

**Faculty of Computing & IT** **University of Sialkot**

**Expert System in Health Care**

**Session: BS-CS Fall 2019-2023**

**Project Advisor: Shaista Irum**

**Submitted By**

|  |  |
| --- | --- |
| *Umer Shabbir* | *19101002-033* |
| *M.Arslan* | *19101002-184* |
| *M.Saqib* | *19101002-032* |

Faculty of Computing & IT University of Sialkot

**STATEMENT OF SUBMISSION**

This is certified that **M. Arslan Roll No 19101002-184**.And **Umer Shabbir Roll No *19101002-033*** **M. Saqib Roll No. 19101002-032** have successfully completed the final year project named as **Expert System in Healthcare** at the Department of Computer Science, University of Sialkot, to fulfill the requirement of the degree of **BS in Computer Science**.

\_

Project Supervisor Project Management Office Faculty of C&IT -USKT

Head of the Department

**Acknowledgement**

We truly acknowledge the cooperation and help make by , Chairman, Department of Information Technology, University of Sialkot. He has been a constant source of guidance throughout the course of this project. We would also like to thank Ms. Shaista Irum for his help and guidance throughout this project. We are also thankful to our friends and families whose silent support led us to complete our project.

1. M.Arslan 19101002-184
2. Umer Shabbir 19101002-033
3. M.Saqib 19101002-032

Date: 29-5-2023

**Abstract**

The Expert prediction system is a web application developed using Python's Stream lit library. The system aims to predict various diseases based on trained machine learning models. The models utilized in this project have been trained on large datasets, enabling accurate predictions.

The procedure for the system involves importing the necessary libraries and datasets. The data is then split into features and target variables, followed by further division into training and testing sets. The machine learning models are trained on the training data, and the trained models are saved using the pickle module for future use.To provide a user-friendly interface, Streamlit is used to create a web application. Users can input their own data and make disease predictions using the saved machine learning models. The application includes a button that processes user input and generates predictions. The Expert prediction system addresses the need for efficient and accurate disease prediction, contributing to early detection and timely medical intervention. The utilization of machine learning techniques and a user-friendly interface enhances the accessibility and usability of the system. Overall, the project aims to improve disease prediction and support healthcare professionals in making informed decisions. The system has the potential to assist in the early diagnosis and treatment of diseases, thereby improving patient outcomes and reducing healthcare costs

**Table of Content**

[**CHAPTER 1: PROJECT FEASIBILITY REPORT 7**](#_bookmark0)

* 1. [INTRODUCTION 8](#_bookmark1)
  2. [PROBLEM STATEMENT 8](#_bookmark2)
  3. [OBJECTIVES 8](#_bookmark3)
  4. [PROJECT MOTIVATION 9](#_bookmark4)
  5. [PROJECT/PRODUCT FEASIBILITY REPORT 10](#_bookmark5)
     1. [Technical Feasibility 10](#_bookmark6)
     2. [Operational Feasibility 10](#_bookmark7)
     3. [Economic Feasibility 11](#_bookmark8)
     4. [Schedule Feasibility 11](#_bookmark9)
     5. [Specification Feasibility 11](#_bookmark10)
     6. [Motivational Feasibility 11](#_bookmark12)
     7. [Legal & Ethical Feasibility 11](#_bookmark13)
  6. [TASK DEPENDENCY TABLE 12](#_bookmark19)
  7. [INTRODUCTION TO TEAM MEMBER AND THEIR SKILL SET 12](#_bookmark22)
  8. [TASK AND MEMBER ASSIGNMENT TABLE 13](#_bookmark23)
  9. TASK DURATION AND DEPENDENCIES…………………………………………………….14
  10. [TOOLS AND TECHNOLOGY WITH REASONING 14](#_bookmark24)
  11. [RISK LIST 15](#_bookmark26)
  12. [PRODUCT FEATURES/ PRODUCT DECOMPOSITION 16](#_bookmark27)

[**CHAPTER 2: SOFTWARE REQUIREMENTS AND SPECIFICATIONS (FOR**](#_bookmark51)[**STRUCTURED APPROACH) 17**](#_bookmark51)

* 1. [INTRODUCTION 18](#_bookmark52)
  2. [ARCHITECTURE PATTERN 19](#_bookmark54)
  3. [ALGORITHM FLOW CHART 20](#_bookmark55)
  4. [SYSTEM SEQUENCE DIAGRAM 21](#_bookmark56)
  5. [SEQUENCE DIAGRAM PATIENT 22](#_bookmark57)
  6. [COLLABORATION DIAGRAM 23](#_bookmark62)
  7. [USE CASE DIAGRAM ADMIN 24](#_bookmark66)
  8. [USE CASE DIAGRAM PATIENT 25](#_bookmark66)
  9. [USE CASE REGISTRATION DESCRIPTION 26](#_bookmark66)

2.10. USE CASE LOGIN DESCRIPTION 27

2.11. USE CASE HEART DISEASE PREDICTION DESCRIPTION 28

2.11. USE CASE DIABETES DISEASE PREDICTION DESCRIPTION 29

2.11. USE CASE DOCTOR VIEW DESCRIPTION 30

[**CHAPTER 3: USER INTERFACE DESIGN 31**](#_bookmark76)

* 1. [SITE MAPS 32](#_bookmark78)
  2. [STORY BOARDS 33](#_bookmark79)

REGISTRATION 34

LOGIN 35

HOME PAGE 36

HEART DISEASE 37

DIABETES 38

DOCTORS 39

* + 1. [PROTOTYPE 40](#_bookmark80)

[**CHAPTER 4: SOFTWARE TESTING 41**](#_bookmark84)

[4.1. INTRODUCTION 42](#_bookmark85)

* 1. [BLACK BOX PLAN/WHITE BOX PLAN/GREY BOX PLAN 42](#_bookmark86)
  2. [WHITE BOX APPROACH 43](#_bookmark87)
  3. TEST PLAN 44
  4. [TEST LOG 44](#_bookmark102)

4.5 [REGISTRATION 44](#_bookmark102)

* 1. [LOGIN 45](#_bookmark102)
  2. [LOGOUT 45](#_bookmark102)
  3. [DIABETES 46](#_bookmark102)
  4. HEART DISEASE 46

4.10. DOCTOR VIEW 47

4.11. HEART DISEASE 47

4.12. DIABESTES DISEASE 48

4.13. DISEASE 48

[**REFERENCES 49**](#_bookmark111)

[**APPENDIX 1: FINAL DOCUMENTATION FORMAT GUIDELINES 49**](#_bookmark112)

**Chapter 1: Project Feasibility Report**

* 1. ***Introduction***

The Expert prediction system is an innovative web application developed using Python's Streamlit library. This system aims to predict various diseases based on user-provided data. By leveraging machine learning algorithms trained on large datasets, the application offers users the ability to assess their health conditions and identify potential diseases.

The system follows a simple and intuitive procedure. First, it imports the necessary libraries and datasets required for the prediction process. Then, the data is split into features and target variables to prepare it for model training. The dataset is further divided into training and testing sets to evaluate the model's performance.

Next, the disease prediction model is trained using the training data. The trained model is then saved using the pickle module, ensuring that it can be reused for future predictions without retraining.

To provide a user-friendly interface, the system utilizes Streamlit to create a web application. This application allows users to input their own health data, such as symptoms or medical history, and obtain predictions based on the saved model. The interface includes a button that processes the user input and generates predictions.

Overall, the Expert prediction system combines the power of machine learning, large datasets, and user-friendly web interfaces to enable early disease detection and improve health outcomes. It empowers individuals to take proactive measures towards their well-being and seek timely medical interventions if necessary.

