**Final Year Project Report**

**Project Name**



**Project Advisor:**

**Submitted By:**

**Eiman Fatima(F2022266290)**

**Farheen Azhar (F2022266257)**

**Mariyam Anwar(F2022266262)**

**Abdul Wahab(F202226671)**

**Session**

**V4**

**University of Management and Technology**

**C-II Johar Town Lahore Pakistan**

**Github :  
[eimaanfatima208/RiskGueardAI](https://github.com/eimaanfatima208/RiskGueardAI)**

## Final Approval

* **Head of Department**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Department of Informatics & Systems

School of Systems & Technology

UMT Lahore

* **Director (Final Year Projects-CS)** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Department of Computer Science.

School of Systems & Technology

UMT Lahore

* **Supervisor** \_\_\_\_\_Ma`am Fasiha Ashraf\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Department of Informatics & Systems.
* School of Systems & Technology

UMT Lahore

* **Co-Supervisor** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Acknowledgment**

**Project Title**

**Objective**

**Undertaken by**

**Supervised by**

**Starting Date**

**Completion Date**

**Tools Used**

**Operating System**

**Documentation**

**Plagairism Report**

**Abstract**

The Health Risk Predictor (HRP) is a web application designed to empower individuals by predicting potential health risks and offering personalized wellness guidance. Leveraging technologies like HTML, CSS, JavaScript, PHP, and MySQL, HRP analyzes users’ symptoms and medical history to generate predictions for possible diseases, accompanied by workout routines and guidance by medical advisor . .

Users begin by creating secure accounts with encrypted password storage and optional medical history input. The intuitive dashboard allows dynamic symptom entry, which the system cross-references with a comprehensive database to predict the top five likely diseases, adjusting probabilities based on the user’s medical background. For known conditions, HRP directly recommends prescriptions and related workouts.

Administrators efficiently manage data through a dedicated interface, performing seamless CRUD (Create, Read, Update, Delete) operations across interconnected tables, ensuring data accuracy without technical expertise. Users personalize their experience through profile settings, including medical history updates, theme customization, and search history tracking for revisiting past analyses.

By bridging symptom analysis with AI-driven insights, HRP enhances preventive healthcare, enabling users to make informed decisions while simplifying backend management for administrators. This tool exemplifies how technology can transform personal health management into an accessible, proactive, and secure experience.

Revision Chart

This chart contains a history of this document’s revisions. The entries below are provided solely for illustration purposes. Those entries should be deleted until the revision/s they refer to have actually been created.

The document itself should be stored in revision control, and a brief description of each version should be entered in the Revision Control System. A brief description can be repeated in this section. Revisions need not be described elsewhere in the document, unless they explain the document.

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| *Draft* | TBD | Initial draft created for distribution and review comments | (To be decided) TBD |
| *Preliminary* | TBD | Second draft incorporating initial review comments, distributed for final review | TBD |
| *Final* | TBD | First complete draft, which is placed under change control | TBD |
| *Revision 1* | TBD | Revised draft, revised according to the change control process and maintained under change control | TBD |
| *Revision 2* | TBD | Revised draft, revised according to the change control process and maintained under change control | TBD |
| *Etc.* | TBD | TBD | TBD |

# Contents

*New paragraphs formatted as Heading 1, Heading 2, and Heading 3 will be added to the table automatically. To update this table of contents in Microsoft Word, put the cursor anywhere in the table and press F9. If you want the table to be easy to maintain, do not change it manually.*

[Contents 1](#_Toc470104823)

[Definitions and Acronyms 3](#_Toc470104824)

[List of Figures 4](#_Toc470104825)

[List of Tables 5](#_Toc470104826)

[1. Introduction 6](#_Toc470104827)

[1.1 Motivations 6](#_Toc470104828)

[1.2 Project Overview 6](#_Toc470104829)

[1.3 Problem Statement 6](#_Toc470104830)

[1.4 Objectives 6](#_Toc470104831)

[2. Domain Analysis 7](#_Toc470104855)

[2.1 Customer 7](#_Toc470104856)

[2.2 Stakeholders 7](#_Toc470104857)

[2.3 Affected Groups with social or economic impact 7](#_Toc470104858)

[2.4 Dependencies/ External Systems 7](#_Toc470104859)

[2.5 Reference Documents 7](#_Toc470104860)

[2.5.1 Related Projects 7](#_Toc470104861)

[2.5.2 Feature Comparison 8](#_Toc470104862)

[3. Requirements analysis 9](#_Toc470104865)

[3.1 Requirements 9](#_Toc470104866)

[3.2 List of Actors 10](#_Toc470104897)

[3.3 List of use cases 10](#_Toc470104898)

[3.4 System use case diagram 10](#_Toc470104899)

[3.5 Extended use cases 10](#_Toc470104900)

[3.6 User interfaces (mock screens) 12](#_Toc470104901)

[4. Data flow diagram (optional) 13](#_Toc470104902)

[4.1 Data Flow Diagram Level 0 13](#_Toc470104903)

[4.2 Data Flow Diagram Level 1 13](#_Toc470104904)

[4.3 Data Flow Diagram Level 2 13](#_Toc470104905)

[5. System Design 15](#_Toc470104908)

[5.1 System Architecture Diagram 15](#_Toc470104909)

[5.2 Class Diagram 16](#_Toc470104910)

[5.3 Collaboration Diagrams 17](#_Toc470104911)

[5.4 Other UMLs 18](#_Toc470104920)

[5.5 ERD 18](#_Toc470104921)

[5.6 Data Dictionary 18](#_Toc470104922)

[6. Implementation details 19](#_Toc470104936)

[6.1 Development Setup 19](#_Toc470104937)

[6.2 Deployment setup 19](#_Toc470104938)

[6.3 Algorithms 19](#_Toc470104939)

[6.4 Constraints 19](#_Toc470104940)

[6.4.1 Assumptions 19](#_Toc470104941)

[6.4.2 System constraints 19](#_Toc470104942)

