Heart Disease Prediction System

*Students:*

|  |  |
| --- | --- |
| **Abdulrhman Alaqeel** | **421107668** |
| **Mohammed Alaqeel** | **421108062** |
| **Yassir Alabeed** | **411107822** |
| **Thamer Alhammad** | **422112105** |

*Supervisor:*

Dr. Abdulaziz Aldribi

*A project report submitted to Qassim University in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science*

*Qassim - Saudi Arabia*

*1446/1447 (2024/2025)*

**Certificate**

It is certified that this project report has been prepared and written under my direct supervision and guidance. This project report is approved for submission for its evaluation.

*Dr. Abdulaziz Aldribi*

**Dedication**

This project is dedicated to our families, friends, and mentors who supported us throughout our journey. Their encouragement, guidance, and belief in us made this accomplishment possible. We extend our heartfelt gratitude to everyone who inspired us to persevere.

## *Abdulrhman Alaqeel*

## *Mohammed Alaqeel*

## *Yassir Alabeed*

## *Thamer Alhammad*

**Acknowledgement**

We would like to express our special thanks and appreciation to Dr. Abdulaziz Aldribi for supervising our project and providing us with all the help and guidance we needed throughout the whole project. His insights and expertise greatly contributed to the successful completion of this work.

*Abdulrhman Alaqeel*

*Mohammed Alaqeel*

*Yassir Alabeed*

*Thamer Alhammad*

**Contents**

[List of Tables 6](#_Toc185225319)

[List of Figures 7](#_Toc185225320)

[Abstract 8](#_Toc185225321)

[Chapter One 1](#_Toc185225322)

[**Introduction** 1](#_Toc185225323)

[1.1 Introduction 1](#_Toc185225324)

[1.2 Project Scope 2](#_Toc185225325)

[1.3 Aim and Objectives 2](#_Toc185225326)

[1.4 Project Plan and Schedule 3](#_Toc185225327)

[1.5 Outline of the Report 4](#_Toc185225328)

[1.6 Challenges 4](#_Toc185225340)

[Chapter Two 5](#_Toc185225341)

**[Background & Literature Review](#_Toc185225342)** [5](#_Toc185225342)

[2.1 Introduction 5](#_Toc185225343)

[2.2 Background 5](#_Toc185225343)

[2.3 Existing Related Systems 9](#_Toc185225343)

[Chapter Three 12](#_Toc185225344)

[**SYSTEM ANALYSIS & REQUIREMENT SPECIFICATION 12**](#_Toc185225345)

[3.1 Introduction 12](#_Toc185225345)

[3.2 User Requirements 12](#_Toc185225348)

[3.3 System Requirements 13](#_Toc185225348)

[3.4 Actor Goal List 14](#_Toc185225350)

[3.5 Use Cases Diagram 15](#_Toc185225351)

[3.6 Use Cases Description 16](#_Toc185225352)

[3.7 Class Diagram 23](#_Toc185225352)

[3.8 Class Diagram Description 23](#_Toc185225352)

[Chapter Four 26](#_Toc185225353)

[System Design 26](#_Toc185225354)

[4.1 Introduction 26](#_Toc185225355)

[4.2 System Architecture 27](#_Toc185225356)

[4.3 Activity Diagram 30](#_Toc185225357)

[4.4 Database Design 32](#_Toc185225358)

[4.5 Algorithms 32](#_Toc185225359)

[4.6 User Interface Design 38](#_Toc185225360)

[Chapter FIVE 42](#_Toc185225362)

[CONCLUSION & Future work 42](#_Toc185225362)

[5.1 Conclusion 42](#_Toc185225355)

[5.2 Future work 42](#_Toc185225356)

[References 43](#_Toc185225363)

[Appendix 45](#_ihv636)

Appendix A 45

Appendix B 47

Appendix C 49

# List of Tables

[**Table 1:** **Cleveland Heart Disease Dataset 2016 8**](#_vx1227)

[**Table 2: Actor Goal List 14**](#_3fwokq0)

[**Table 3: Use case: Regester 16**](#_41mghml)

[**Table 4:** **Use case: Login 16**](#_vx1227)

[**Table 5: Use case: Log out 17**](#_3fwokq0)

[**Table 6: Use case: Enter symptoms 17**](#_41mghml)

[**Table 7:** **Use case: View Diagnosis 18**](#_vx1227)

[**Table 8: Use case: View medical record 18**](#_3fwokq0)

[**Table 9: Use case: Add user 19**](#_41mghml)

[**Table 10:** **Use case: Delete user 19**](#_vx1227)

[**Table 11: Use case: update user 20**](#_3fwokq0)

[**Table 12: Use case: Contact doctor 20**](#_41mghml)

[**Table 13:** **Use Case: Enter personal information 21**](#_vx1227)

[**Table 14: Use case: Provide medical record 21**](#_3fwokq0)

[**Table 15: Use case: Update personal information 22**](#_41mghml)

[**Table 16: Use case: provide diagnosis 22**](#_41mghml)

[**Table 17: Class Diagram Description 23**](#_41mghml)

# List of Figures

**Figure 1: Gantt Chart of work breakdown structures 3**

[**Figure 2: Heart Disease prediction app requirements**](about:blank) **12**

[**Figure 3: Use case diagram**](about:blank) **15**

[**Figure 4: Class Diagram**](about:blank) **23**

[**Figure 5: System architecture**](about:blank) **27**

[**Figure 6: Prediction Model Subsystem**](about:blank) **28**

[**Figure 7: Doctor Subsystem**](about:blank) **29**

[**Figure 8: Chat Subsystem**](about:blank) **29**

[**Figure 9: Main Subsystem**](about:blank) **30**

[**Figure 10: Activity Diagram**](about:blank) **31**

[**Figure 11: Database Entity Relationship Diagram**](about:blank) **32**

[**Figure 12: Login page**](about:blank) **38**

[**Figure 13: Sign up page**](about:blank) **39**

[**Figure 14: Home page**](about:blank) **40**

# Abstract

Heart disease is a leading global cause of mortality, with many cases being curable if detected early. However, individuals with low-risk symptoms often delay seeking medical attention. This project aims to reduce mortality rates by developing a mobile-optimized web application that simplifies heart disease risk assessment. By analyzing an individual’s medical history and relevant risk factors, the tool provides an accessible and straightforward solution for early detection and intervention.

The problem at hand is the existing fragmented approach to heart disease risk assessment, which often relies on scattered medical records and subjective judgments. Our proposed solution centers around the creation of an intuitive web application, compatible with any smart device, streamlining data collection and management. Users, including healthcare professionals and individuals, can effortlessly input their medical histories and lifestyle details, which will then undergo analysis through advanced machine learning algorithms. The implementation plan comprises four key steps: first, the development of the user-friendly web application; second, the secure storage and management of collected data; third, the application of advanced machine learning models to analyze this data, considering factors such as age, gender, family history, cholesterol levels, blood pressure, and lifestyle choices; and finally, the provision of personalized heart disease risk scores to empower individuals for proactive prevention while assisting healthcare providers in early intervention efforts.