* 1. ***PROBLEM STATEMENT:***

The problem addressed by the Expert prediction system is the need for an efficient and accurate method to predict various diseases based on individual health data. Traditional diagnostic approaches may require extensive medical testing and expertise, leading to delays in disease identification and treatment. This project aims to develop a web application that leverages machine learning algorithms to provide timely and accurate disease predictions, enabling early intervention and better health management.

* 1. **OBJECTIVES**:

The objectives of the Expert prediction system are as follows:

Develop a web application that allows users to input their health data and obtain disease predictions.

Train machine learning models using large datasets to achieve accurate disease predictions.

Implement a user-friendly interface that enables easy data input and result visualization.

Improve early disease detection by providing timely predictions, assisting users in proactive health management.

Ensure the security and privacy of user data by implementing robust data protection measures.

* 1. ***Project Motivation***

The motivation behind developing the Expert prediction system stems from the critical need for accurate and timely disease diagnosis. The current healthcare landscape faces challenges such as misdiagnosis, delayed treatment, and limited access to healthcare services, which can have severe implications for patients' well-being.

The project aims to address these challenges by harnessing the power of machine learning and data analysis to enable early disease detection and intervention. By providing accurate predictions based on user-provided symptoms, the system empowers individuals to take proactive steps towards managing their health effectively.

The motivation for this project is to:

**Improve Healthcare Outcomes:**

Early disease detection is crucial for effective treatment and better patient outcomes. By accurately predicting multiple diseases based on symptoms, the system enables timely medical intervention, potentially saving lives and improving healthcare outcomes.

**Enhance Access to Healthcare:**

The web-based nature of the system ensures that individuals can access disease prediction services from anywhere, breaking barriers of geographical limitations and providing healthcare support, especially to underserved communities.

**Reduce Healthcare Costs**:

Misdiagnosis and delayed treatment contribute to increased healthcare costs. By providing accurate disease predictions, the system helps in avoiding unnecessary medical procedures and facilitates cost-effective healthcare management.

**Empower Individuals:**

By providing personalized predictions and recommendations, the system empowers individuals to actively participate in their healthcare management. It promotes awareness, encourages preventive measures, and enables informed decision-making regarding seeking medical attention.

**Support Healthcare Professionals:**

The system serves as a valuable tool for healthcare professionals, assisting them in making accurate diagnoses and providing appropriate treatment plans. It complements their expertise and helps streamline the diagnostic process.

The motivation behind the Expert prediction system is to leverage technology to overcome the limitations of traditional diagnostic methods, improve healthcare outcomes, and empower individuals to take control of their health. By combining the power of machine learning and data analysis, the system strives to make a positive impact in the healthcare domain and contribute to proactive healthcare management.

* 1. ***Project Feasibility Report***

There are many types of feasibilities:

* + - Technical
    - Operational
    - Economic
    - Schedule
    - Specification
    - Information
    - Motivational
    - Legal and Ethical
    1. **Technical Feasibility**

The project utilizes machine learning algorithms and data analysis techniques to predict multiple diseases based on user-provided medical data and symptoms. It involves the development of a web application using technologies such as Python, Django, and machine learning libraries.

* + 1. **Operational Feasibility**

He system allows users to register, input their medical data, and receive predictions for possible diseases. It provides an interface for administrators to manage user accounts, update prediction models, and view user data. The system aims to improve disease prediction accuracy and assist in early diagnosis.

* + 1. **Economic Feasibility**

The project has potential economic benefits by reducing the time and effort required for manual disease diagnosis. Early detection of diseases can lead to cost savings in healthcare by enabling timely intervention and treatment. Additionally, it can enhance the efficiency of medical professionals by automating the prediction process.

* + 1. **Schedule Feasibility**

The project development timeline and milestones may vary based on the project scope and available resources. A typical schedule may involve tasks such as requirement analysis, system design, implementation, testing, and deployment. Regular progress monitoring and coordination among team members are essential for timely completion

* + 1. **Specification Feasibility**

The system should support user registration and authentication, data upload in specific formats, disease prediction using machine learning algorithms, displaying prediction results to users, and providing administrative functionalities such as user management and model updates. It should ensure data privacy and security.

* + 1. **Motivational Feasibility**

The project aims to enhance disease prediction accuracy, leading to early detection and intervention, which can potentially save lives and improve patient outcomes. It empowers individuals by providing them with insights into their health conditions and promotes proactive healthcare management.

* + 1. **Legal & Ethical Feasibility**

The project should comply with applicable laws and regulations regarding data privacy, security, and healthcare practices. It should obtain informed consent from users for using their medical data and ensure transparency in how the data is used. Ethical considerations include maintaining fairness, avoiding bias, and handling sensitive health information responsibly.

* 1. ***Task Dependency Table***

| **Task ID** | **Task Description** | **Predecessor Tasks** |
| --- | --- | --- |
| T1 | Import necessary libraries | - |
| T2 | Load datasets | - |
| T3 | Split data into features | T2 |
| T4 | Split data into targets | T2 |
| T5 | Split data into train/test | T3, T4 |
| T6 | Train machine learning models | T5 |
| T7 | Save models using pickle | T6 |
| T8 | Create Streamlit web app | - |
| T9 | Design user input interface | T8 |
| T10 | Implement prediction button | T9 |
| T11 | Process user input | T10 |
| T12 | Generate disease predictions | T11, T7 |
| T13 | Test and debug the system | T12 |

**Table: 1.1. Task Dependency**

* 1. ***Introduction to Team member and their skill set***

1. ***Member 1 – Umer Shabbir***

Role: Project Manager

Skill Set: Strong leadership skills, excellent organizational and communication skills, experience in project planning, Designing and coordination, proficient in documentation and project management tools.

1. **Member 2 – M.Arslan**

Role: Backend Developer

Skill Set: Proficient in Python, experienced in web development frameworks (e.g., Django), database management (e.g., SQL), knowledge of RESTful APIs, familiarity with software development best practices.

1. ***Member 3 – Muhammad Saqib***

Role: Frontend Developer

Skill Set: Proficient in HTML, CSS, and JavaScript, experience with frontend frameworks (e.g., React, Vue.js), knowledge of user interface design principles, familiarity with responsive design and cross-browser compatibility.

Together, our team combines project management expertise, backend development skills, and frontend development capabilities to successfully execute the project. We collaborate closely, leveraging our respective skills to ensure the smooth development and delivery of the Multiple Disease Prediction System Web App.

* 1. ***Task and Member Assignment Table***

| **Task ID** | **Task Description** | **Assigned Member** |
| --- | --- | --- |
| T1 | Importing necessary libraries and datasets | Umer Shabbir |
| T2 | Data preprocessing | Muhammad saqib |
| T3 | Data splitting into features and targets | Muhammad Arslan |
| T4 | Data splitting into training and testing | Muhammad Arslan |
| T5 | Model training | Umer Shabbir |
| T6 | Model evaluation | Muhammad Arslan |
| T7 | Model saving using pickle | Muhammad saqib |
| T8 | Web application development | Muhammad saqib |
| T9 | User interface design | Umer Shabbir |
| T10 | User input processing and prediction | Muhammad Arslan |
| T11 | Database setup and management | Muhammad saqib |
| T12 | Data integration | Muhammad Arslan |
| T13 | Data retrieval and storage | Umer Shabbir |

**Table: 1.2. Task and Member Assignment**

* 1. **Task durations and dependencies**

| **Task** | **Duration (weeks)** | **Dependencies** |
| --- | --- | --- |
| Define project scope | 2 | None |
| Gather requirements | 2 | Define project scope |
| Design system | 4 | Gather requirements |
| Develop code | 7 | Design system |
| Test system | 3 | Develop code |
| Deploy system | 2 | Test system |
| Train users | 1 | Deploy system |
| Implement sentiment analysis | 4 | Develop code, Test system |

**Table: 1.3. Task duration and dependencies**

* 1. ***Tools and Technology with reasoning***

**Front End:**

Streamlit: Streamlit is used for creating the interactive web application interface where users can input their data and receive disease predictions.

