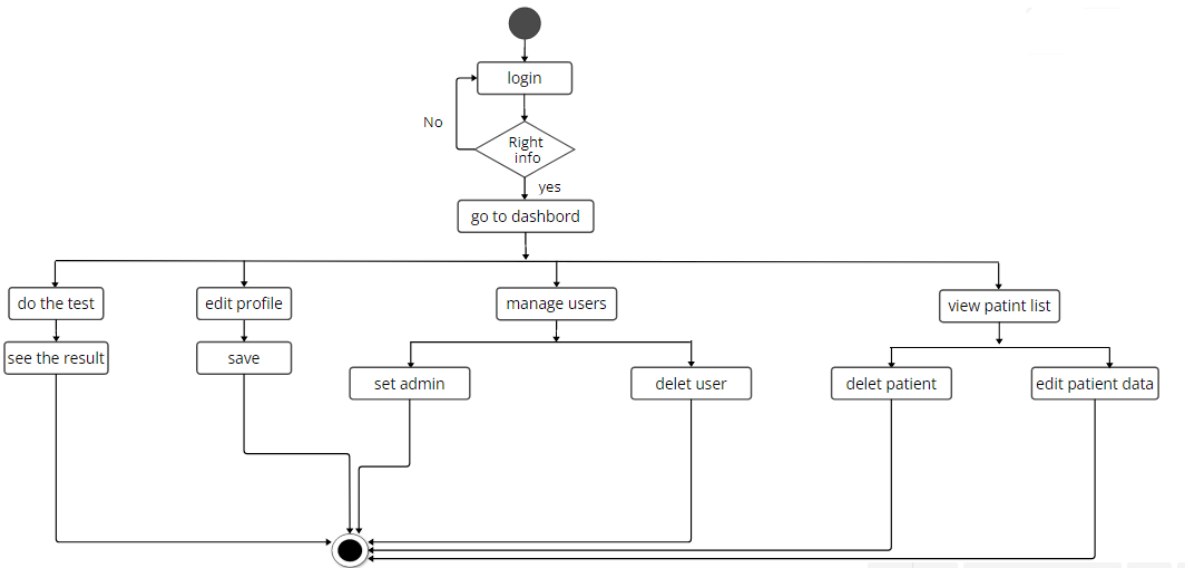


3.4.5 Delete patient sequence diagram:



3.5 Activity diagram:





Chapter 4 : Implementation

4.1: Software Architecture:

The software architecture for this project consists of both a client-side and server-side component. The client-side is built using ReactJS, which is a JavaScript framework that allows for the creation of dynamic web applications. The server-side is built using Node.js, which is a runtime environment that allows for server-side JavaScript execution. The communication between the client and server is facilitated using RESTful APIs, which allow for the transfer of data between the two components.

Object-relational mappings (ORMs) are used to facilitate data transfer between the client and server. The ORM used in this project is Sequelize, which is a popular Node.js ORM that provides an easy-to-use interface for working with databases. The ORM maps data between objects in the client-side code and the corresponding tables in the server-side database, allowing for seamless data transfer between the two components.

Overall, the software architecture for this project is designed to provide a robust and efficient system for predicting heart disease risk.

1-Login

```
export const logIn = asyncHandler(async (req, res, next) => {
  const { email, password } = req.body;
  const user = await userModel.findOne({ email });
  // Check User
  if (!user) {
    return next(new Error("Account Not Found !", { cause: 404 }));
  }
  // Check If Confirmed Email
  else if (!user.confirmEmail) {
    return next(new Error("Please Confirm Email First", { cause: 400 }));
  }
  // Check Password
  else {
    const match = compare({ plainText: password, hashedValue: user.password });
    if (!match) {
      return next(new Error("In-Valid Password !", { cause: 400 }));
    }
    // Valid Email & Password
    else {
      const accessToken = createToken({
        payload: {
          id: user._id,
          email: user.email,
          status: user.status,
          role: user.role,
          isLoggedIn: true,
        },
        expiresIn: 60 * 60 * 24,
      });
      const refreshToken = createToken({
        payload: {
          id: user._id,
          email: user.email,
          status: user.status,
          role: user.role,
          isLoggedIn: true,
        },
        expiresIn: 60 * 60 * 24 * 365,
      });
      user.status = "OnLine";
      await user.save();
      return res
        .status(200)
        .json({ message: "Success", accessToken, refreshToken });
    }
  }
});
```