# Chapter One

## Introduction

### 1.1 Introduction

We live in a world where technology is constantly changing, and almost everything around us has become digital. Technology is now an essential part of our daily lives, and we rely on it for many things. It helps improve our daily routines, broaden our knowledge, and manage our time more efficiently. We believe that having access to heart disease detection services at any time is a basic human right. Many heart diseases can only be treated if detected early, but people with mild or low-risk symptoms often don’t visit a doctor. As a result, many individuals lack this important access. Our project aims to create a portable system that takes advantage of the fact that almost everyone carries a smart device, like a tablet, or computer. The goal is to help people predict and detect heart diseases more easily, reducing the time and money spent on medical consultations for occasional, mild symptoms. Even if you don't visit a doctor often, the time spent waiting can add up. Instead, our system lets you quickly input your symptoms and receive instant heart disease detection using data mining and machine learning techniques. The system also decides if seeing a doctor is necessary, providing an extra service to users. Furthermore, you have the option to connect with a real professional doctor through the application to receive a precise prescription.

### 1.2 Project Scope

The scope of this project is to develop a web-based heart disease prediction system, designed to work on computer devices. The system will collect users' medical history and risk factors and analyze the data using advanced machine learning models. These models will give users a risk score for heart disease, helping them decide when to seek medical attention when needed. The system aims to make early heart disease detection more accessible by simplifying the process of entering and analyzing data, which can save both time and money compared to visiting a healthcare provider.

### 1.3 Aim and Objectives

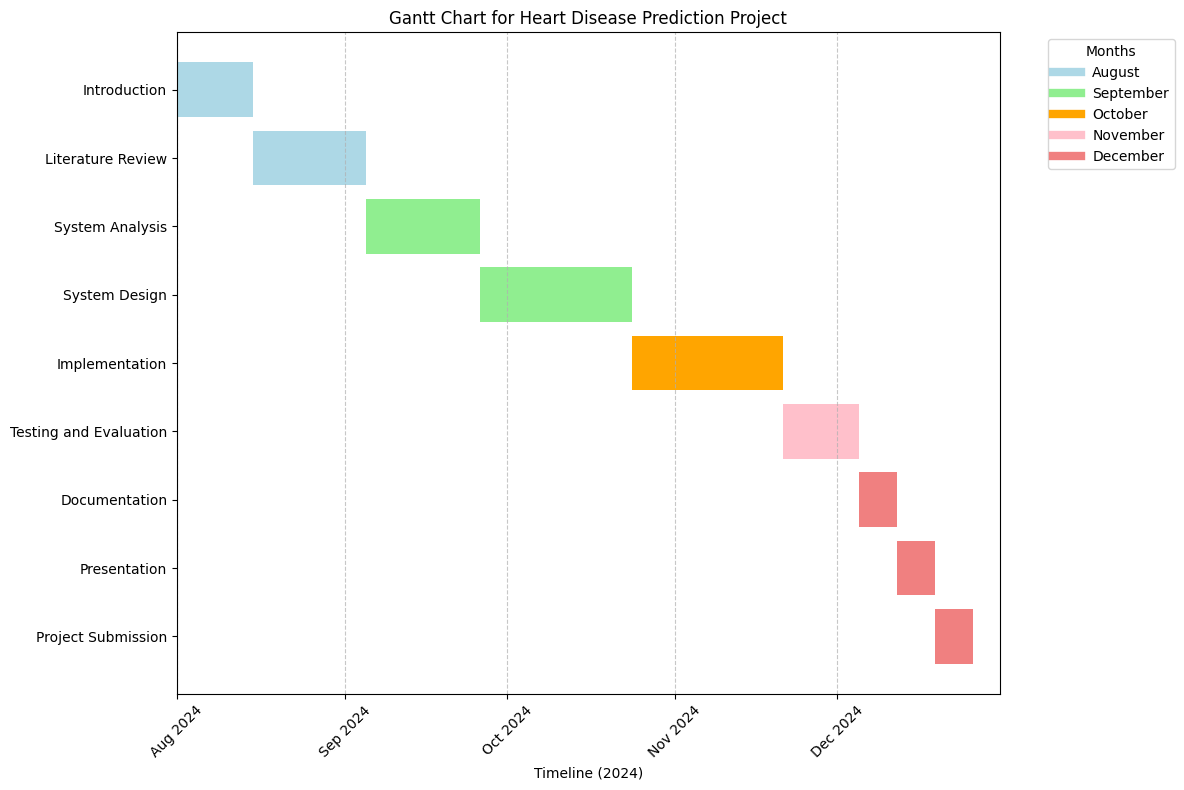
This section will survey the main Aim of the project and the objectives to achieve this Aim of the project**.**

The main aim of this project is to develop a system that can detect heart disease early, which can be accessed on any device with an internet browser. This will allow users to check their risk level and encourage them to see a doctor when necessary.

* To create a heart disease prediction system that can be used on devices (PC or laptop).
* To allow users to input their medical history and symptoms, receiving an accurate risk score based on machine learning models.
* To assist healthcare providers in early detection by providing a reliable tool for initial assessments.

### 1.4 Project Plan and Schedule

Our work had follow the plan shown in the Gantt chart can be shown in Figure1



***Figure 1: Gantt Chart of work breakdown structures***

### 1.5 Outline of the Report

### Our report is organized as follows:

### • Chapter One: Introduction

### Introduces the project, defines the problem and motivation, and discusses the challenges faced during development.

### • Chapter Two: Background and Literature Review

### Provides an overview of related works and discusses existing methods, concepts, and techniques relevant to the project.

### • Chapter Three: System Analysis and Design

### Describes the problem in detail, analyzes the system, specifies its requirements, and presents the overall system design, including machine learning models and algorithms.

### • Chapter Four: Implementation and Results

### Covers the implementation phase, explains the tools and technologies used, and presents the key results and system performance.

### • Chapter Five: Conclusion and Future Work

### Summarizes the project’s contributions, highlights its potential impact, and discusses areas for future research and development to enhance the system further.

### 1.6 Challenges

The project will face several challenges, with one major issue being the lack of high-quality medical data. Quality data is essential for doctors to validate AI models both clinically and technically. To overcome these challenges, we are focused on learning from previous mistakes and doing thorough testing and prototyping before launching the program. Our main goal is to make sure the system meets users' needs and promotes their well-being.

# Chapter Two

## Background & Literature Review

### 2.1 Introduction

Heart disease is a complex condition that affects many people around world. As medicine and technology improve, the leading causes of death have shifted from infectious diseases to those that may not have a cure. This makes early detection of diseases more important, especially for individuals who may have difficulty accessing healthcare, such as the elderly. To help with this, a heart disease prediction system could be very useful. This system would analyze a person’s symptoms and provide predictions, helping those who need it and those who don’t have much time for hospital visits. Patients could have real conversations with a real doctor to get more information about their health.

2.2 Background

2.2.1 Machine Learning

Machine learning, a component of artificial intelligence (AI), empowers software programs to continually enhance their predictive capabilities without the need for explicit programming.[10] The utilization of machine learning in the healthcare sector has gained significant attention, particularly in the realm of disease prediction, where various machine learning algorithms are harnessed to enhance the precision of predictions. In our specific scenario, the application of machine learning is essential for identifying valuable patterns within extensive datasets and accurately categorizing patients' symptoms. Supervised learning, encompassing classification and regression, proves indispensable for forecasting labeled data through a variety of techniques, including several renowned algorithms that we will subsequently elaborate upon. We will use Python for our development of the machine learning model.