[6.4.3 Restrictions 19](#_Toc470104943)

[6.4.4 Limitations 19](#_Toc470104944)

[7. Testing 20](#_Toc470104948)

[7.1 Extended Test Cases 20](#_Toc470104949)

[7.2 Decision Table 20](#_Toc470104950)

[7.2.1 Code snippet 20](#_Toc470104951)

[7.2.2 Decision coverage table 20](#_Toc470104952)

[7.3 Traceability Matrix 21](#_Toc470104953)

[7.3.1 RID vs UCID (requirements vs use cases) 21](#_Toc470104954)

[7.3.2 Prototypes (RID vs PID) 22](#_Toc470104955)

[7.3.3 Test Cases (RID vs TID) 22](#_Toc470104956)

[7.3.4 Coverage (UCID vs TID) 22](#_Toc470104957)

[8. Results/Output/Statistics 23](#_Toc470104958)

[8.1 %completion 23](#_Toc470104959)

[8.2 %accuracy 23](#_Toc470104960)

[8.3 %correctness 23](#_Toc470104961)

[9. Conclusion 24](#_Toc470104962)

[10. Future work 25](#_Toc470104963)

[11. Bibliography 26](#_Toc470104964)

[11.1 Books 26](#_Toc470104965)

[11.2 Journals 26](#_Toc470104966)

[11.3 Articles 26](#_Toc470104967)

[11.4 Research papers 26](#_Toc470104968)

[11.5 Other References 26](#_Toc470104969)

[12. Appendix 27](#_Toc470104970)

[12.1 Glossary of terms 27](#_Toc470104971)

[12.2 Pre-requisites 27](#_Toc470104975)

## Definitions and Acronyms:

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| **HRP** | Health Risk Predictor |
| **Medical history** | Stored records of a user’s past illnesses and conditions |
| Symptoms Database | Collection of all possible symptoms used for disease prediction |
| Prescription Database | Collection of medical prescriptions related to diseases |
| Workout Database | Collection of recommended workouts based on medical conditions |
| **API** | Application Programming Interface |
| **DBMS** | Database Management System |
| **HTML** | HyperText Markup Language |
| **CSS** | Cascading Style Sheets, used for styling web pages |
| **PHP** | Hypertext Preprocessor, used for data connectivity |
| **CRUD** | Create, Read, Update, Delete |

## List of Figures

New figures that are given captions will be added to the table automatically.

* **Insert caption:**
  1. select picture
  2. right click
  3. select “insert caption”
  4. under “options”, choose label as “figure”
  5. Under “caption”, an automatic insertion of “figure no” will appear. Give your figure an appropriate caption
* **Update list of figures:**To update this list in Microsoft Word, put the cursor anywhere in the table and press F9.
* **Note:**  If you want the table to be easy to maintain, do not change it manually.

Figure 1: sample use case diagram with explanation 10

Figure 2: System Architecture 15

## List of Tables

New Tables that are given captions will be added to the table automatically.

* **Insert caption:**
  1. select picture
  2. right click
  3. select “insert caption”
  4. under “options”, choose label as “table”
  5. Under “caption”, an automatic insertion of “table no” will appear. Give your table an appropriate caption
* **Update table:** To update this table of contents in Microsoft Word, put the cursor anywhere in the table and press F9.
* **Note:**  If you want the table to be easy to maintain, do not change it manually.

[Table 1: table of acronyms and definitions 3](#_Toc470104183)

[Table 2: list of stakeholders 7](#_Toc470104184)

# Introduction

## Motivations

The increasing prevalence of health issues and the need for early disease prediction have driven the development of the Health Risk Predictor (HRP). Many individuals ignore early symptoms due to a lack of awareness or medical access. This project aims to assist users in predicting potential diseases based on symptoms and medical history, offering workout suggestions and medical prescriptions for prevention and treatment

## Project Overview

The HRP is a web-based application that predicts possible diseases based on user-input symptoms and medical history. It also suggests workouts and provides medical prescriptions. The system enables users to create accounts, manage their medical history, and retrieve personalized health-related insights. An admin panel facilitates user and database management.

**Sample Artifacts:**

**Problem Overview Statement:** Difficulty in early disease prediction and lack of accessible personalized health recommendations.

**Customer:** General public, fitness enthusiasts, and individuals with a medical history.

**Goals:** Provide an intuitive interface for symptom analysis, accurate disease prediction, and tailored prescriptions/workouts.

**System Functions:** User authentication, symptom input, disease prediction, prescription and workout recommendation, user profile management.

## Problem Statement

Many people struggle to identify potential health risks at an early stage due to lack of medical expertise or access to immediate healthcare. The HRP aims to bridge this gap by providing a web-based solution where users can input their symptoms and receive predictions for potential diseases, along with medical prescriptions and workout recommendations. This system ensures that users can take preventive actions before their conditions worsen.

## Objectives

These are the objective of our Project:

* Develop an interactive system to predict diseases based on user-input symptoms.
* Provide personalized workout and prescription recommendations.
* Implement a secure login and account management system.
* Store and retrieve medical history for better analysis and predictions.
* Offer an admin panel for user and database management

# Domain Analysis

## Customer

The customer for our project are the general public, specifically individuals who are conscious of their health and want to take proactive steps in managing it. These users may include those experiencing symptoms, individuals with pre-existing medical conditions, or health-conscious people looking to predict possible health risks based on their medical history and symptoms. The application will cater to users who want personalized disease predictions, medical prescriptions, and rehabilitation workout recommendations.