2-Registration

```
12  export const signup = asyncHandler(async (req, res, next) => {      You, 4 days ago • First …
13    const { firstName, lastName, email, password } = req.body;
14    // Check if User Exist In The System
15    const checkUser = await userModel.findOne({ email });
16  > if (checkUser) {···
18    }
19    // Sending Email To Confirm
20    // First Token Valid For 5 Mins
21  > const token = createToken({···
25    });
26    const link = `${req.protocol}://${req.headers.host}/auth/confirmEmail/${token}`;
27    // Second Token Valid For 30 Days To Request New Confirmation
28  > const refreshToken = createToken({···
32    });
33    const refreshLink = `${req.protocol}://${req.headers.host}/auth/confirmNewEmail/${refreshToken}`;
34  > const html = `<!DOCTYPE html>···
230   </html>`;
231   const info = await sendEmail({ to: email, subject: "Confirm Email", html });
232 > if (!info) {···
234   }
235   // Hashing Password
236   const hashPassword = hash({ plainText: password });
237   // Creating Model
238 > const user = await userModel.create({···
244   });
245   return res.status(201).json({ message: "Success", user: user._id });
246 });
```

3-Forget Password

```
538 export const forgetPassword = asyncHandler(async (req, res, next) => {
539   const { email } = req.body;
540   const user = await userModel.findOne({ email });
541   if (!user || !user.confirmEmail) {
542     return next(new Error("Email Not Found !", { cause: 404 }));
543   } else {
544     const nanoid = customAlphabet(process.env.alphabet, 6);
545     const resetCode = nanoid();
546     // console.log(resetCode);
547     user.resetCode = resetCode;
548     await user.save();
549   > const html = `<!DOCTYPE html>···
578     </html>`;
579     const info = await sendEmail({
580       to: email,
581       subject: "Reset Password",
582       html,
583     });
584     // console.log({ info });
585     if (!info) {
586       return next(new Error("Rejected Email"), { cause: 400 });
587     }
588     return res.status(201).json({ message: "Success", user: user.email });
589   }
590 });
```

4-Get Users

```
647  export const getUsers = asyncHandler(async (req, res, next) => {
648    const admin = await userModel.findById(req.user._id);
649    if (admin.role != "Admin") {
650      return next(new Error("Not Allowed To Show Users"), { cause: 400 });
651    }
652    const users = await userModel
653      .find({ _id: { $ne: req.user._id } })
654      .select("firstName lastName email status phone gender role confirmEmail");
655    // Check User
656    if (!users) {
657      return next(new Error("Users Not Found !", { cause: 404 }));
658    }
659    return res.status(200).json({ message: "Success", users });
660  });
```

5-Delete User

```
661  // 09- Delete User
662  export const deleteUser = asyncHandler(async (req, res, next) => {
663    const { id } = req.body;
664    const admin = await userModel.findById(req.user._id);
665    if (admin.role != "Admin") {
666      return next(new Error("Not Allowed !"), { cause: 400 });
667    }
668    const user = await userModel.findByIdAndDelete(id);
669    // Check User
670    if (!user) {
671      return next(new Error("Account Not Found !", { cause: 404 }));
672    }
673    return res.status(200).json({ message: "Success", user: user._id });
674  });
675
```

6-Set Admin

```
627  // 08- Set Admin
628  export const setAdmin = asyncHandler(async (req, res, next) => {
629    const { email } = req.body;
630    const admin = await userModel.findById(req.user._id);
631    if (admin.role != "Admin") {
632      return next(new Error("Not Allowed To Change Roles"), { cause: 400 });
633    }
634    const user = await userModel.findOne({ email });
635    // Check User
636    if (!user) {
637      return next(new Error("Account Not Found !", { cause: 404 }));
638    }
639    if (user.role === "Admin") {
640      return next(new Error("User Allready Admin !", { cause: 400 }));
641    }
642    user.role = "Admin";
643    await user.save();
644    return res.status(200).json({ message: "Success", user: user._id });
645  });
```