2.2.1.1 Naive Bayes

Naive Bayes is a straightforward yet powerful probabilistic machine learning algorithm commonly used for classification tasks. It is grounded in Bayes' theorem, which calculates the probability of a data point belonging to a specific class. A key assumption of the Naive Bayes algorithm is that the features used for classification are independent of one another, which is considered a "naive" assumption. Despite its simplicity, this assumption often leads to effective results in practice. The algorithm works by determining the probability of a given data point being associated with a particular class based on the probabilities of its individual features. Naive Bayes is particularly popular for tasks such as spam email detection and sentiment analysis.

2.2.1.2 Decision Tree

A decision tree is a widely used model in machine learning that represents a decision-making process in a tree-like structure. It is typically employed for classification problems but can also be used for regression tasks. The structure of a decision tree includes nodes that represent the features or attributes of the data, branches that indicate decision rules or conditions, and leaves that provide the final classification or prediction outcomes. By following the path from the root to a leaf, predictions are made based on the characteristics of the input data. Decision trees are valued for their simplicity, interpretability, and effectiveness in capturing complex decision-making patterns, making them useful in a wide range of applications.

2.2.1.3 Performance Metrics

In a classification problem, [13] the model learns to categorize data into different classes based on the training data it receives. used training data to understand patterns and relationships, and then applies this knowledge to classify new data into specific groups or categories. This results in the model making predictions such as labeling data as "Yes" or "No," assigning values like "0" or "1," or categorizing messages as "Spam" or "Not Spam," and so on. To evaluate how well a classification model performs, various metrics are employed, including the following:

Accuracy: accuracy is calculated as the number of all correct predictions of heart disease divided by the total number of the dataset. Accuracy comparison is based on the performance among the four classification algorithms. True Positives (TP): true positives are the cases when the actual class of the data point is True, and the predicted is also True. True Negatives (TN): true negatives are the cases when the actual class of the data point is False, and the predicted is also false. False Positives (FP): false positives are the cases when the actual class of the data point is False, and the predicted is True. False Negatives (FN): false negatives are the cases when the actual class of the data point is True, and the predicted is False.

The formula is:

Accuracy = (TP + TN) ÷ (TP + FP + TN + FN)

Precision: it tells what fraction of predictions of a positive class are actually heart diseases positive. The high precision means the result of the measurements is consistent or the repeated values of the reading are obtained. The low precision means the value of the measurement varies.

The formula is:

Precision = TP ÷ (TP + FP)

Recall: recall refers to a test's ability to designate an individual with heart disease as positive. A highly sensitive test means that there are few false negative results, and thus fewer cases of heart disease are missed.

The formula is:

Recall = TP ÷ (TP + FN)

F1-Score: A binary classification model is evaluated using the F1 Score metric based on the predictions generated for the positive class. With the use of Precision and Recall, it is measured. It is a particular kind of score that combines Precision and Recall. As a result, the F1 Score can be determined by taking the harmonic mean of both precision and recall and giving each variable an equal weight.

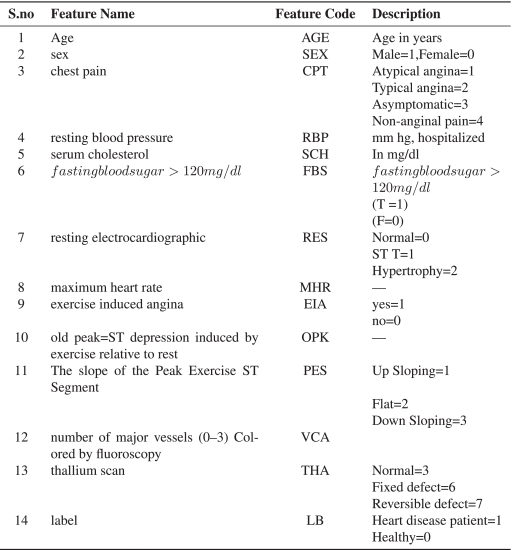
The formula is:

F1-Score = 2\*((Precision \* Recall) ÷ (Precision + Recall).

2.2.2 Dataset

Cleveland Heart Disease  dataset is considered for testing purpose in this study. [14] During the designing of this dataset there were 303 instances and 75 attributes, however all published experiments refer to using a subset of 14 of them. In this work, we performed pre-processing on the data set, and 6 samples have been eliminated due to missing values. The remaining samples of 297 and 13 features dataset is left and with 1 output label. The output label has two classes to describe the absence of HD and the presence of HD. Hence features matrix 297\*13 of extracted features is formed. The dataset matrix information’s are given in Table 1.

***TABLE 1:******Cleveland Heart Disease Dataset 2016***

**

2.3 Existing Related Systems

Jian Ping Li [1] presented an efficient system for diagnosing heart disease using machine learning techniques. The author proposed a system based on various classification algorithms, including Support Vector Machine, Logistic Regression, Artificial Neural Network, K-nearest neighbor, Naïve Bayes, and Decision Tree. The author also employs feature selection algorithms such as Relief, Minimal redundancy maximal relevance, least absolute shrinkage selection operator, and Local learning to remove irrelevant and redundant features. Additionally, they introduced a novel feature selection algorithm called fast conditional mutual information (FCMIM). The goal of this research is to improve classification accuracy and reduce execution time. They apply the leave-one-subject-out cross-validation method for model assessment and hyperparameter tuning, using various performance metrics to evaluate classifier performance. The paper highlighted the importance of feature selection , for improving machine learning model performance, as well as the need for balanced datasets and relevant feature selection.

Sibgha Taqdees [2] focused on the application of machine learning and data mining techniques to predict heart disease. By analyzing various health parameters such as age, gender, cerebral palsy, blood pressure, and fasting blood sugar test results, the study explored the effectiveness of five different algorithms, including Naïve Bayes, k Nearest Neighbor (KNN), Decision tree, Artificial Neural Network (ANN), and Random Forest. The experiments conducted on a dataset from the UCI Machine Learning repository reveal that Naïve Bayes achieved the highest accuracy of 88%, outperforming the other algorithms. The study emphasizes the importance of early detection of heart disease and highlights the potential of data mining techniques in healthcare for accurate decision-making. Overall, the author underscored the significance of leveraging machine learning and data mining in the healthcare industry, offering a promising avenue for improving the early diagnosis and management of heart disease. The author highlighted the potential of these techniques in enhancing healthcare decision-making and service quality.

Md.Imam Hossain [3] focused on improving the prediction and diagnosis of heart disease by leveraging data mining and machine learning techniques. The author aimed to identify the most significant attributes for heart disease prediction and compared various machine learning algorithms for accuracy. The dataset consists of 19 features, including age, gender, smoking, obesity, diet, physical activity, stress, chest pain type, and more collected from patients in Bangladesh. The Correlation-based Feature Subset Selection Technique with Best First Search is used to select the most relevant features. The author evaluated seven machine learning models, including logistic regression, Naïve Bayes, K-nearest neighbor, support vector machine, decision tree, random forest, and multilayer perceptron, using both the dataset with all features and the dataset with selected features. The author found that random forest, when applied to the selected features dataset, achieves the highest accuracy rate of 90% compared to other algorithms and the use of all features. The study highlighted the importance of feature selection in improving heart disease prediction accuracy.