**Back End:**

1. Python: Python is used as the programming language for the back-end development, including data processing, model training, and prediction generation.
2. Machine Learning Libraries: Libraries like scikit-learn and TensorFlow are used for training and deploying machine learning models for disease prediction.
3. Database Management System: A database management system, such as MySQL, is used for storing and managing large datasets used in training the prediction models.
4. Git and GitHub: Git and GitHub are used for version control, collaboration, and code sharing among team members during the development process.
5. Integrated Development Environment (IDE): IDEs like PyCharm or Visual Studio Code can be used for efficient coding, debugging, and project management.

These tools and technologies work together to create a comprehensive and functional system that combines front-end user interaction with back-end data processing and prediction generation.

* 1. ***Risk List***

Data Security Breach: There is a risk of unauthorized access or data breaches that could compromise the privacy and confidentiality of user data. Robust security measures, encryption techniques, and regular vulnerability assessments must be implemented to mitigate this risk.

Inaccurate Predictions: The machine learning algorithms may provide inaccurate predictions due to limitations in the training dataset or the complexity of certain diseases. Continuous evaluation, testing, and improvement of the prediction models are essential to minimize this risk.

Lack of Sufficient Data: The availability of a comprehensive and diverse dataset is crucial for training the machine learning models effectively. There is a risk of insufficient data or biased data, which can lead to inaccurate predictions. Collaborations with healthcare institutions and data collection efforts can help mitigate this risk.

Regulatory Compliance: The project must comply with relevant data protection and privacy regulations, such as GDPR or HIPAA, depending on the jurisdiction. Non-compliance could result in legal and financial consequences. Thorough understanding of the regulations and implementation of necessary measures are essential to manage this risk.

User Adoption and Acceptance: User acceptance and adoption of the Expert prediction system may be a challenge. It is essential to provide user-friendly interfaces, clear instructions, and educational resources to promote user engagement and trust in the system.

Ethical Considerations: The use of personal health data raises ethical concerns regarding informed consent, data anonymization, and responsible use. Clear policies and guidelines must be established to address these ethical considerations and ensure transparency and accountability.

Integration with Healthcare Systems: Integrating the system with existing healthcare systems or electronic health record (EHR) systems may pose technical challenges and compatibility issues. Effective coordination and communication with healthcare providers and IT teams are necessary to mitigate this risk.

System Scalability: As the user base and data volume grow, there is a risk of the system becoming overwhelmed and experiencing performance issues. Scalability considerations, such as load testing and infrastructure scaling, must be taken into account to ensure smooth system operation.

It is important to continuously assess and monitor these risks throughout the project lifecycle and implement appropriate risk mitigation strategies to ensure the success and effectiveness of the Multiple Disease Prediction System.

* 1. ***Features***

**Disease Prediction:**

The system utilizes machine learning models trained on large datasets to predict various diseases based on user input and medical data. It provides users with an accurate assessment of potential health conditions.

**User-Friendly Interface:**

The web application offers an intuitive and user-friendly interface, making it easy for users to input their data, view the results, and navigate through the system.

**Data Privacy and Security:**

The system ensures the privacy and security of user data by implementing robust data encryption, access controls, and adherence to relevant data protection regulations. Users can trust that their personal and medical information is handled with utmost confidentiality.

**Scalability:**

The system is designed to handle a large number of users and data inputs without compromising performance. It can effectively scale up to accommodate increasing user demands.

Easy Integration:

The web application can be easily integrated with existing healthcare systems, allowing healthcare professionals to access and analyze user data for further medical evaluation and treatment planning

**Chapter 2: Software Requirements and Specifications**

**2.1. Introduction:**

The Multiple Disease Prediction Web App is a Python-based project developed using Streamlit library. It leverages machine learning models trained on large datasets to predict various diseases. With a user-friendly interface, the web app allows users to input their data and receive disease predictions.

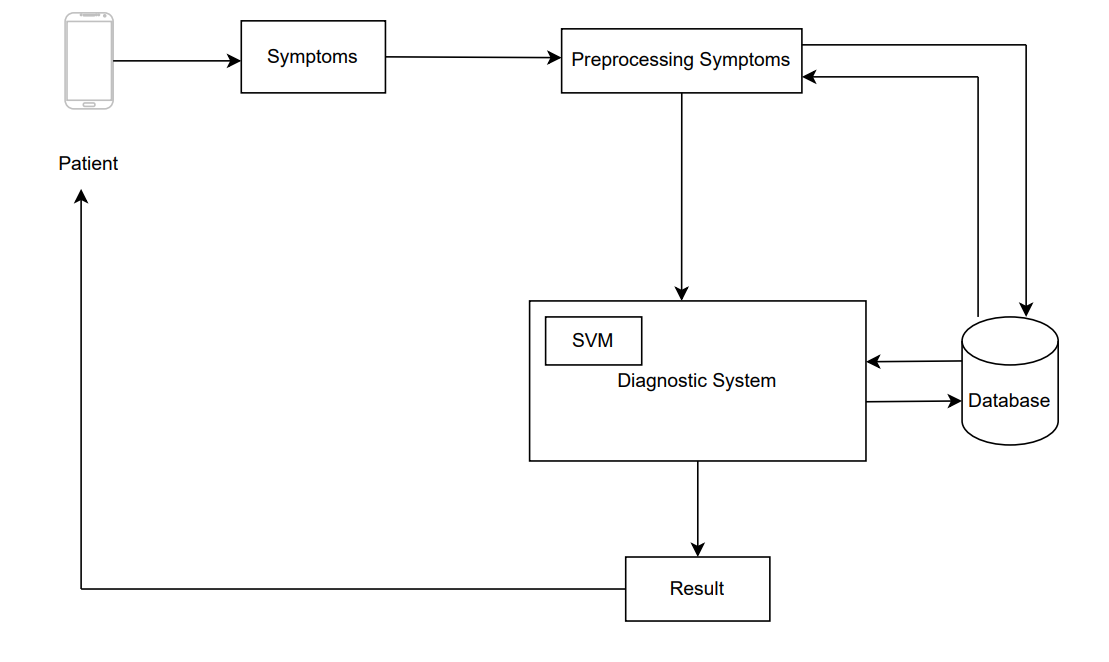
The project's main objective is to provide an efficient platform for disease prediction, empowering users with timely and accurate insights into their health. By utilizing advanced machine learning techniques, the web app aims to enable early detection and proactive management of diseases.

The development of this web app offers several advantages, including personalized healthcare recommendations and improved accessibility to healthcare services. It has the potential to enhance health outcomes by enabling users to take proactive measures for disease prevention and management.

Overall, the Multiple Disease Prediction Web App serves as a valuable tool for individuals to gain insights into their health status and make informed decisions regarding their well-being.