7-Profile

```
6   // 01- Get User Profile
7   export const getProfile = asyncHandler(async (req, res, next) => {
8     const user = await userModel
9       .findById(req.user._id)
10      .select("firstName lastName email status phone gender profilePic role");
11    await res.set("Cache-Control", "no-cache");
12    return res.status(200).json({ message: "Success", user });
13  });       You, 4 days ago • First
14  // 02- Profile Picture
15  export const profilePic = asyncHandler(async (req, res, next) => {
16    const user = await userModel.findById(req.user._id);
17    // console.log(req.file);
18    const { secure_url, public_id } = await cloudinary.uploader.upload(
19      req.file.path,
20      {
21        folder: `${process.env.APP_NAME}/User/${req.user._id}/Profile`,
22      }
23    );
24    // If You Need To Delete Last Image
25    // await cloudinary.uploader.destroy(user.profilePic.public_id);
26    user.profilePic = { secure_url, public_id };
27    await user.save();
28    return res.status(202).json({ message: "Success", user: user.profilePic });
29  });
```

8-Confirm Email

```
247  // 02- Confirm Email Method
248  export const confirmEmail = asyncHandler(async (req, res, next) => {
249    // Destructing Email From Token
250    const { token } = req.params;
251    const { email } = verifyToken({ token, signature: process.env.EMAIL_TOKEN });
252    if (!email) {
253      return next(new Error("In-Valid Token Payload", { cause: 400 }));
254    }
255    // Update User status
256    const user = await userModel.updateOne({ email }, { confirmEmail: true });
257    return user.modifiedCount
258      ? res.status(200).redirect(`${process.env.FE_URL}/log-in`)
259      : next(new Error("Not Register Account !", { cause: 400 }));
260  });
```

9-Logout

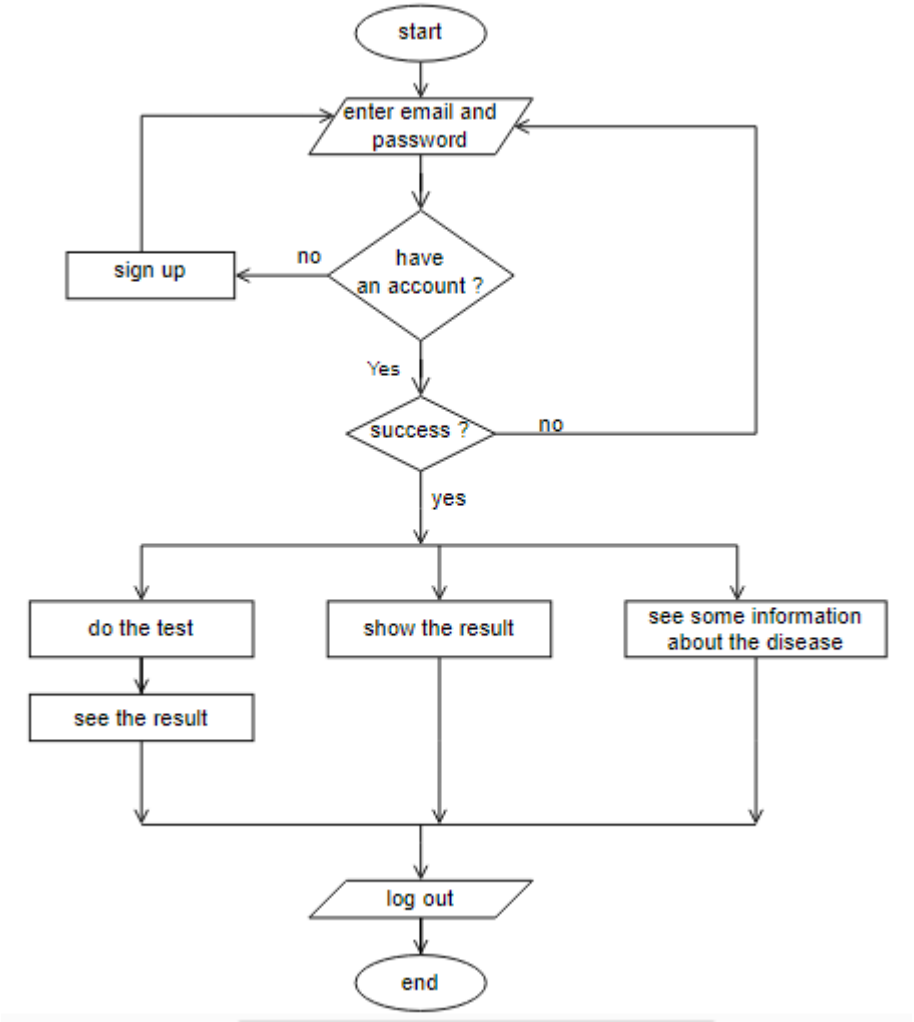
```
616  // 07- Logout Method
617  export const logout = asyncHandler(async (req, res, next) => {
618    const user = await userModel.findById(req.user._id);
619    // Check User
620    if (!user) {
621      return next(new Error("Account Not Found !", { cause: 404 }));
622    }
623    user.status = "OffLine";
624    await user.save();
625    return res.status(200).json({ message: "Success" });
626  });
```