Santosh Shinde [4] explored the application of machine learning methods in predicting health risks and improving healthcare systems. It serves as a guideline for researchers, offering a comprehensive survey of machine learning, its techniques, and their potential applications in health prediction. The paper highlights the significance of electronic health records (EHRs) in improving health risk prediction and suggests that incorporating various encoding techniques, such as the International Classification of Disease (ICD), may enhance predictive models. Additionally, the author suggested directions for future research, including the use of wearable sensors for real- time data collection and prediction of chronic diseases. He also encouraged the evaluation of different models, such as random forests and support vector machines (SVM), for health record predictions and emphasizes the importance of studying various factors and complex disease patterns using deep learning and artificial neural network (ANN) techniques. In conclusion, the author underscored the potential of machine learning in enhancing healthcare and predicting health risks, offering a roadmap for further research in this domain.

Sagar Yeruva [5] studied the application of machine learning algorithms for predicting equipment failures in advance. While reactive and preventive maintenance approaches exist, they have limitations. Detecting a machine's health early can enhance its reliability, efficiency, and availability. Machine learning algorithms play a crucial role in achieving more accurate health predictions for machines. The study collects acoustic signals from the machines, performs signal processing to extract various audio characteristics (such as spectral centroid, spectral bandwidth, spectral Rolloff, Zerocrossing rate, RMSE, MFCC, etc.), and utilizes these features to build predictive models. The author evaluated the performance of several machine learning algorithms, including decision tree, XGBoost, support vector machine, and K-nearest neighbors. Among these, the XGBoost model demonstrates the highest accuracy, followed by the decision tree model. Additionally, the author introduced a user-friendly interface that aids in understanding anomalies and allows users to monitor and comprehend the health status of industrial machines with ease and precision. When applied in real-time, this technique can enhance system performance and resilience while reducing maintenance costs significantly. Future work could involve experimenting with deep learning models, making them adaptable forvarious real-world applications.

# Chapter Three

**System Analysis & Requirement Specification**

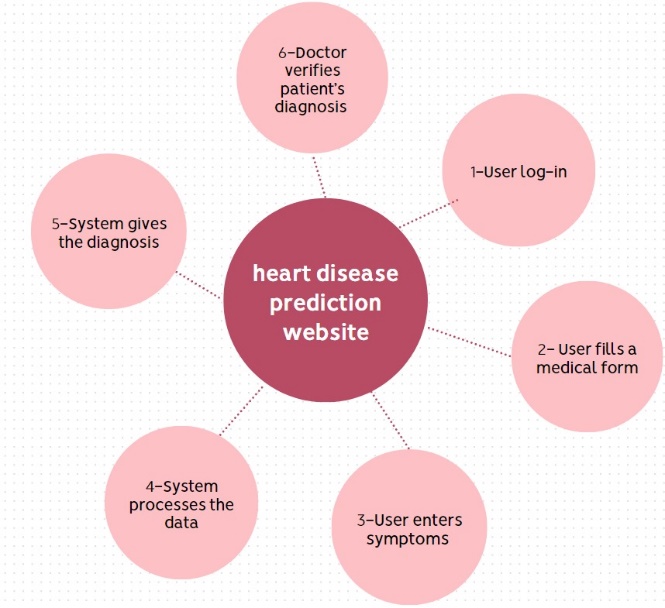
### 3.1 Introduction

In this chapter, we will explore system analysis and specification. This includes identifying user needs, defining both functional and non-functional system requirements, listing actor goals, creating use case diagrams and descriptions, and developing a class diagram.

## 3.2 User Requirements

The heart disease prediction system is designed to make life easier for people. It provides a user-friendly and secure platform optimized for smartphones, helping individuals manage their health and calculate their Body Mass Index (BMI) effortlessly. The system also enables users to get prescriptions from medical professionals and access accurate diagnoses through an intelligent health prediction system powered by machine learning, always ensuring reliable quality and excellent user experience.

***Figure 2:*** ***Heart Disease prediction app requirements***

**

## 3.3 System Requirements

## 3.3.1 Functional Requirements

1. The system must enable users to create an account seamlessly within the platform.
2. Upon creating an account, users should be able to input their medical history.
3. Users must have the ability to log in and out of the system easily.
4. Users should have the capability to update their personal information within the system.
5. The system must display the status of the patient.
6. Users must have the capability to connect with licensed medical practitioners through the system.
7. Users should be able to engage with a medical chat via the system.
8. The system must offer a Body Mass Index calculator service.
9. Users must be able to search for diseases associated with their symptoms using the system.

## 3.3.2 Non-Functional Requirements

1. The system must be compatible with any smart devices.
2. The system must remain on standby and accessible round the clock, 24/7.
3. The system must offer support for both Arabic and English languages.
4. The system should prioritize ease of use for users.
5. The system must rely on reliable and trustworthy datasets.
6. The system's data must undergo frequent backups.
7. The system should be designed to be easily maintainable