## Stakeholders

### 2.2 Stakeholders

| **Stakeholder** | **Role in System** |
| --- | --- |
| Users (Patients) | Users who interact with the system to predict there diseases, get medical prescriptions, and follow workout routines . |
| Moderator | Manages the database, performs CRUD operations on user data |
| Developers | Responsible for the development, implementation, and maintenance of the application |
| Database Administrator | Manages the database, ensuring that user data, medical history, symptoms, diseases, prescriptions, and workouts are stored securely and efficiently. |
| Medical  Experts (Advisors) | Provide expert knowledge to ensure the accuracy of disease predictions, prescriptions, and workout routines generated by the system. |
| UI/UX Designers | Design the user interface and ensures that application is user-friendly, accessible, and meets the needs of all user types. |
| Quality Assurance Team | Tests the system for bugs, ensures reliability, and confirms that it meets the specified requirements before release. |
| Client (Owner) | Oversees the project, provides feedback and requirements, and ensures the system aligns with the goals of the health risk prediction and management application. |
| **Doctors/Medical Advisors** | Help make sure the disease predictions, prescriptions, and workouts provided by the app are accurate and medically correct. |

## Affected Groups with social or economic impact

* End Users (Patients)  
  Users will have access to disease predictions, prescriptions, and personalized workouts, improving health management
* Doctors/Medical Advisors  
  Doctors can use the system for faster and more accurate diagnoses, improving patient care efficiency and reducing consultation time.
* Project Team (Developers)  
  Developers gain valuable real-world experience in healthcare application development, enhancing their careers.
* Healthcare System (Global Impact)  
  The app can reduce global healthcare burdens by promoting early health intervention and preventive care, potentially lowering healthcare costs globally.

## Dependencies/ External Systems

* Doctors/Medical Advisor
* Healthcare Professionals (For Validation)
* External Medical Databases (Optional)
* Rehabilitation specialist

## Reference Documents

* M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang,“Disease prediction by machine learning over big data from healthcare communities” IEEE Access, vol. 5, no.1, pp.8869–8879, 2017.
* Sayali Ambekar, Rashmi Phalnikar, “Disease RiskPrediction by Using Convolutional Neural Network” IEEE, 978-1-5386-5257-2/18, 2018.
* Dhiraj Dahiwade, Gajanan Patle and Ektaa Meshram, “Designing Disease Prediction Model Using Machine Learning Approach” IEEE Xplore Part Number: CFP19K25-ART; ISBN: 978-1-5386-7808-4, pp. 1211-1215, 2019.
* Pahulpreet Singh Kohli and Shriya Arora, “Application of Machine Learning in Disease Prediction” IEEE, 978-1-5386-6947-1/18, pp. 1-4, 2018.

### Related Projects

* [Maternal-Health-Risk-Predictor](https://github.com/Vikhram-S/Maternal-Health-Risk-Predictor)
* [Multiple-Disease-Prediction-System-using-Machine-Learning](https://github.com/Amit380/Multiple-Disease-Prediction-System-using-Machine-Learning)
* [Heart\_disease\_prediction](https://github.com/chayandatta/Heart_disease_prediction)
* [Disease-Prediction-from-Symptoms](https://github.com/anujdutt9/Disease-Prediction-from-Symptoms)

### Feature Comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr No.** | **Comparison Feature** | **FMS** | **BHMS** | **CIAS** | **remarks** |
| **1** | **Validate User Credentials** | System checks if the username and the password are correct. | BHMS does not support feature ABC | CIAS suggests implementing multi-factor authentication for enhanced security. | Using the ABC feature from FMS and improving it with abc algorithm can provide maximum efficiency |
| **2** | **Predict Disease** | System analyzes symptoms and medical history and provides top 5 most accurate disease predictions with probability percentages. | BHMS has limited prediction capabilities | CIAS recommends integrating AI-based prediction models for improved accuracy.. | Using the ABC feature from FMS and improving it with abc algorithm can provide maximum efficiency |
| **3** | **Manage User Profile** | Allows users to update username, password, and profile picture with encryption and validation . | Limited profile management option | suggests implementing stronger encryption and biometric authentication for enhanced security. | Using the ABC feature from FMS and improving it with abc algorithm can provide maximum efficiency |
| **4** | **View User Record** | Enables users to input past diseases, allergies, and other medical details for future analysis. | Lacks structured data storage for medical history | CIAS recommends using a standardized format for storing medical data to improve consistency and predictive accuracy. | Using the ABC feature from FMS and improving it with abc algorithm can provide maximum efficiency |
| **5** | **Receive Workout Recommendation** | Provides personalized workout routines based on diagnosed diseases and prescribed medicine. | Lacks integration with disease prediction | CIAS suggests integrating AI-based fitness recommendations based on user health data for better personalization. | Using the ABC feature from FMS and improving it with abc algorithm can provide maximum efficiency |

# Requirements analysis

## Requirements

**3.1 Requirements**

The Health Risk Predictor (HRP) system has several requirements categorized into functional and non-functional aspects. Below is a detailed list:

### Functional Requirements:

| **No.** | **Requirement** | **Description** |
| --- | --- | --- |
| **1** | **User Authentication** | The system should allow users to register and log in securely. |
| **2** | **Symptom Input** | Users must be able to enter multiple symptoms for health risk analysis. |
| **3** | **Disease Prediction** | The system should analyze symptoms and provide a ranked list of possible diseases. |
| **4** | **Medical History Management** | Users should be able to add, update, or remove medical history records. |
| **5** | **Prescription Recommendations** | Based on the predicted diseases, the system should suggest relevant prescriptions. |
| **6** | **Workout Suggestions** | The system should provide workout routines based on medical conditions. |
| **7** | **User Profile Management** | Users should be able to update personal details and security credentials. |
| **8** | **Admin Panel** | Admins must be able to manage users and maintain system data integrity. |
| **9** | **Search History Tracking** | The system should maintain a log of user searches for future reference. |
| **10** | **User Interface Customization** | Users should be able to switch between light and dark modes for better accessibility. |
| **11** | ***Data Requirements*** | *User data such as personal Information, symptoms, medical history, and prediction records must be store in a database* |
| **12** | ***External-Interface Requirements*** | *The system must be integrate with third-party AI/ML services for disease prediction* |

### Non-Functional Requirements:

| **No.** | **Requirement** | **Description** |
| --- | --- | --- |
| **1** | **Performance** | The system should provide search results within 5 seconds. |
| **2** | **Security** | User passwords must be encrypted before storage. |
| **3** | **Scalability** | The application should handle multiple concurrent users efficiently. |
| **4** | **Availability** | The system should be accessible 24/7 with minimal downtime. |
| **5** | **Usability** | The interface should be intuitive and easy to navigate for users of all ages. |
| **6** | **Data Storage** | All user and system data should be stored in a structured database with proper indexing. |

## List of Actors

* User
* System
* Database
* Admin
* API

## List of use cases

1. **Validate User Credentials**

* System checks if the username and password are correct

**2. Manage User Profile**

* User can update username, password, and profile picture.
* Password strength validation and encryption are applied.
* User can update,view and input their symptoms.