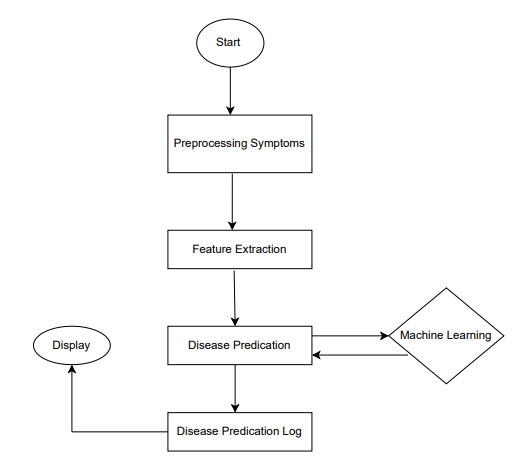
* 1. Architecture Pattern
  2. Algorithm Flow Chart
  3. Sequence Diagram admin
  4. Sequence Diagram patient
  5. Collaboration Diagram
  6. Use case diagram admin
  7. Patient use case diagram
  8. Use case registration description
  9. Use case login description
  10. Use case heart disease prediction description
  11. Use case diabetes disease prediction description
  12. Use case view doctor description

* 1. **Architecture Pattern**



**Figure: 2.1. Architecture Pattern**

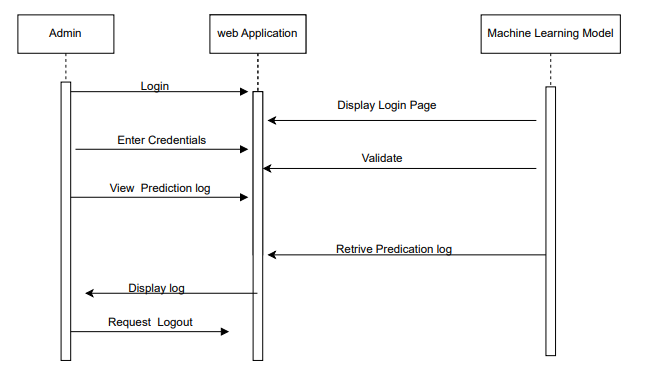
* 1. ***Algorithm Flow Chart***



**Figure: 2.2. Algorithm Flow chart**

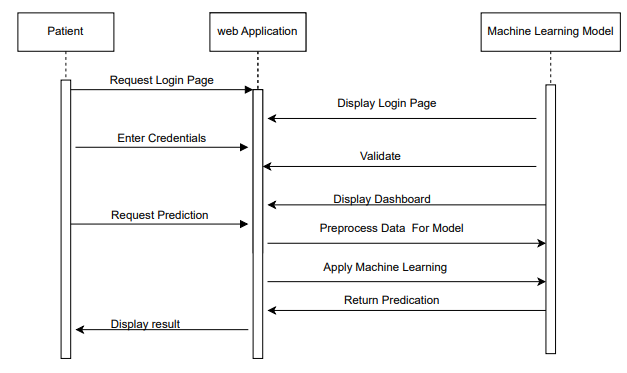
***2.4. System Sequence Diagram***

In this enhanced sequence diagram, the admin can perform. The "Add Disease Data" scenario shows the admin entering disease information, which is then saved to the database. The "Manage Users" scenario demonstrates the admin viewing a list of users, selecting a user, managing their details, and updating the user data in the database.



**Figure: 2.3. System Sequence Diagram**

* 1. ***Patient Sequence Diagram***

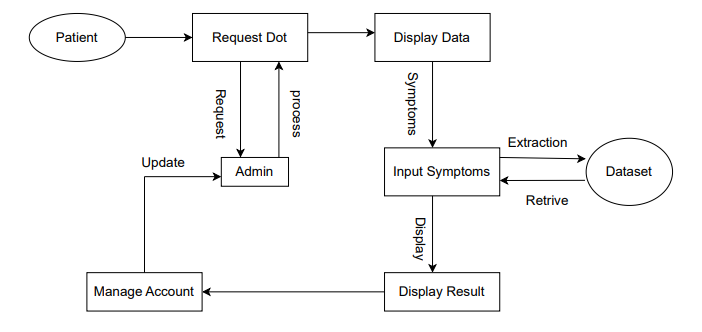
******

**Figure: 2.4. Patient Sequence Diagram**

***2.6. Collaboration Diagram***

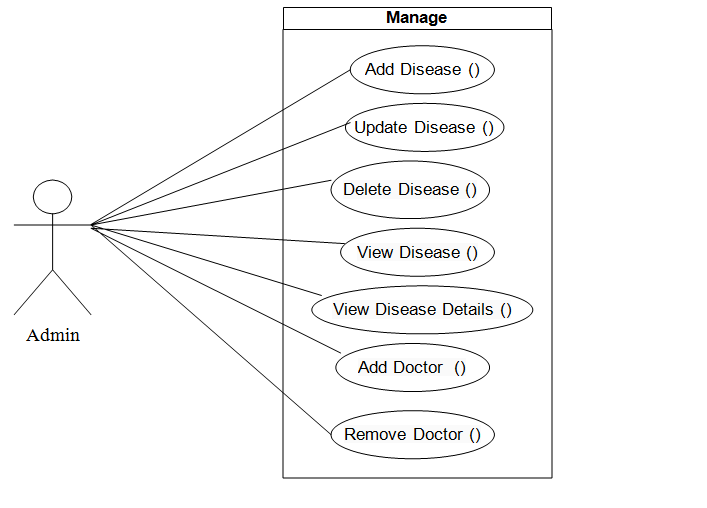
In this collaboration diagram, the "Admin" interacts with the "Web Application" by logging in, accessing the admin panel, managing patient data, and training the disease prediction model. The "Admin" communicates with the "Disease Prediction Model" by initiating the training process and the "Web Application" by accessing and loading the trained model.

On the other hand, the "Patient" interacts with the "Web Application" by accessing it, entering personal details, selecting and choosing symptoms, and submitting the symptom selections. The "Web Application" communicates with the "Disease Prediction Model" by sending the symptom data and receiving the predicted diseases. Finally, the "Web Application" displays the predicted diseases to the "Patient"



**Figure: 2.5. Collaboration Diagram**

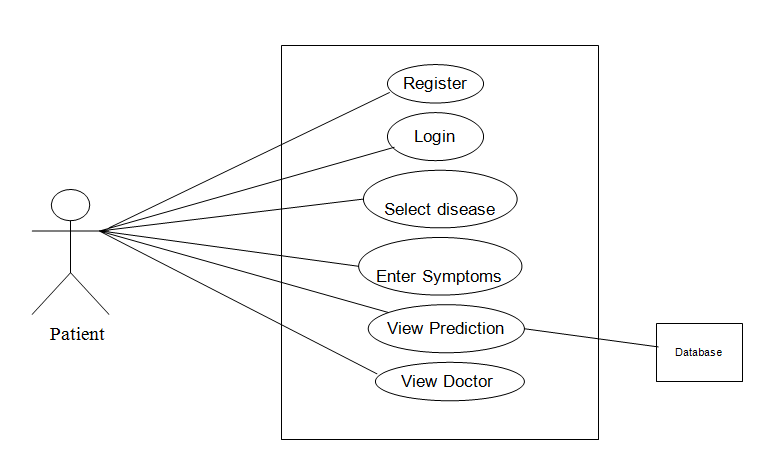
* 1. ***Admin Use case Diagram***





**Figure: 2.6. Admin use case diagram**

* 1. ***Patient Use case Diagram***



**Figure: 2.7. Patient use case diagram**

* 1. ***Use Case Registration Description***

**Description:** This use case describes the process of a user registering an account on the Expert prediction system.

**Actor:** User

**Precondition:**  None

**Postcondition:** User account is created and user is logged in

**Main Flow**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. User opens the registration page. |  |
| 1. User enters their personal details, such as name, email, and password. | 1. System validates the user input. |
| 1. User submits the registration form. | 1. System creates a new user account and assigns a unique user ID. |
|  | 1. System logs the user in and redirects them to the home page. |
| **Use Case End** | | | |

**Table: 2.1. Use Case Registration Description**

* 1. ***Use Case Login Description***

**Description:** This use case describes the process of a user login an account on the Expert prediction system.

**Actor:** User

**Precondition:**  None

**Postcondition:** User is logged in

**Main Flow**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. User opens the login page. |  |
| 1. User enters their personal details, such as name, email, and password. | 1. System validates the user input. |
| 1. User submits the login form. | 1. System creates a new user account and assigns a unique user ID. |
|  | 1. System logs the user in and redirects them to the home page. |
| **Use Case End** | | | |

**Table: 2.2. Use Case Login Description**

* 1. ***Use Case Heart Disease Prediction***

**Description:** This use case involves the core functionality of the system, where a user inputs their symptoms or and medical history to get predictions for possible diseases.