## 

## 3.4 Actor Goal List

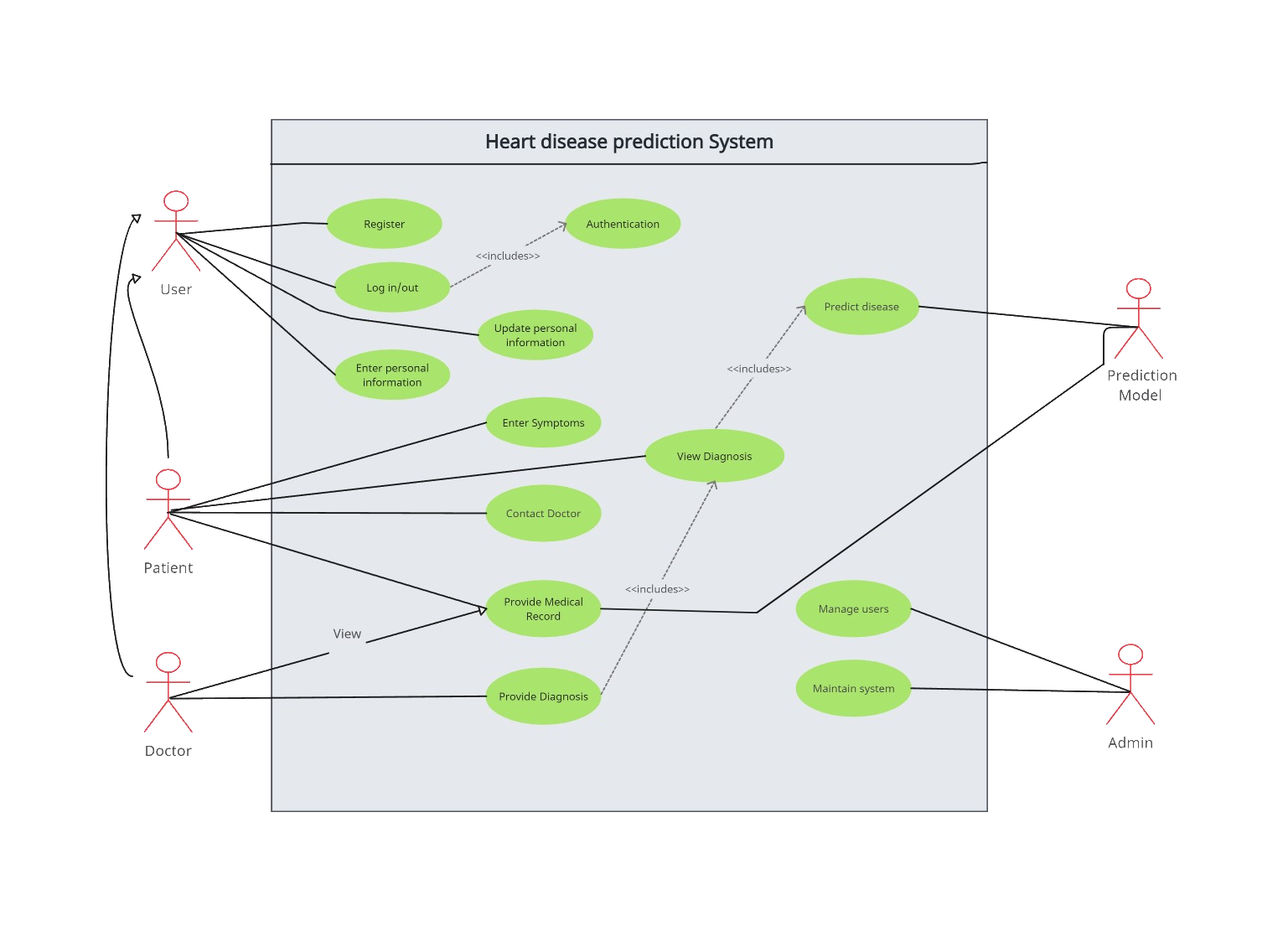
***Table 2: Actor Goal List.***

|  |  |
| --- | --- |
|  |  |
| **Admin: manages and maintains systems, handling tasks like maintaining the system and managing users.** | **● Login**  **● Maintain System**  **● Add User**  **● Delete User**  **● Update User**  **● Logout** |
| **Doctor: licensed volunteer offering medical expertise and advice to assist individuals seeking healthcare within our system.** | ● **Register**  ● **Login**  ● **Enter personal information**  ● **Update personal information**  ● **Provide diagnosis**  ● **View patient's medical record**  ● **Logout** |
| **Patient: the primary recipient of the system is the patient. They input their symptoms and receive a forecast indicating potential diseases they might be experiencing.** | ● **Register**  ● **Login**  ● **Enter personal information**  ● **Update personal information**  ● **Provide medical record**  ● **Enter symptoms**  ● **View predicted disease**  ● **Verify with a doctor**  ● **Logout** |
| **Prediction Model: identify diseases by analyzing the provided symptoms.** | ● **Search a disease** |

## 

## 3.5 Use Cases Diagram

A Use Case Diagram in Unified Modeling Language (UML) is a visual representation that illustrates how a system interacts with external entities, or actors, to accomplish specific tasks or goals. Actors, depicted as stick figures or blocks, represent entities outside the system, while ovals represent use cases, describing the system's functionalities or interactions. Lines connecting actors and use cases indicate associations, outlining the relationships between them. The diagram includes a system boundary, typically a box, delineating what is internal and external to the system. Use Case Diagrams are instrumental for conveying a high-level view of a system's functionality and behavior, making them valuable tools for stakeholders and developers.

***Figure 3: Use Case Diagram.***

## 

## 3.6 Use Cases Description

**Use Cases Name: Register**

Actor(s): Patient, Doctor

Pre-Condition(s): User is not registered in the system.

Post-Condition(s): User is registered in the system

***Table 3: Use Case: Register***

|  |  |
| --- | --- |
| User Action | System Response |
| User enters his/her information to register | System creates the account and saves the information into database. EXP1 |

Exception 1: User data already exists in the database*.*

**Use Cases Name: Login**

Actor(s): Patient, Doctor

Pre-Condition(s): User is registered.

Post-Condition(s): User is logged in*.*

***Table 4: Use Case: Login.***

|  |  |
| --- | --- |
| User Action | System Response |
| User enters the username and password for his/her account | System grants the user access and functionality of the system. **EXP1** |

Exception 1: Invalid username or password

**Use Cases Name: Log Out**

Actor(s): Patient, Doctor.

Pre-Condition(s): User is logged in.

Post-Condition(s): User is logged out.

***Table 5: Use Case: Log out***.

|  |  |
| --- | --- |
| User Action | System Response |
| User Activates the log out by pressing the designated button | System logs out the user. |

**Use Cases Name: Enter symptoms.**

Actor(s): Patient

Pre-Condition(s): User is registered in the system.

Post-Condition(s): User has the diagnosis based on his/her symptoms.

***Table 6: Use Case: Enter symptoms.***

|  |  |
| --- | --- |
| User Action | System Response |
| User enters his/her symptoms and fills the medical record form | 1. If the symptoms are entered the doctor will verify the diagnosis.  2. If the symptoms indicate minimal risk for the patient, the system provides a diagnosis matching those symptoms. |

**Use Cases Name: View Diagnosis**

Actor(s): Patient

Pre-Condition(s): User has entered his medical record and entered his/her personal information.

Post-Condition(s): User can view his/her diagnosis provided by the system.

***Table 7: Use Case: View Diagnosis.***

|  |  |
| --- | --- |
| User Action | System Response |
| User presses view diagnosis. | The system will display the predicted diagnosis provided by the prediction model***.*** |

**Use Cases Name: View medical record.**

Actor(s): Doctor.

Pre-Condition(s): Doctor is logged in and there is a patient in need.

Post-Condition(s): Doctor views Patient's medical history.

***Table 8: Use Case: View medical record***

|  |  |
| --- | --- |
| User Action | System Response |
| Doctor presses the view medical record. | The system grants the doctor to view the medical record. |

***.***

**Use Cases Name: Add user.**

Actor(s): Admin.

Pre-Condition(s): Use exists in the system.

Post-Condition(s): User does not exist in the system.

***Table 9: Use Case: Add user***

|  |  |
| --- | --- |
| User Action | System Response |
| Admin presses add user and enters a name and a password. | 1. System will check if user already exists in the database. EXP1    2. System will add the user in the database. |

Exception 1: User already exists in the system*.*

**Use Cases Name: Delete user.**

Actor(s): Admin

Pre-Condition(s): User exist in the system.

Post-Condition(s): User does not exist in the system

***Table 10: Use Case: Delete user***

|  |  |
| --- | --- |
| User Action | System Response |
| Admin presses delete user and chooses the designated user | System will delete the user from the database**.** |

**Use Cases Name: Update user.**

Actor(s): Admin

Pre-Condition(s): User exist in the system.

Post-Condition(s): User information is updated

***Table 11: Use Case: update user***

|  |  |
| --- | --- |
| User Action | System Response |
| Admin presses update user and chooses the designated user | System will Update the user in the database**.** |

**Use Cases Name: Contact doctor.**

Actor(s): Patient.

Pre-Condition(s): User needs a doctor to give diagnosis.

Post-Condition(s): User fulfills his diagnosis need.

***Table 12: Use Case: Contact doctor.***

|  |  |
| --- | --- |
| User Action | System Response |
| User presses the contact doctor. | System will give the user the ability to verify with a doctor. |

**Use Cases Name: Enter personal information.**

Actor(s): Patient, Doctor

Pre-Condition(s): User is registered in the system.

Post-Condition(s): User's account is finalized

***Table 13:*** ***Use Case: Enter personal information***

|  |  |
| --- | --- |
| User Action | System Response |
| User enters personal information. | System will allow the user to use the application. |

**Use Cases Name: Provide medical record.**

Actor(s): Patient

Pre-Condition(s): User is registered in the system.

Post-Condition(s): User is ready to be diagnosed.

***Table 14: Use Case: Provide medical record.***

|  |  |
| --- | --- |
| User Action | System Response |
| User provides the medical record. | System will grant the user to get the diagnosis. |

**Use Cases Name: Update personal information.**

Actor(s): Patient, Doctor

Pre-Condition(s): Personal information is not updated.