**3. View User Record**

* User can input past diseases, allergies, and other medical details.
* System stores the information for future predictions.

**4. Predict Disease**

* System analyzes symptoms and medical history.
* Provides the top 5 most accurate disease predictions with probability percentages.

**5. Receive Workout Recommendation**

* System provides workout routines related to the diagnosed disease and prescribed medicine.

**6.Admin Database**: Manages system administration data, user roles, permissions, and tracks activity logs for system security.

1. **Symptoms Database**: Stores user-reported symptoms and analyzes them for disease prediction and health trend insights.
2. **Medical History Database**: Stores users' past medical diagnoses, treatments, and health conditions for personalized care.
3. **API Health Insights** – The system identifies health trends and provides proactive health suggestions, analyzing user data stored in the database to deliver personalized advice.

## System use case diagram

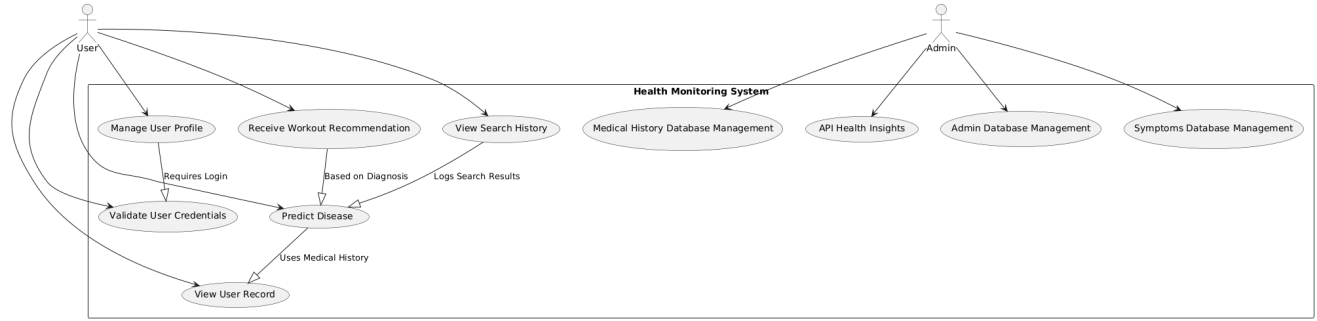


Figure 1: sample use case diagram with explanation

## Extended use cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-1.1 | | |
| **Use Case Name:** | Validate User Credentials | | |
| **Created By:** |  | Last Updated By: |  |
| **Date Created:** |  | Last Revision Date: |  |
| **Actors:** | Primary: User Secondary: System | | |
| **Description:** | System verifies the accuracy of the user’s credentials during the login process. If the username and password match the stored information, access is granted. | | |
| **Trigger:** | User attempts to log in. | | |
| **Preconditions:** | User must have a registered account.  The system must have stored credentials for comparison. | | |
| **Post conditions:** | 1. If credentials are valid, the user is granted access. 2. If credentials are invalid, the system denies access and prompts for re-entry | | |
| **Normal Flow:** | 1. User enters username and password. 2. System checks credentials against stored records. 3. If credentials are correct, user is logged in. 4. If credentials are incorrect, an error message is displayed. | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | **Alternative Flows:** Alternative Flow 1 - Incorrect Credentials  If the credentials are incorrect:  1.System displays an error message.  2.User is prompted to re-enter credentials.  3.Use case resumes at Step 1. | | |
| **Exceptions:** | **Exceptions:** Exception 1 - Account Locked If a user enters incorrect credentials more than 5 times:  1.System locks the account.  2.User must reset the password to regain access. | | |
| **Includes:** | NONE | | |
| **Frequency of Use:** | On demand; every time a user attempts to log in. | | |
| **Special Requirements:** | 1.System should handle multiple failed login attempts securely.  2.Passwords must be encrypted. | | |
| **Assumptions:** | Users remember their credentials.  Users have a stable internet connection for authentication. | | |
| **Notes and Issues:** | Should the system implement biometric authentication in the future? | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-1.2 | | |
| **Use Case Name:** | Manage User Profile | | |
| **Created By:** |  | **Last Updated By:** |  |
| **Date Created:** |  | **Last Revision Date:** |  |
| **Actors:** | Primary: User Secondary: System | | |
| **Description:** | User can update their profile details, including username, password, and profile picture. Password strength validation and encryption are applied. Users can also update, view, and input their symptoms. | | |
| **Trigger:** | User accesses the profile management section. | | |
| **Preconditions:** | User must be logged in.  System must have stored user information. | | |
| **Post conditions:** | * Updated profile information is saved successfully. * If password is changed, the new password is encrypted. | | |
| **Normal Flow:** | 1. User accesses profile settings. 2. User updates profile details (username, password, profile picture). 3. System validates inputs (e.g., password strength). 4. System saves the updated information. 5. System encrypts the password (if updated). 6. User receives confirmation of the successful update. | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | Alternative Flows: Alternative Flow 1 - Weak Password 3a. If the user enters a weak password:  System prompts the user to choose a stronger password.  User enters a stronger password.  Use case resumes at Step 4. | | |
| **Exceptions:** | 1. Exception 1 - System Error If an error occurs while updating profile details: 2. System displays an error message. 3. User is prompted to retry. | | |
| **Includes:** | None | | |
| **Frequency of Use:** | On demand; whenever a user wishes to update their profile. | | |
| **Special Requirements:** | System should provide strong password recommendations.  Profile pictures should adhere to size and format constraints. | | |
| **Assumptions:** | Users enter valid information. | | |
| **Notes and Issues:** | Should users receive email confirmation when updating profile details? | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-1.3 | | |
| **Use Case Name:** | View User Record | | |
| **Created By:** |  | **Last Updated By:** |  |
| **Date Created:** |  | **Last Revision Date:** |  |
| **Actors:** | Primary: User  Secondary: System | | |
| **Description:** | User can input past diseases, allergies, and other medical details. The system stores this information for future predictions and personalized healthcare insights | | |
| **Trigger:** | 1. User accesses the medical records section. | | |
| **Preconditions:** | 1. User must be logged in.   2. System must have a database to store user medical records. | | |
| **Post conditions:** | 1. User’s medical history is successfully stored.  2. The system updates records for future reference in disease prediction. | | |