**Actor:** User

**Precondition:** User is logged in

**Postcondition:** User receives a list of predicted diseases

**Main Flow**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. User navigates to the disease prediction section. |  |
| 1. User provides their symptoms and medical history through the input form. | 1. System analyzes the user's input. |
| 1. User submits the form | 1. System applies the trained models to predict potential diseases. |
|  | 1. System displays results to the user. |
| **Use Case End** | | | |

**Table: 2.3. Use Case Heart Disease Prediction**

* 1. ***Use Case Diabetes Disease Prediction***

**Description:** This use case involves the core functionality of the system, where a user inputs their symptoms or and medical history to get predictions for possible diseases.

**Actor:** User

**Precondition:** User is logged in

**Postcondition:** User receives a list of predicted diseases

**Main Flow**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. User navigates to the disease prediction section. |  |
| 1. User provides their symptoms and medical history through the input form. | 1. System analyzes the user's input. |
| 1. User submits the form | 1. System applies the trained models to predict potential diseases. |
|  | 1. System generates whether a patient diabetic or not. |
|  | 1. System displays results to the user. |
| **Use Case End** | | | |

**Table: 2.4. Use Case Diabetes Disease Prediction**

* 1. ***Use Case View Doctor Description***

**Description:** This use case involves the functionality of the system, where a user view Doctor List

**Actor:** User

**Precondition:** User is logged in

**Postcondition:**  View doctor list

**Main Flow**

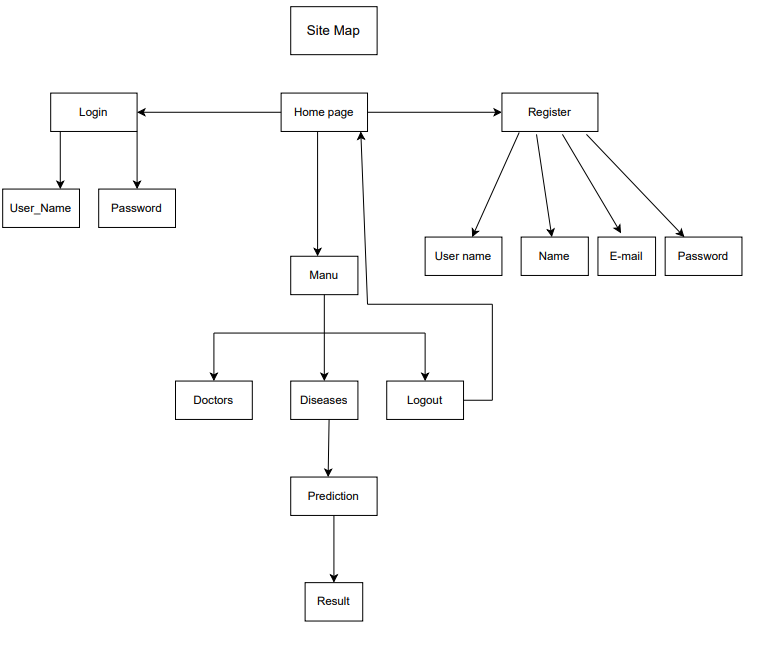
|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. User navigates to the menus |  |
| 1. User view doctor list |  |
|  |  |
|  |  |
| **Use Case End** | | | |

**Table: 2.5. Use Case View Doctor Description**

**Chapter 3: User Interface Design**

* 1. ***Site Map***

.