Post-Condition(s): Personal information is updated

***Table 15: Use Case: Update personal information****.*

|  |  |
| --- | --- |
| User Action | System Response |
| User updates his personal information. | System permits the update to the personal information. |

**Use Cases Name: provide diagnosis.**

Actor(s): Doctor.

Pre-Condition(s): Doctor can view medical record.

Post-Condition(s): patient knows the diagnosis*.*

***Table 16: Use Case: provide diagnosis****.*

|  |  |
| --- | --- |
| User Action | System Response |
| Provide the diagnosis to the patient. | System will display the diagnosis to the patient. |

3.7 Class Diagram

A class diagram serves as a visual representation of an application's structure, showcasing the relationships between its various components. It offers a snapshot of how various parts of the application are connected and organized, depicting classes, their attributes, methods, and the associations among them. This diagram provides a clear overview of the application's architecture, aiding developers in understanding its design and facilitating effective decision-making during development and maintenance phases.

A computer screen shot of a computer

Description automatically generated*Figure 4: Class Diagram.*

3.8 Class Diagram Description

***Table 17: Class Diagram Description***.

|  |  |
| --- | --- |
| **Attributes:**  UserName: username for the admin.  Password: password for the admin.  **Methods:**  AddUser (): add users to the system.  UpdateUser (): update users on the system.  DeleteUser (): delete users from the system.  MaintainSystem (): add/delete/update functionality of the system. | **Admin Class:** will manage users, control prediction model, maintains the system. |
| **Attributes:**  Name: name for the user  Password: password for the user.  Email: a valid email for the user.  Phone: a valid phone for the user.  **Methods:**  Register (Name, Password, Email, phone): register new user to the system.  LogIn (): see if the user has logged in or not.  LogOut (): see if the user has logged out or not  **Methods:** ProvideMedicalRecord ():  provide medical record of the patient.  EnterSymptoms (): enter symptoms for the  diagnosis.  ViewDiagnosis (): view the predicted disease based on symptoms.  **Methods:** ProvideDiagnosis (): provide a prescription according to the patient's diagnosis and medical record .  ViewPatientsMedHistory (): view the patient’s  **Attributes:**  Symptoms: list of symptoms a patient has.  Disease: list of diseases a patient may have.  **Methods:**  SearchADisease (): predict a disease based on the symptoms. | **User Class:** will input personal details and make updates as necessary later.  **Patient Class:** will add  access and update the medical  record as needed.  submit symptoms for prediction.  view predicted diseases based  on the symptoms submitted.  **Doctor Class:** Provide  diagnoses to confirmed patients  and access their medical records within the system.  **Prediction model Class:** will search for the matching disease based on the given symptoms. |

# Chapter Four

## System Design

4.1 Introduction

This chapter provides a detailed overview of the platform's design, focusing on its overall system architecture. It includes various diagrams, such as the System Architecture diagram and Activity Diagram, along with key algorithms planned for system implementation. These elements collectively offer a clear understanding of the platform's structure and functionality.

4.2 System Architecture

The platform's system architecture provides a comprehensive elucidation of all system facets, emphasizing the importance of various subsystems and illustrating the seamless integration with the Firebase authentication for secure registration. Furthermore, it demonstrates the interconnections of specific subsystems with a cloud provider, specifically Auto ML for machine learning model training, Also Firebase for backend services and a database to store the dataA diagram of a device

Description automatically generated

***Figure 5: System architecture***

Within Figure 6, the process begins by cleaning the data, ensuring the absence of null values. Subsequently, the data undergoes preparation to build the model. Next, the system initiates model construction before leveraging it to predict the disease*.*

A diagram of a model

Description automatically generated

***Figure 6: Prediction Model Subsystem.***

Within Figure 7, the initial step involves viewing patient information. Subsequently, the system reviews the predicted disease for the patient, after which the final actions involve verifying the patient’s symptoms, having collected all necessary information*.*

A diagram of a medical procedure

Description automatically generated

***Figure 7: Doctor Subsystem.***

Within Figure 8, users have the capability to engage in conversations with an AI doctor.

A diagram of a chat with a cylinder

Description automatically generated

***Figure 8: Chat Subsystem.***

Figure 9 collects and archives the patient's disease and medical details within the database for storage*.*

***Figure 9: Main Subsystem.*** A diagram of a data flow

Description automatically generated

4.3 Activity Diagram

activity diagram representing the workflow of the system. The process begins with users registering on the application, where their account information is validated. Once validated, users provide personal details and, depending on their role, either submit medical records or a medical license. After approval, patients can enter their symptoms into the system, which uses a dataset to analyze and predict potential diseases. If a condition is detected, the system notifies the patient. If additional help is needed, the system connects the patient with doctor, who can prescribe medication based on the reported symptoms.A diagram of a flowchart

Description automatically generated

***Figure 10: Activity Diagram.***

4.4 Database Design

The database design outlines are entities, attributes, and their interconnections in the heart disease prediction system. It illustrates shared attributes facilitating seamless data exchange and correlations. This framework establishes efficient data flow and connectivity, ensuring cohesive system operation.

A diagram of a patient

Description automatically generated

***Figure 11: Database Entity Relationship Diagram.***

4.5 Algorithms

For a heart disease prediction web application with a chat section with a smart bot,[12] the Random Forest Tree algorithm can be a strong choice for the predictive model due to its effectiveness in classification tasks.

**Random Forest Algorithm Overview:**

*Function trainRandomForest(data):*

*# Step 1: Initialize class probabilities*

*classProbabilities ← Compute the probability of each target class in data*

*# Step 2: Calculate mean and standard deviation for each feature per class*

*mean ← {}*

*stdDev ← {}*

*For each feature in data:*

*For each class in target variable:*

*mean[class][feature] ← CalculateMean(feature values for this class)*

*stdDev[class][feature] ← CalculateStdDev(feature values for this class)*

*# Step 3: Store computed class probabilities, mean, and standard deviations*

*Return classProbabilities, mean, stdDev*

*Function predictRandomForest(data, classProbabilities, mean, stdDev):*

*predictedClasses ← []*

*# Step 1: For each instance in the data*

*For each instance in data:*

*probabilities ← {}*

*# Step 2: Calculate probabilities for each class*

*For each class in classProbabilities:*

*probability ← classProbabilities[class] # Start with prior probability*

*# Apply Gaussian probability density function for each feature*

*For each feature in instance:*

*probability ← probability \* GaussianPDF(instance[feature], mean[class][feature], stdDev[class][feature])*

*probabilities[class] ← probability*

*# Step 3: Assign instance to the class with the highest probability*

*predictedClass ← Class with max(probabilities[class])*

*Append predictedClass to predictedClasses*

*Return predictedClasses*

*Function GaussianPDF(x, mean, stdDev):*

*numerator ← exp(-((x - mean)^2) / (2 \* stdDev^2))*

*denominator ← sqrt(2 \* π \* stdDev^2)*

*Return numerator / denominator*

**Website Application Functionalities Overview:**

**User Authentication:**

Allow users to sign up and log in securely.

**Heart Disease Prediction:**

Collect user data (age, gender, blood pressure, cholesterol, etc.). Use the Random Forest model to predict the likelihood of heart disease based on user input. Display the prediction result to the user.

**Chat with a smart bot:**

Implement a chat interface where users can communicate with a smart bot. Allow users to ask questions, discuss concerns, and receive advice regarding heart health. Ensure secure and private communication between the user and the smart bot.