| **Normal Flow:** | 1. User navigates to the medical records section. 2. User selects the option to add or update medical history. 3. User inputs past diseases, allergies, and other medical details. 3. System validates the input. 4. System stores the medical details in the database.   6. User receives confirmation of successful record update. | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | **Alternative Flow 1 - Incomplete Information**  If the user provides incomplete medical details:   1. System prompts the user to complete the required fields. 2. User provides the missing details.   3. Use case resumes at Step 5. | | |
| **Exceptions:** | 1. **Exception 1 - System Error** If an error occurs while saving the record: 1. System displays an error message. 2. User is prompted to retry. | | |
| **Includes:** | None. | | |
| **Frequency of Use:** | On demand; whenever a user wishes to update their medical history | | |
| **Special Requirements:** | 1. The system should ensure data privacy and encryption.   2. Medical records should be accessible for future analysis. | | |
| **Assumptions:** | 1. Users provide accurate and complete information. | | |
| **Notes and Issues:** | 1. Should the system allow users to upload scanned medical documents?   2. Should medical history updates require approval from a healthcare provider? | | |
| **Use Case ID:** | UC-1.4 | | |
| **Use Case Name:** | Predict Disease | | |
| **Created By:** |  | **Last Updated By:** |  |
| **Date Created:** |  | **Last Revision Date:** |  |
| **Actors:** | Primary: User  Secondary: System | | |
| **Description:** | The system analyzes user-input symptoms and medical history to generate the top 5 most accurate disease predictions, along with probability percentages. | | |
| **Trigger:** | User enters symptoms and requests a diagnosis. | | |
| **Preconditions:** | 1.User must be logged in.  2. User must have an existing medical history stored (optional but improves accuracy).  3. System must have access to a disease prediction model | | |
| **Post conditions:** | 1.System provides the top 5 predicted diseases with probability percentages.  2. User receives personalized health insights. | | |
| **Normal Flow:** | 1.User navigates to the **Disease Prediction** section.  2.User enters symptoms.  3.System retrieves the user's medical history (if available).  4.System processes the data using an AI-based disease prediction model.  5.System generates a list of the **top 5 possible diseases**, ranked by probability.  6.System displays the results with probability percentages.  7. System provides additional recommendations (workout) | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | **Alternative Flow 1 - Insufficient Data**  If the user provides very few symptoms or conflicting inputs:  1.System prompts the user to enter more details.  2.User updates the symptoms list.  3.Use case resumes at Step 5. | | |
| **Exceptions:** | **Exception 1 - System Error**  If an error occurs while predicting the disease:   1. System displays an error message.   2. User is prompted to retry. | | |
| **Includes:** | **Use Case UC-1.3 (View User Record)** - System retrieves the user's medical history. | | |
| **Frequency of Use:** | On demand; whenever a user wants to check symptoms. | | |
| **Special Requirements:** | 1. The system should ensure high accuracy in predictions. 2. Data privacy and encryption should be enforced.   3. The AI model should be regularly updated with new medical data | | |
| **Assumptions:** | 1. Users provide accurate symptom descriptions.   2. System has sufficient medical data for prediction. | | |
| **Notes and Issues:** | 1. Should users receive alerts if their symptoms match high-risk diseases? | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-1.5 | | |
| **Use Case Name:** | Receive Workout Recommendation | | |
| **Created By:** |  | **Last Updated By:** |  |
| **Date Created:** |  | **Last Revision Date:** |  |
| **Actors:** | **Primary:** User **Secondary:** System | | |
| **Description:** | The system provides workout routines based on the diagnosed disease and prescribed diet to help improve the user's health condition | | |
| **Trigger:** | A disease diagnosis is generated, or the user requests a workout recommendation. | | |
| **Preconditions:** | 1. User must be logged in. 2. System must have access to the user's medical history and prescribed diet.   3. System must have a database of workout routines mapped to medical conditions. | | |
| **Post conditions:** | 1. User receives a list of **personalized workout routines** suitable for their condition. | | |
| **Normal Flow:** | 1. User navigates to the **Workout Recommendation** section.   2. System retrieves the user's diagnosed disease and prescribed diet.  3.System matches the user's medical data with **suitable workout routines**  4. System generates a personalized workout plan.  5. System displays the recommended exercises, including duration, intensity, and precautions.  6. User reviews the workout plan and selects a routine to follow.  7. System logs the user's workout selection for tracking purposes. | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | **Alternative Flow 1 - No Existing Diagnosis**  If the user has not been diagnosed with a disease or prescribed diet:  1. System prompts the user to enter health goals (e.g., weight loss, muscle gain) | | |
| **Exceptions:** | **Exception 1 - System Error**  If an error occurs while generating recommendations:  1) System displays an error message.  2) User is prompted to retry. | | |
| **Includes:** | **Use Case:**  **UC-2.1 (Predict Disease)** - System retrieves diagnosed diseases.  **Use Case UC-1.3 (View User Record)** - System retrieves the user's medical history. | | |
| **Frequency of Use:** | On demand; whenever a user wants. | | |
| **Special Requirements:** | 1. The system should recommend **only medically safe workouts** based on the diagnosed condition. 2. Workouts should include intensity levels, precautions, and alternatives.   3. Data privacy and security should be enforced. | | |
| **Assumptions:** | 1. Users follow the recommended workouts responsibly.   2. The system has an updated database of medical workout recommendations. | | |
| **Notes and Issues:** | 1. Should users be able to request professional trainer consultations? | | |