10-Update Password

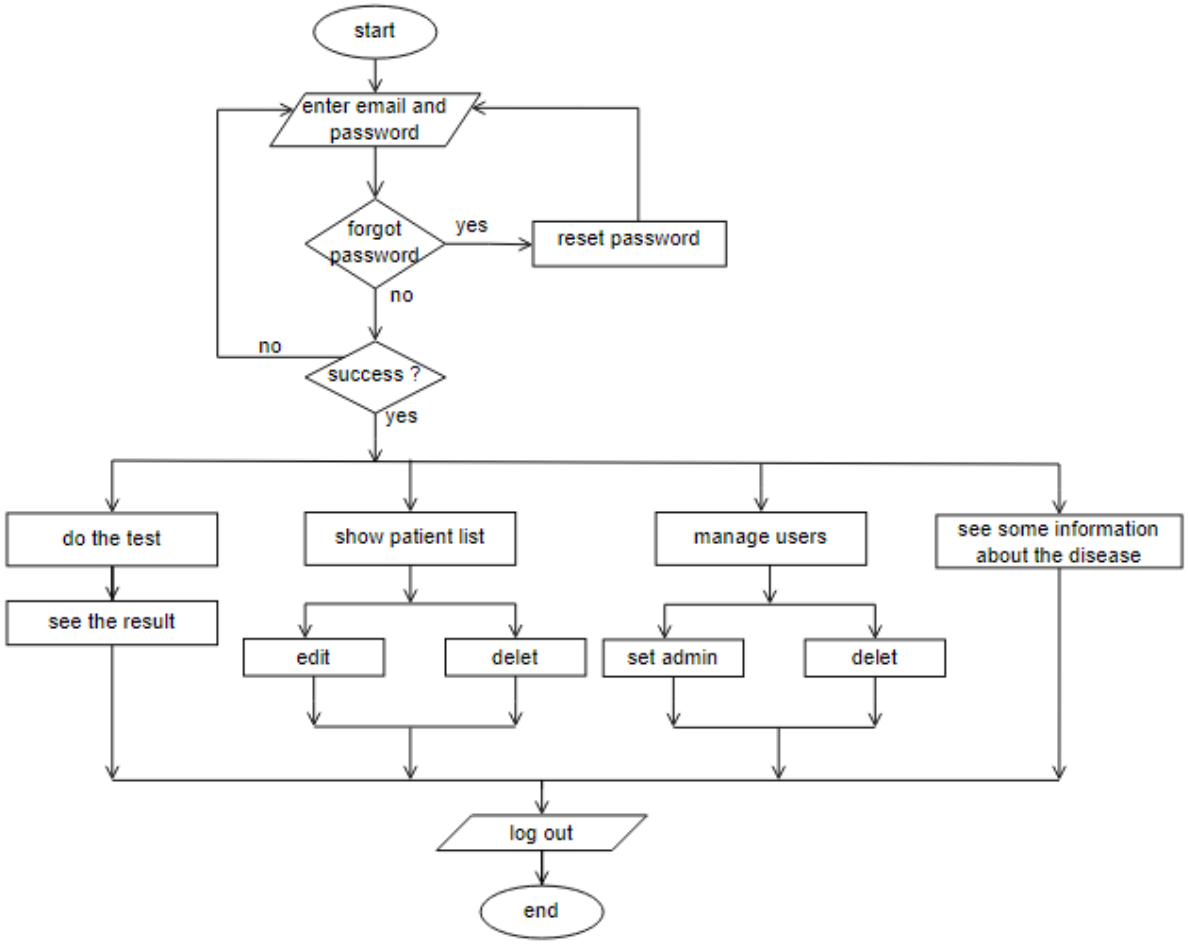
```
42  // 04- Update Password
43  export const updatePassword = asyncHandler(async (req, res, next) => {
44    const { oldPassword, newPassword } = req.body;
45    const user = await userModel.findById(req.user._id);
46    // Check Old Password
47    if (!compare({ plainText: oldPassword, hashedValue: user.password })) {
48      return next(new Error("In-Valid Old Password !", { cause: 400 }));
49    }
50    // Compare Old With New
51    else if (newPassword === oldPassword) {
52      return next(
53        new Error("Can't Be The Same As Old Password !", { cause: 400 })
54      );
55    }
56    // Hashing Password & Update
57    else {
58      const hashPassword = hash({ plainText: newPassword });
59      user.password = hashPassword;
60      await user.save();
61      return res.status(200).json({ message: "Success", user: user._id });
62    }
63  });
```