**User Authentication:**

*Function signUp(username, password):*

*# Step 1: Validate input fields*

*If username is empty OR password is empty:*

*Return "Error: Username and password cannot be empty"*

*If username already exists in database:*

*Return "Error: Username already taken"*

*# Step 2: Create user account*

*hashedPassword ← Hash(password) # Hash the password for security*

*Save (username, hashedPassword) to userDatabase*

*Return "Sign-up successful"*

*Function logIn(username, password):*

*# Step 1: Validate input fields*

*If username is not found in userDatabase:*

*Return "Error: Username not found"*

*# Step 2: Validate credentials*

*hashedPassword ← Hash(password)*

*storedPassword ← Retrieve password for username from userDatabase*

*If hashedPassword equals storedPassword:*

*Return "Login successful. Access granted."*

*Else:*

*Return "Error: Invalid credentials"*

**Heart Disease Prediction:**

*# Step 1: Gather user data*

*Function collectUserData():*

*# Step 2: Organize the collected data into a feature vector*

*userData ← []*

*# Step 3: Pass the data to the Random Forest prediction model*

*prediction ← RandomForestModel.predict(userData)*

*# Step 4: Return the prediction result*

*Return prediction*

*Function displayPredictionResult(prediction):*

*# Step 1: Interpret and display the prediction result*

*If prediction = "High Risk":*

*Print("Warning: You may be at high risk for heart disease. Please consult a doctor immediately.")*

*Else If prediction = "Moderate Risk":*

*Print("You may be at moderate risk for heart disease. Consider seeking medical advice.")*

*Else If prediction = "Low Risk":*

*Print("You are at low risk for heart disease. Maintain a healthy lifestyle.")*

*Else:*

*Print("Error: Unable to determine risk. Please provide complete and accurate data.")*

**Chat with a Smart Bot:**

Function openChat():

# Step 1: Initialize the chat interface

Print("Welcome to the Smart Bot Chat!")

Print("You can ask your questions or discuss your concerns here.")

# Step 2: Start the chat loop

While True:

message ← GetInput("You: ") # Prompt user to enter a message

If message = "exit" OR message = "quit":

Print("Chat session ended. Goodbye!")

Break # Exit the chat loop

# Step 3: Send the user message to the smart bot

response ← sendMessage(message)

# Step 4: Display the bot's response

Print("Smart Bot: " + response)

Function sendMessage(message):

# Step 1: Process the user message

processedMessage ← PreprocessMessage(message) # Clean and structure the message for processing

# Step 2: Send the message to the smart bot's backend

response ← SmartBotBackend.process(processedMessage) # Example: AI NLP model or API

# Step 3: Return the bot's response

Return response

This structure outlines key functionalities and provides a basic structure for implementing the Random Forest algorithm and the Web application's features. Actual implementation would involve integrating the Random Forest model, setting up secure communication channels, designing the UI, and handling user inputs and interactions effectively.

4.6 User Interface Design

In this section we are going to provide multiple sections of the user interface designs such as Login, Sign Up Page, Home Page, etc.

### 4.6.1 Login page

It shows the Login page which allows the user to login using his email address and the password. If he forgets his password, he can update his password to a new password by a verification sent to his email address. If he does not have an account, he can register as a new user by clicking sign up.

***Figure 12: Login Page.***

*A screenshot of a login screen

Description automatically generated*

### 4.6.2 Sign Up Page

It shows the Signup page, which allows the user to create a new account by entering their email address and password. If the user already has an account, they can navigate to the Login page to access their account. Additionally, the Signup page may include fields for other user information, such as a username, full name, or phone number. Once the user fills in the required details and submits the form, their account is created, and they can use these credentials to log in.

***Figure 13: Sign up Page.***

*A screenshot of a login form

Description automatically generated*

### 

### 4.6.3 Home Page

Shows the home page of the web application, presenting the available sections: Home, About, Services, and Contact. By pressing on any section, the website will jump you right onto the section needed. Listing valuable information about Why Choose Heart Disease Prediction System.

***Figure 14: Home Page.***

A screenshot of a website

Description automatically generated

### 4.6.4 About page

The About page lists some excellent features and gives information to the user about: Number of doctors, Number of specialties, Smart bot availability, and Model accuracy.

### 

### 4.6.5 Services Section

Shows the Services Section, and it is an overview of all the available services for the registered user. When you click on any service, a new form will appear, and the screen will slide to that form for you to utilize the service efficiently.

### 4.6.6 Contact Section

Displays the Contact section, where users can share their thoughts about the website or report any problems they encounter.

### 4.6.7 AI Doctor

The AI Doctor is an interactive feature designed to provide real-time communication with a smart bot. Users can access it by navigating to the Services page and selecting the Chatbot option. Once activated, the AI Doctor interface allows users to ask questions on a variety of topics, offering instant responses and support.

# Chapter FIVE

**Conclusion & Future Work**

**5.1 Conclusion**

In conclusion, the Heart Disease Prediction System is a significant step toward improving early detection and prevention of heart disease using machine learning technology. The system simplifies the process of assessing heart disease risk, making it accessible to a wide range of users, including patients and healthcare providers. By combining advanced algorithms with a user friendly design, the project demonstrates how technology can be used to address real-world health challenges effectively.

Through this project, we learned the importance of data quality, user interface design, and the challenges of developing an accurate and reliable prediction model. Despite some limitations, such as a lack of diverse data sources, the system provides a solid foundation for future improvements and innovations in e-health.

**5.2 Future Work**

Although our project achieved its main objectives, there are several areas for improvement and future development:

1. **Incorporating Real-Time Data**: Adding support for wearable devices and real-time monitoring to make the system more dynamic and accurate.
2. **Mobile Application Development**: Building a mobile app to make the system more accessible for users who rely on smartphones for healthcare solutions.
3. **Expanding Dataset and Features**: Using larger and more diverse datasets to improve model accuracy and adding more health indicators like physical activity and diet.
4. **Multilingual Support**: Expanding language options to include more users from different regions and backgrounds.
5. **Telemedicine Integration**: Adding features for virtual doctor consultations to enhance the overall user experience.
6. **Health Progress Tracking**: Allowing users to monitor their health over time and receive personalized recommendations based on their data.

By addressing these areas, the Heart Disease Prediction System can become an even more powerful tool for improving healthcare and empowering individuals to take control of their health. This project has been a valuable learning experience, and we hope it inspires further research and development in the field of medical technology.