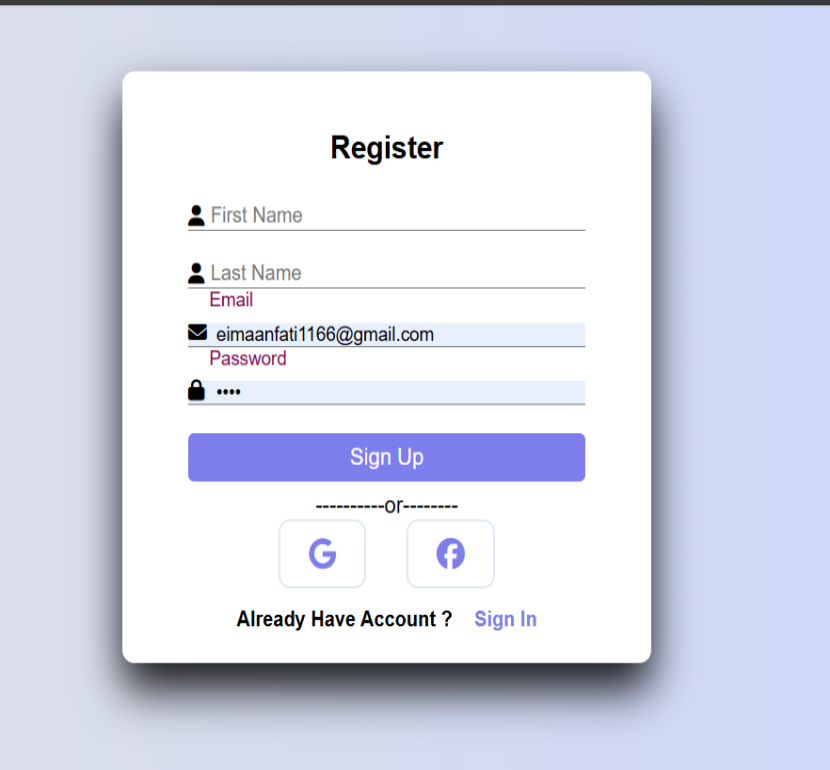
|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-1.6 | | |
| **Use Case Name:** | Admin Database Management | | |
| **Created By:** |  | **Last Updated By:** |  |
| **Date Created:** |  | **Last Revision Date:** |  |
| **Actors:** | **Primary:** Admin  **Secondary:** System | | |
| **Description:** | The admin manages system administration data. | | |
| **Trigger:** | The admin accesses the **Admin Database Management** section | | |
| **Preconditions:** | 1. Admin must be logged in with the necessary privileges.   2. System must have an active database of users, roles, and logs. | | |
| **Post conditions:** | 1. Admin successfully manages user roles, permissions, or views system logs.   2. Changes made are securely recorded in the system. | | |
| **Normal Flow:** | 1. Admin navigates to **Admin Dashboard**. 2. Admin selects an option and performs necessary actions. 3. System updates the database accordingly.   4.Admin confirms changes, and the system logs the activity. | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | **Alternative Flow 1 - Unauthorized Access**  If a non-admin user tries to access the Admin Database:   1. System denies access and logs the attempt.   2. System notifies the admin of the unauthorized attempt | | |
| **Exceptions:** | **Exception 1 - System Error**  If an error occurs while updating data:  1) System displays an error message.  2) Admin is prompted to retry later. | | |
| **Includes:** | none | | |
| **Frequency of Use:** | Regularly; whenever an admin needs to update roles, permissions, or monitor logs. | | |
| **Special Requirements:** | 1. System must ensure **high-level security** for admin operations. 2. All admin actions must be **logged.**   3. Only **authorized admins** should be able to access and modify data. | | |
| **Assumptions:** | 1. Only authorized admins have access to the admin panel.   2. The system maintains an **activity log** for security monitoring. | | |
| **Notes and Issues:** | 1. Should the system support **multi-factor authentication** for admin access?   2. Should admins receive **real-time alerts** for suspicious activities? | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-1.7 | | |
| **Use Case Name:** | Symptoms Database Management | | |
| **Created By:** |  | **Last Updated By:** |  |
| **Date Created:** |  | **Last Revision Date:** |  |
| **Actors:** | **Primary:** System  **Secondary:** User, Admin | | |
| **Description:** | The system collects and stores user-reported symptoms, analyzes them for disease prediction, and provides health trend insights. | | |
| **Trigger:** | A user reports symptoms, or the system initiates symptom analysis. | | |
| **Preconditions:** | 1. The system must have an updated **database of diseases and symptoms**. | | |
| **Post conditions:** | 1. User-reported symptoms are **securely stored** in the database.   2. System processes symptoms for **disease prediction** and generates insights. | | |
| **Normal Flow:** | 1. User navigates to the **Report Symptoms** section. 2. System presents a **symptom entry form**. 3. User enters symptoms and submits the form. 4. System stores symptoms in the database. 5. System analyzes symptoms against the medical database.   6. System generates a **disease prediction report** with probability scores.  7. System updates health trend insights based on collected data. | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | **Alternative Flow 1 - No Symptoms Entered**  If the user does not enter symptoms and tries to proceed: 1. System prompts the user to enter at least one symptom.  2. User enters symptoms and continues. | | |
| **Exceptions:** | **Exception 1 - System Error**  If an error occurs while storing or analyzing symptoms:   1. System displays an error message.   2) User is prompted to retry later. | | |
| **Includes:** | **Use Case UC-2.1 (Predict Disease)** - System uses stored symptoms for disease prediction.  **Use Case UC-5.1 (Health Trend Analysis)** - System generates reports on health trends. | | |
| **Frequency of Use:** | Daily; whenever a user reports symptoms or the system performs analysis. | | |
| **Special Requirements:** | 1. The system must use **secure encryption** to store symptom data. 2. Data privacy must be ensured, following **health regulations**. 3. The system should provide **anonymized health trends** for research insights. | | |
| **Assumptions:** | 1. Users will provide **accurate symptom information**. 2. The system's disease prediction **accuracy improves over time** with more data. | | |
| **Notes and Issues:** | 1. Should users be able to edit or delete previously reported symptoms?   2. Should the system allow real-time symptom tracking for better analysis? | | |