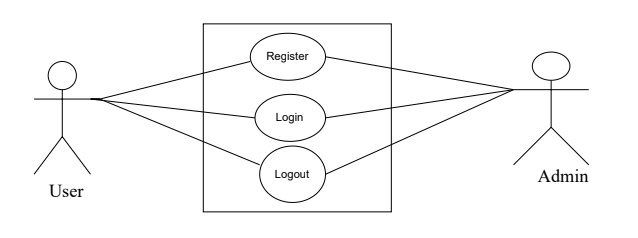


**Figure: 3.1. Site Map**

* 1. ***Story boards***

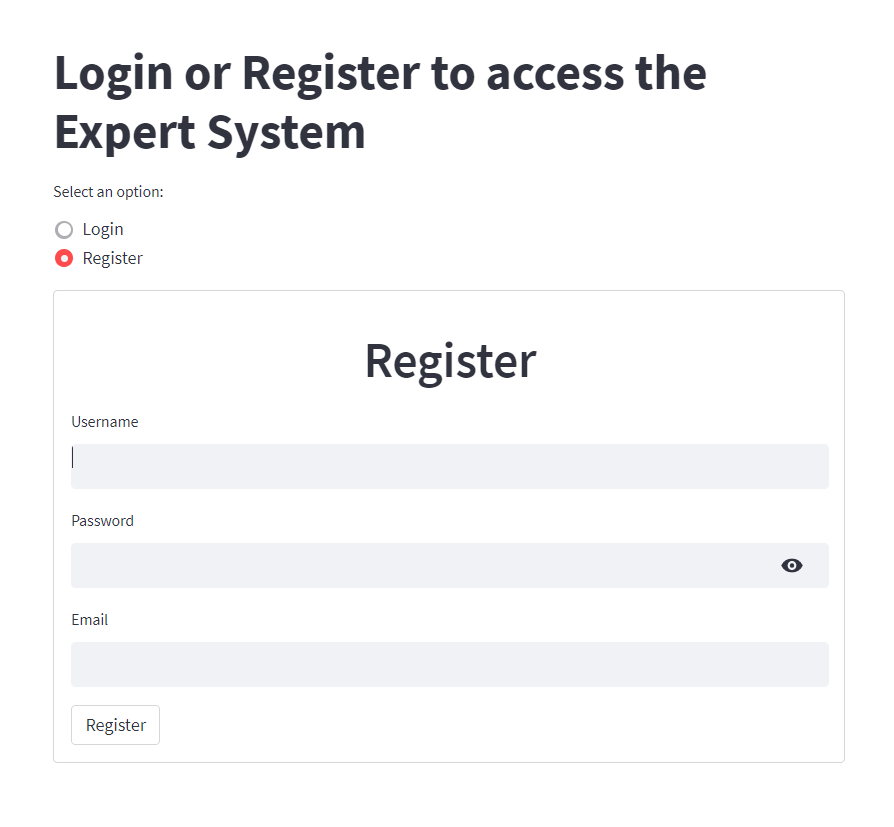
**What user see:**

* **Home page**
* **Login**
* **Registration**
* **Diseases**
* **Doctors**



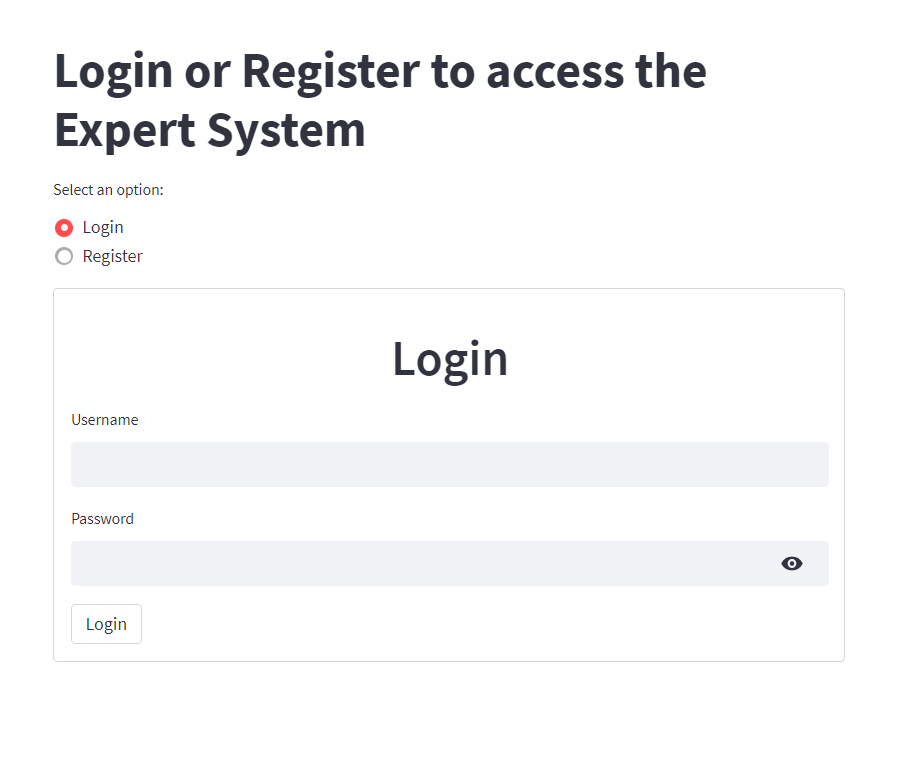
**Figure: 3.2. Story Board**

***Registration***



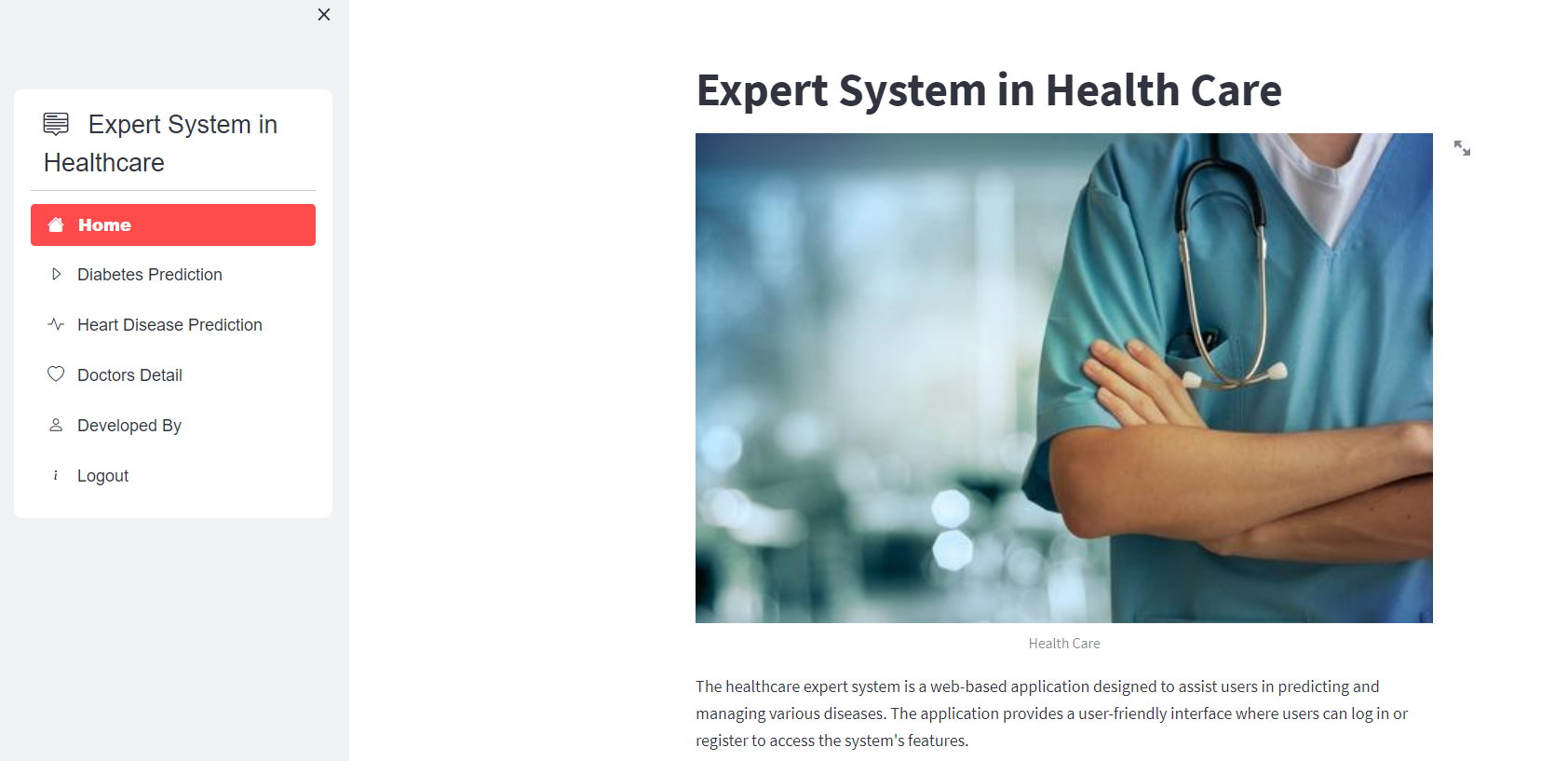
**Figure: 3.1. Registration**

***Login***



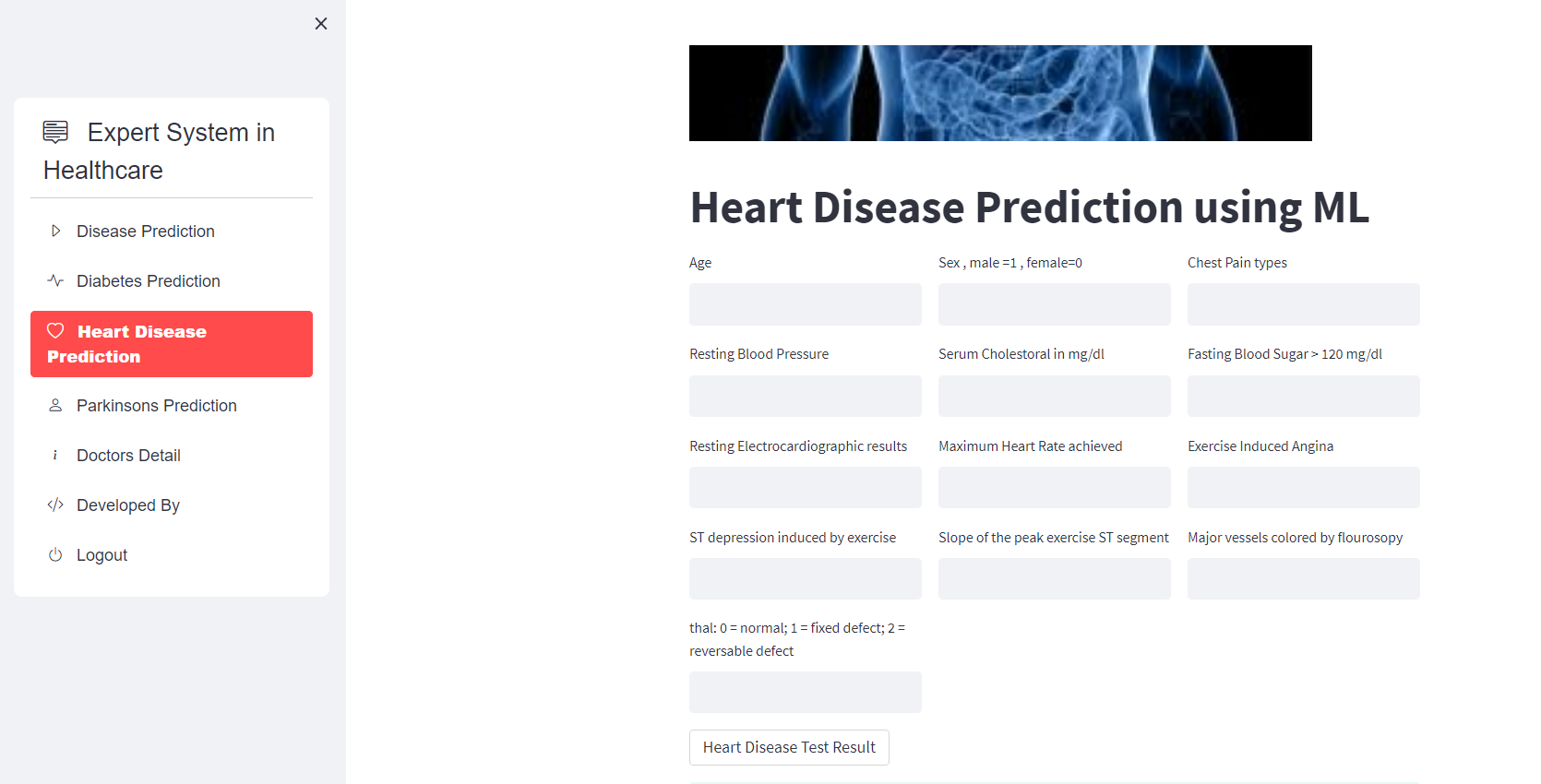
**Figure: 3.2. Login**

***Home Page***



**Figure: 3.3. Home Page**

* ***Heart Disease***



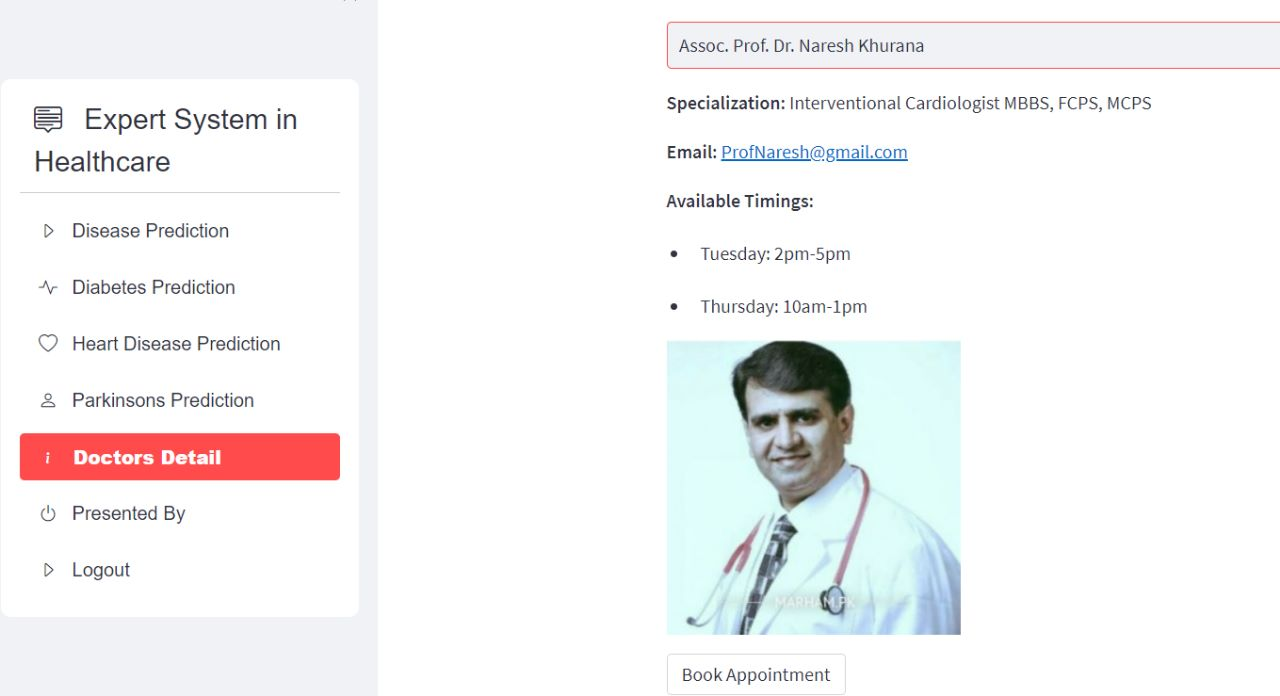
**Figure: 3.4. Heart Disease**

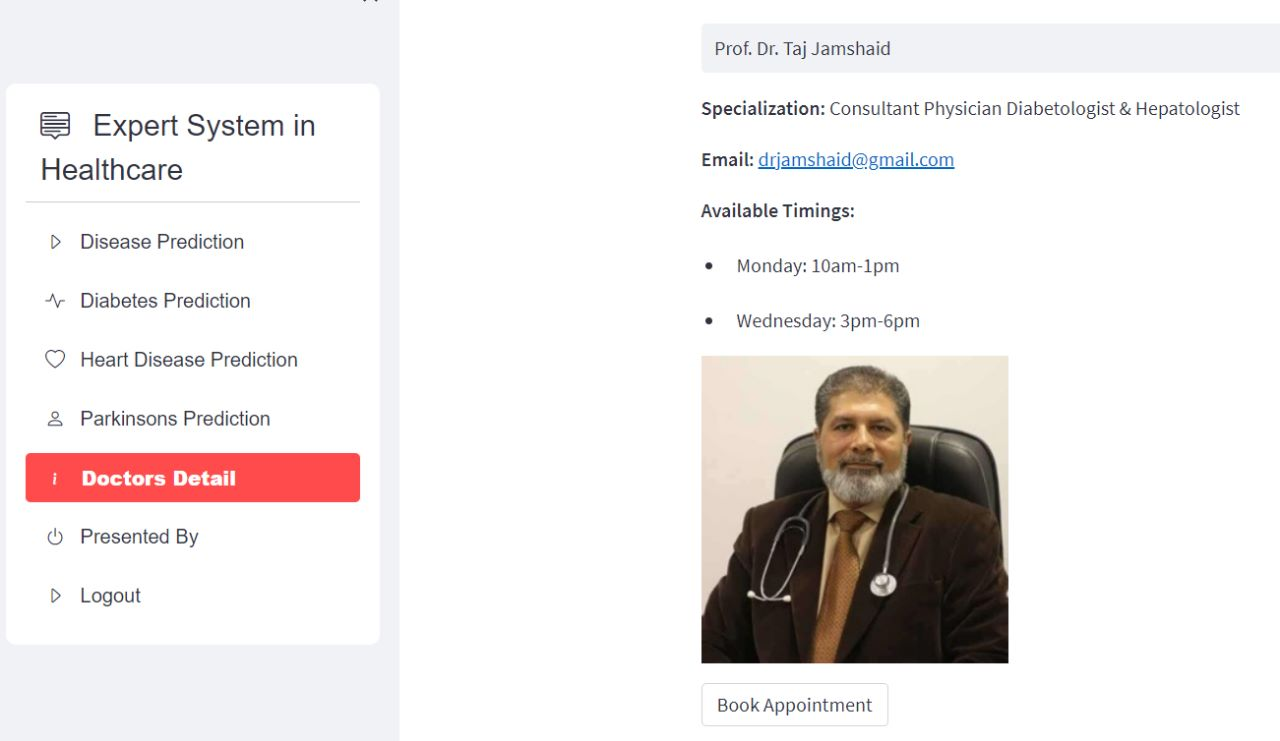
* ***Diabetes Disease***



**Figure: 3.4. Diabetes Disease**

* ***Doctors***





**Figure: 3.5. Doctor**

* 1. ***Prototype***

**Chapter 4: Software Testing**

* 1. **Introduction:**

Software testing is a crucial aspect of the Multiple Disease Prediction System project. It ensures the system's quality, accuracy, and reliability. Testing includes functional, performance, and user acceptance testing to verify that the system meets requirements, performs well, and satisfies user expectations. Defects are identified, logged, and resolved to ensure a reliable and user-friendly application.