4.2 Flowchart:

User Flowchart:



Admin flowchart:



Pseudocode:

Here is the communication steps between the client and server…

A. Client side:

1. Open a connection to the server
2. Request to add patient
3. Wait for response from the server
4. Parse the response and display it to the user
5. Start filling the form
6. Submit the form
7. Result displayed in the table
8. Close the connection to the server

B. Server side:

1. Listen for incoming connections from clients
2. Accept a connection from client
3. Receive request from client
4. Process the request
5. Display the form
6. Accept form submission
7. Add new patient in database
8. Process on submitted data
9. Display the result to the client
10. Close the connection to the client

Chapter 5: Testing

5.1 Unit Testing:

What is unit testing?

Unit Testing is a type of software testing where individual units or components of the software are tested. Its goal is to ensure that every piece of software code functions as intended. Developers perform unit testing while creating an application (the coding phase). Unit tests isolate a specific piece of code and validate its correctness. An individual function, method, procedure, module, or object might be considered a unit.

Advantages and disadvantages of unit testing…

1. Advantages to unit testing include:

-The earlier a problem is identified, the fewer compound errors occur.
-The costs of resolving a problem sooner rather than later can often be significantly less.
-Processes for debugging are made much easier.
-The code base is easily modifiable by developers.
- Developers can also reuse code, migrating it to new projects.

2. Disadvantages include:

-Not all bugs will be found during tests.
-Unit testing does not detect integration issues; it just checks data sets and their functionality.
-More lines of test code may need to be written to test one line of code—creating a potential time investment.

5.2 Integrated Testing:

We performed integration testing after each module was developed and tested separately, and we tested the system to ensure that all components worked together seamlessly, Unit testing was also performed to ensure that every piece of software code functions working properly. However, we encountered a low accuracy rate during testing, and thus we made further efforts to improve the accuracy of the system

5.3AdditionalTesting:

Usability testing:

Introduction:

Welcome to our CardioVision web app, which predicts the risk of developing heart disease based on some data from the user. Thank you for taking the time to participate in this usability test, which will help us improve the app. Please let me know if you have any questions before we begin.

Task Instruction:

For this test, I'd like you to imagine that you have been experiencing chest pains and shortness of breath. You want to use the heart disease prediction app to determine your risk of developing heart disease. Please complete the following tasks:

a. Enter your personal information, such as age, gender, weight, etc.

b. Answer the medical history questions.