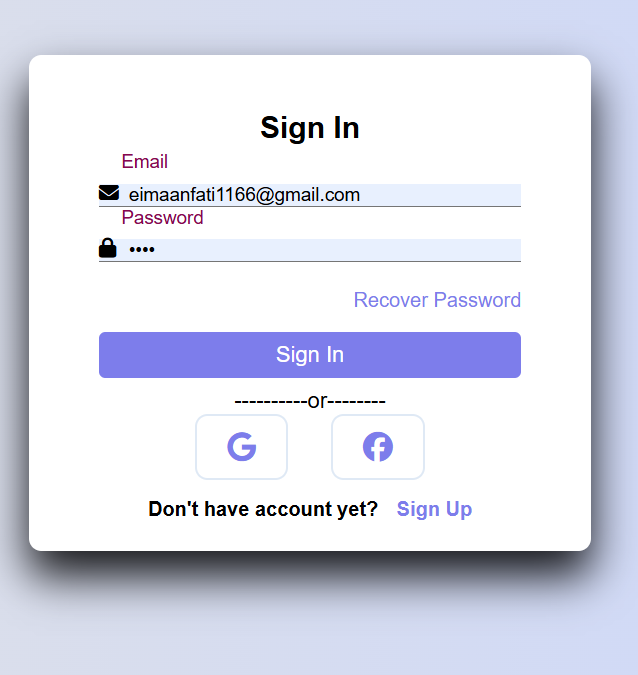
|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-1.8 | | |
| **Use Case Name:** | Medical History Database Management | | |
| **Created By:** |  | **Last Updated By:** |  |
| **Date Created:** |  | **Last Revision Date:** |  |
| **Actors:** | **Primary:** System  **Secondary:** User, Admin | | |
| **Description:** | The system stores and manages users' past medical diagnoses, treatments, and health conditions to provide personalized care and improve disease prediction accuracy | | |
| **Trigger:** | A user updates their medical history, or the system retrieves data for analysis and recommendations. | | |
| **Preconditions:** | 1. User must be **logged in** to update or view medical history. 2. System must **ensure secure storage**.   3. System must integrate with **disease prediction and recommendation modules**. | | |
| **Post conditions:** | 1. Medical history is **securely stored and updated**.   2. System uses medical history to **enhance disease prediction**.. | | |
| **Normal Flow:** | 1. User navigates to the **Medical History** section. 2. System displays past diagnoses, treatments, and health conditions. 3. User can **add, edit, or remove** medical history entries.   4. System **validates** and securely updates the database.  5. System uses updated medical history for **disease prediction and health recommendations**.System ejects ATM card] | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | NO ALTERNATIVE FLOW | | |
| **Exceptions:** | **Exception 1 - System Error**  If an error occurs while storing or retrieving medical history:   1. System displays an error message.   2) User is prompted to retry later | | |
| **Includes:** | **Use Case UC-2.1 (Predict Disease)** - System uses stored medical history for better predictions. | | |
| **Frequency of Use:** | On demand; whenever a user updates or views their medical history. | | |
| **Special Requirements:** | 1.Data encryption should be used to protect sensitive medical records.  2.. Users should have the ability to **export** their medical history. | | |
| **Assumptions:** | 1. Users provide **accurate medical information**.   2. The system ensures **data integrity** and prevents unauthorized access. | | |
| **Notes and Issues:** | 1. Should users be able to share their medical history with doctors or family members?   2. Should the system allow integration with **hospital or clinic databases** for automatic updates? | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-1.9 | | |
| **Use Case Name:** | API HEALTH INSIGHT MANAGEMENT | | |
| **Created By:** |  | **Last Updated By:** |  |
| **Date Created:** |  | **Last Revision Date:** |  |
| **Actors:** | **Primary:** System  **Secondary:** User, Admin | | |
| **Description:** | The system analyzes stored user data, including symptoms, medical history. | | |
| **Trigger:** | The user requests health insights, or the system automatically generates periodic health trend reports. | | |
| **Preconditions:** | 1. The user must be **logged in** and have data stored in the system. 2. The system must have access to **user symptoms, medical history, and lifestyle data**.   3. The system must utilize **AI-driven analysis** for trend identification. | | |
| **Post conditions:** | 1. User receives **personalized health insights** based on their data.   2. System updates its health trend analysis database. | | |
| **Normal Flow:** | 1. User navigates to the **Health Insights** section. 2. System retrieves user medical history, symptom logs, and other relevant data. 3. System analyzes the data to **identify patterns and trends**. 4. System generates a **personalized health report** with insights and recommendations.   5.System displays insights in an easy-to-understand format, including **charts, graphs, and summaries**.  6. User reviews insights and can take **recommended actions**. | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | **Alternative Flow 1 - Insufficient Data for Analysis**  If the system determines that there is not enough user data for meaningful insights:   1. System informs the user and suggests **logging more health data**.   2. User is prompted to enter missing information (e.g., symptoms, workout logs). | | |
| **Exceptions:** | **Exception 1 - System Error**  If an error occurs while processing health insights:   1. System displays an error message.   2) User is prompted to retry later. | | |
| **Includes:** | **Use Case UC-2.1 (Predict Disease)** - System uses health insights for improved disease predictions. | | |
| **Frequency of Use:** | On demand | | |
| **Special Requirements:** | 1. The system must **protect user privacy** . 2. AI-based analysis should be accurate and continuously updated for better predictions.   3. The system should present insights in **easy-to-understand language**. | | |
| **Assumptions:** | 1. Users will provide **accurate and sufficient** health data. 2. The system has **enough historical data** to generate meaningful insights. | | |
| **Notes and Issues:** | 1. Should users receive **alerts** if a concerning health trend is detected?   2. Should the system allow **integration with wearable devices** for real-time health tracking? | | |