* 1. ***Black box plan/White box plan/Grey box plan***
     1. **Black Box Testing**

This approach involves testing the functionality of the multiple disease prediction system without considering its internal implementation. Testers will focus on the inputs and outputs of the system, ensuring that it behaves as expected based on the given specifications. They will validate the accuracy of disease predictions, user input handling, and overall system performance without delving into the internal code.

* + 1. **White Box Testing**

White-box testing for your project would involve examining the internal structure and code of the multiple disease prediction system. Testers will design test cases based on their understanding of the code logic and perform detailed code-level testing. This approach will help identify any coding errors, boundary cases, or logical flaws that could affect the accuracy or performance of the disease prediction algorithm.

* + 1. **Grey Box Testing**

Gray-box testing combines aspects of black-box and white-box testing. Testers will have limited knowledge of the internal workings of the system, allowing them to design test cases that target specific areas of interest. They will leverage their understanding of the code and system behavior to uncover potential issues related to data handling, algorithmic calculations, or user interactions.

* 1. **White Box Approach**

In the context of a multi-disease prediction system, applying a white box approach means that we have access to the internal structure and code of the system, allowing you to conduct testing based on that knowledge. Here's how the white box approach can be applied to our project:

**Code Analysis**: Review the codebase of your multi-disease prediction system. Understand the algorithms, data structures, and logic used in the system. This analysis will help you gain insights into how the system processes input data and makes predictions.

**Unit Testing:** Write unit tests to verify the correctness of individual components or functions within the system. This involves creating test cases that exercise different paths and scenarios to ensure that each component behaves as expected.

**Code Coverage:** Measure the code coverage of your tests to ensure that a significant portion of the codebase is exercised. This helps identify any areas of the code that are not adequately tested and ensures comprehensive testing of the system.

**Integration Testing:** Test the interaction and integration between different modules or components of the system. This ensures that the system as a whole function correctly and that the integration points are working as intended.

**Boundary Testing:** Test the system with inputs at the upper and lower limits of their valid ranges. This includes testing extreme values, edge cases, and invalid inputs to verify that the system handles them properly.

**Performance Testing:** Assess the performance and scalability of the system under different loads and stress levels. This involves measuring response times, resource utilization, and the system's ability to handle concurrent requests.

**Security Testing:** Evaluate the system's security measures to identify potential vulnerabilities and ensure that sensitive data is properly protected. This may involve testing authentication mechanisms, data encryption, and protection against common security threats.

By using a white box approach, you can gain a deeper understanding of how your multi-disease prediction system works and ensure that it functions correctly and reliably. It allows you to thoroughly test the internal components, identify and fix any issues early on, and improve the overall quality and reliability of your system.

* 1. ***Test plan***

1. login
2. login user
3. register user
4. logout
5. selecting disease
6. prediction diabetes
7. prediction heart disease
8. view doctor
   1. ***Test log***

**User Registration Test Case: TC1**

**Test Engineer:** Arslan

**DATE:** 4-4-2023

**PURPOSE:** Registration

**Test data:** name:

Email:

Pass:

**Done**

Steps

1. Visit Registration page
2. Enter name
3. Enter password
4. Enter register
   1. ***Login***

**User login Test Case: TC2**

**Test Engineer:** Umer Shabbir

**DATE:** 5-4-2023

**PURPOSE:** login

**Test data:** name:

Pass

**done**

Steps

1. Visit login page
2. Enter name
3. Enter password
4. Enter login
   1. ***Logout***

**User login Test Case: TC3**

**Test Engineer:** Umer Shabbir

**DATE:** 5-4-2023

**PURPOSE:** logout

**Test data:**

**done**

* 1. ***Diabetes***

**Diabetes disease Test Case: TC4**

**Test Engineer:** Muhammad Saqib

**DATE:** 6-4-2023

**PURPOSE:** disease view

**Test data:** input required data

Steps

1. Visit home page
2. Select disease
3. Enter values
4. done
   1. ***Heart disease***

**Heart disease Test Case: TC5**

**Test Engineer:** Muhammad Arslan

**DATE:** 7-4-2023

**PURPOSE:** disease view

**Test data:** input required data

Steps

1. Visit home page
2. Select disease
3. Enter values
4. done
   1. ***doctor view***

**Test Case: TC6**

**Test Engineer:** Umer Shabbir

**DATE:** 9-4-2023

**PURPOSE:** doctor view

**Test data:** select

Steps

1. Visit home page
2. Select doctor menu
3. See all relevant doctor
4. done
   1. ***Heart Disease***

**Test Case: TC7**

**Test Engineer:** Umer Shabbir

**DATE:** 9-4-2023

**PURPOSE:** Heart prediction view

**Test data:** select

Steps

1. Visit home page
2. Select Disease
3. Input Values
4. Result
   1. ***Diabetes Disease***

**Test Case: TC8**

**Test Engineer:** Umer Shabbir

**DATE:** 9-4-2023

**PURPOSE:** Diabetes prediction view

**Test data:** select

Pass

Steps

* 1. Visit home page
  2. Select doctor menu
  3. See all relevant doctor
  4. Result
  5. ***Disease***

**Test Case: TC9**

**Test Engineer:** Umer Shabbir

**DATE:** 9-4-2023

**PURPOSE:** prediction view

**Test data:** select

Pass

Steps

1. Visit home page
2. Select Disease
3. Input Values
4. Result

***REFERENCES***

Certainly! Here are some IEEE research papers related to disease prediction and healthcare that you may find helpful for your project:

1. "A Survey on Machine Learning Techniques for Disease Diagnosis" by P. Deepika, S. Sudha, and V. Nandhini.

- This paper provides a comprehensive survey of machine learning techniques used for disease diagnosis, including decision trees, support vector machines, and neural networks.

2. "A Machine Learning Approach for Heart Disease Prediction" by B. S. Desai and M. L. Nirmala.

- Focusing on heart disease prediction, this paper explores the application of machine learning algorithms such as k-nearest neighbors, support vector machines, and random forests.

3. "Prediction of Diabetes Mellitus Using Machine Learning Techniques" by G. S. Raju and K. Srinivas.

- This paper discusses the use of machine learning techniques for predicting diabetes mellitus, including logistic regression, decision trees, and naive Bayes classifiers.

4. "A Review of Machine Learning Techniques for Cancer Prediction and Diagnosis" by R. Krishna and A. K. Chaudhary.

- This review paper provides an overview of machine learning techniques used for cancer prediction and diagnosis, including feature selection methods, classification algorithms, and ensemble techniques.

5. "Machine Learning-Based Models for Early Prediction of Parkinson's Disease" by K. Srinivasan and M. Krishnan.

- Focusing on Parkinson's disease, this paper explores the application of machine learning models for early prediction, including random forests, support vector machines, and artificial neural networks.