# References

[1] Jian Ping Li, Amin Ul Haq, Salah Ud Din, Jalaluddin Khan, Asif Khan, Abdus Saboor, “Heart Disease Identification Method Using Machine Learning Classification in E-Healthcare," in IEEE Access, vol. 8, pp. 107562-107582, 2020. Retrieved from

[*https://ieeexplore.ieee.org/abstract/document/9112202*](https://ieeexplore.ieee.org/abstract/document/9112202)

[2] Sibgha Taqdees, Nayab Akhtar, Kanwal Dawood. Heart Disease Prediction. Retrieved from

[*https://www.researchgate.net/publication/349140147\_Heart\_Disease\_Prediction*](https://www.researchgate.net/publication/349140147_Heart_Disease_Prediction)

[3] Imam Hossain, Mehadi Hasan, Ashikur Rahman, Farida Siddiqi, Sharmin Fatema, Sabbir Ejaz, Ahnaf Sad. Heart disease prediction using distinct artificial intelligence techniques: performance analysis and heart disease retrieved from

[*https://link.springer.com/article/10.1007/s42044-023 00148-7*](https://link.springer.com/article/10.1007/s42044-023%2000148-7)

[4] Santosh Shinde, Raja Rajewari. Intelligent health risk prediction systems using machine learning. Retrieved from

[*https://www.researchgate.net/publication/326253594\_Intelligent\_health\_risk\_prediction\_systems\_using\_machine\_learning\_A\_review*](https://www.researchgate.net/publication/326253594_Intelligent_health_risk_prediction_systems_using_machine_learning_A_review)

[5] Sagar Yeruva, Jeshmitha Gunuganti, Sravani Kalva, Surender Reddy, Seong-Cheol Kim. Smart Machine Health Prediction Based on Machine Learning in Industry Environment. Retrieved from

[*https://www.mdpi.com/2078-2489/14/3/181*](https://www.mdpi.com/2078-2489/14/3/181)

[6] Ravi Kumar, P. S. Rao, “Heart Disease Prediction Using Artificial Neural Networks and Genetic Algorithms,” Journal of Health Informatics, vol. 6, no. 4, pp. 235-242, 2019.

[7] Md. Aftab Alam, M. S. Mollah, “A Hybrid Machine Learning Approach for Heart Disease Prediction,” Journal of Computational Biology, vol. 18, pp. 122-134, 2021.

[8] M. Arunachalam, P. Rajalakshmi, “Predictive Analysis of Heart Disease Using Data Mining Algorithms,” Journal of Artificial Intelligence Research, vol. 11, pp. 200-215, 2020.

Firebase, Google

[*https://firebase.google.com/*](https://firebase.google.com/)

[9] Obermeyer, Z., & Emanuel, E. J. (2016). Predicting the future—Big Data, machine learning, and clinical medicine. The New England Journal of Medicine, 375(13), 1216-1219.

[10] Amin, M. S., Chiam, Y. K., & Varathan, K. D. (2019). Identification of significant features and data mining techniques in predicting heart disease. Tehnički vjesnik, 26(1), 149-155.

[11] Kaur, P., Sharma, M., & Mittal, M. (2019). Big data and machine learning-based secure healthcare framework. Procedia Computer Science, 132, 1049-1059.

[12] Polat, H., & Güneş, S. (2007). A novel hybrid intelligent method based on C4.5 decision tree classifier and one-against-all approach for multi-class classification problems. Expert Systems with Applications, 36(8), 11051-11059.

[13] Eysenbach, G. (2001). What is e-health? Journal of Medical Internet Research, 3(2), e20.

[14] Kaggle, Heart diseases dataset

[*https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset*](https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset)

# Appendix

### Appendix A:

#Sign up page HTML code:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Sign Up</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f4f4f4;

}

.container {

max-width: 400px;

margin: 50px auto;

background: white;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

padding: 20px;

}

h2 {

text-align: center;

color: #f44336;

}

form {

display: flex;

flex-direction: column;

}

form label {

margin-bottom: 5px;

font-weight: bold;

}

form input {

margin-bottom: 15px;

padding: 10px;

border: 1px solid #ccc;

border-radius: 5px;

}

form button {

background-color: #f44336;

color: white;

border: none;

padding: 10px;

font-size: 16px;

border-radius: 5px;

cursor: pointer;

}

form button:hover {

background-color: #d32f2f;

}

.footer {

text-align: center;

margin-top: 20px;

}

.footer a {

color: #2196f3;

text-decoration: none;

font-weight: bold;

}

.footer a:hover {

text-decoration: underline;

}

</style>

</head>

<body>

<div class="container">

<h2>Sign Up</h2>

<form action="/signup" method="post">

<label for="name">Full Name:</label>

<input type="text" id="name" name="name" required>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required>

<label for="confirm\_password">Confirm Password:</label>

<input type="password" id="confirm\_password" name="confirm\_password" required>

<button type="submit">Sign Up</button>

</form>

<div class="footer">

<p>Already have an account? <a href="/login">Login</a></p>

</div>

</div>

</body>

</html>

### Appendix B:

#Login page HTML code:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Login</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f4f4f4;

}

.container {

max-width: 400px;

margin: 50px auto;

background: white;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

padding: 20px;

}

h2 {

text-align: center;

color: #f44336;

}

form {

display: flex;

flex-direction: column;

}

form label {

margin-bottom: 5px;

font-weight: bold;

}

form input {

margin-bottom: 15px;

padding: 10px;

border: 1px solid #ccc;

border-radius: 5px;

}

form button {

background-color: #f44336;

color: white;

border: none;

padding: 10px;

font-size: 16px;

border-radius: 5px;

cursor: pointer;

}

form button:hover {

background-color: #d32f2f;

}

.footer {

text-align: center;

margin-top: 20px;

}

.footer a {

color: #2196f3;

text-decoration: none;

font-weight: bold;

}

.footer a:hover {

text-decoration: underline;

}

</style>

</head>

<body>

<div class="container">

<h2>Login</h2>

<form action="/login" method="post">

<label for="email">Email:</label>

<input type="email" id="email" name="email" required>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required>

<button type="submit">Login</button>

</form>

<div class="footer">

<p>Don't have an account? <a href="/signup">Sign Up</a></p>

</div>

</div>

</body>

</html>

### 

### Appendix C:

#Home page HTML code:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Heart Disease Prediction System</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

scroll-behavior: smooth;

}

.header {

background-color: #f44336;

color: white;

padding: 1em 0;

position: sticky;

top: 0;

z-index: 1000;

}

.navbar ul {

list-style: none;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

}

.navbar li {

margin: 0 15px;

}

.navbar a {

color: white;

text-decoration: none;

font-weight: bold;

}

.navbar a:hover {

text-decoration: underline;

}

section {

padding: 50px;

text-align: center;

}

.home {

background: #e91e63;

color: white;

}

.about, .services, .contact {

background: #f4f4f4;

margin-top: 20px;

}

.cta {

display: inline-block;

margin-top: 20px;

padding: 10px 20px;

color: white;

background: #2196f3;

text-decoration: none;

border-radius: 5px;

}

.cta:hover {

background: #1976d2;

}

footer {

background: #333;

color: white;

text-align: center;

padding: 20px 0;

}

form {

display: inline-block;

text-align: left;

}

form label {

display: block;

margin-bottom: 5px;

}

form input, form textarea, form button {

width: 100%;

margin-bottom: 15px;

padding: 10px;

border: 1px solid #ccc;

border-radius: 5px;

}

form button {

background-color: #f44336;

color: white;

border: none;

cursor: pointer;

}

form button:hover {

background-color: #d32f2f;

}

</style>

</head>

<body>

<header class="header">

<nav class="navbar">

<ul>

<li><a href="#home">Home</a></li>

<li><a href="#about">About</a></li>

<li><a href="#services">Services</a></li>

<li><a href="#contact">Contact</a></li>

</ul>

</nav>

</header>

<section id="home" class="home">

<h1>Welcome to the Heart Disease Prediction System</h1>

<p>Your health is our priority. Discover how our system can assist you in monitoring and predicting heart health effectively.</p>

<a href="#about" class="cta">Get Started</a>

</section>

<footer class="footer">

<p>&copy; 2024 Heart Disease Prediction System. All Rights Reserved.</p>

</footer>

</body>

</html>