Observation and Questions:

As you use the app, I'll be observing your actions and taking note of any issues or difficulties you encounter. Please also share your thoughts and feelings as you complete the tasks. Here are some questions to help guide our discussion:

a. How easy or difficult was navigating the app?

b. Did you encounter any problems or confusion during the process?

c. Was there anything that stood out to you as particularly frustrating or confusing?

d. What did you like/dislike about the app?

e. Were there any features that you thought were missing or could be improved?

f. How confident do you feel in the accuracy of the heart disease prediction?

g. Would you feel comfortable sharing the results with a doctor or medical professional?

h. How likely are you to recommend this app to a friend or family member who may be at risk of heart disease?

Closing:

Thank you again for your time and feedback.

Do you have any additional comments or suggestions for the app? Is there anything else you'd like to share about your experience? We appreciate your input and will use it to make the app as user-friendly and effective as possible.

Chapter 6: Results and Discussion

6.1 Results:

6.1.1 Expected result:

The expected result of the project was to develop a web app that assigns patients to the doctor related to them and predicts whether or not the patient has heart disease. Specifically, the app was designed to:

-Allow doctors to view a list of their assigned patients and their relevant medical data.

-Provide an accurate prediction of each patient's risk of developing heart disease based on their medical history and other factors.

-Enable doctors to take appropriate action based on the predicted risk, such as scheduling follow-up appointments or ordering additional tests.

-Ensure the privacy and security of patient data in accordance with industry standards and regulations.

6.1.2 Actual results

After completing the project, we can report that we achieved our intended goals and the app works well. Specifically, we found that:

-The app successfully assigns patients to their doctors and presents relevant medical data in an organized and user-friendly way.

-The heart disease prediction algorithm we implemented shows good accuracy, with a high percentage of correctly identified cases of heart disease.

-The app provides doctors with clear and actionable information to guide their treatment decisions and improve patient outcomes.

-We implemented appropriate security measures to protect patient data, including encryption and access controls

-Overall, we are satisfied with the performance of the app and believe it will be a valuable tool for doctors in the diagnosis and treatment of heart disease.

6.2 Discussion:

We managed to get the same expected result.

-The website allocates patients to their doctors successfully.
-The application provides medical professionals clear and useful information to help them make treatment decisions and enhance patient outcomes.
-The heart disease prediction algorithm we implemented shows good accuracy.
-To safeguard patient data, we put in place the necessary security precautions, such as encryption and access limits.

Chapter 7: Conclusion

Our report sums up our project idea from different perspectives. We identified our purpose , plan , requirements , software design , implementation & testing. At the end we achieved what we are looking for from the beginning. We end up with a website with a capability of predicting whether the user is ischemic heart diseased or not according to data we collected from him with a satisfactory accuracy.

Chapter 8: Future work

we are going to add more features to our website Like:

- if a person does not have IHD, perhaps we Can try to estimate the likelihood of this Person's Condition.

- we also are going to enhance its Performance and accuracy.

- we will try to establish the necessary routine for the affected person to prevent their Condition from deteriorating.

Bibliography

https://www.kaggle.com/datasets/alexteboul/heart-disease-health-indicators-dataset?resource=download

https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

https://www.ncbi.nlm.nih.gov/books/NBK209964/

https://0c122ole5-1103-y-https-iopscience-iop-org.mplbci.ekb.eg/book/edit/978-0-7503-1997-3

https://www.investopedia.com/terms/d/datamining.asp#:~:text=Data%20mining%20is%20the%20process,increase%20sales%2C%20and%20decrease%20costs

https://www.ncbi.nlm.nih.gov/books/NBK499956/

https://www.researchgate.net/publication/349510392_Support_Vector_Machine_And_K-Nearest_Neighbor_Based_Liver_Disease_Classification_Model

https://www.ahajournals.org/doi/10.1161/CIRCULATIONAHA.117.029652

https://www.cdc.gov/heartdisease/heart_attack.htm

https://link.springer.com/referenceworkentry/10.1007/978-3-642-37393-0_70-1

[Cardio Vision]

It's Your Way To a Healthier Life

THANK YOU!