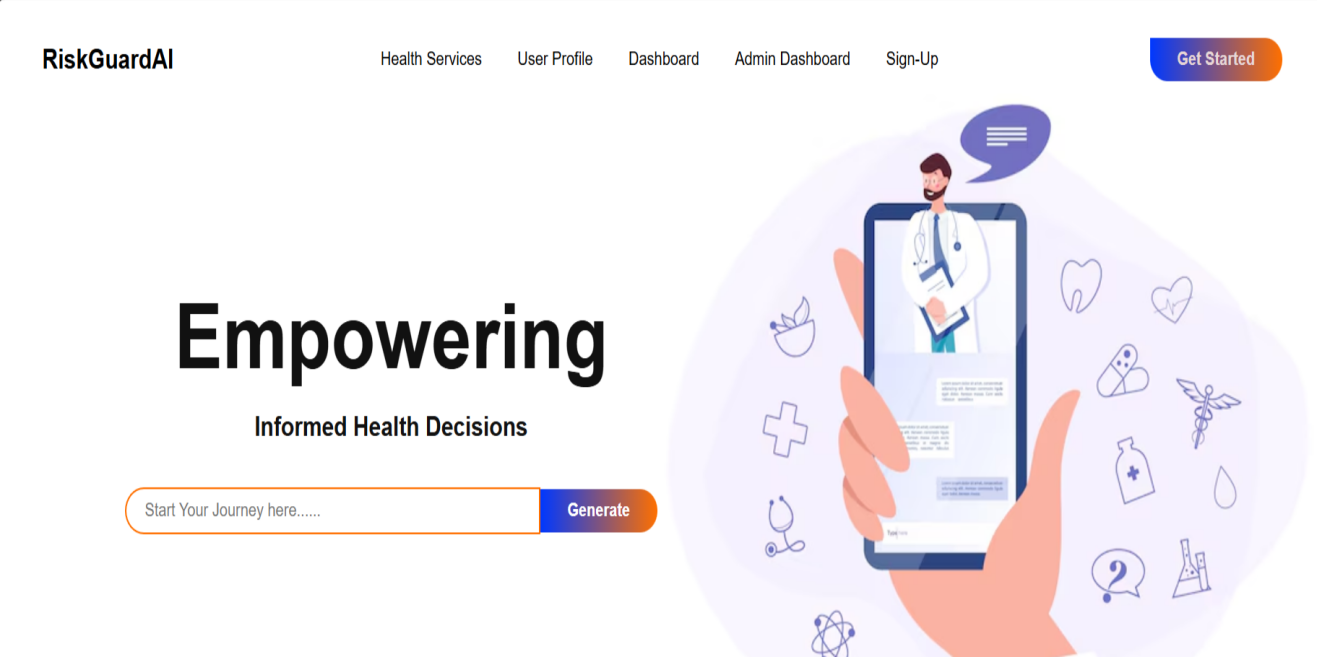
## User interfaces (mock screens)



Prototype1: (P1) register a new member



Prototype2: (P2) Login Page



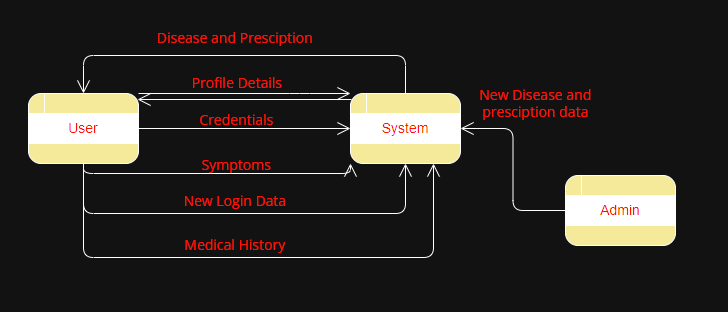
Prototype3: (P3) Front Page



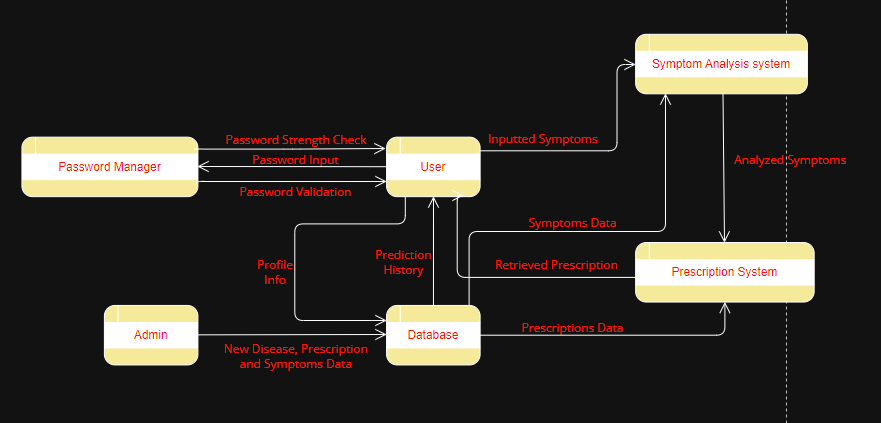
Prototype4: (P3) Dashboard Page

# Data flow diagram (optional)

## Data Flow Diagram Level 0



## Data Flow Diagram Level 1



## Data Flow Diagram Level 2



# System Design

## System Architecture Diagram

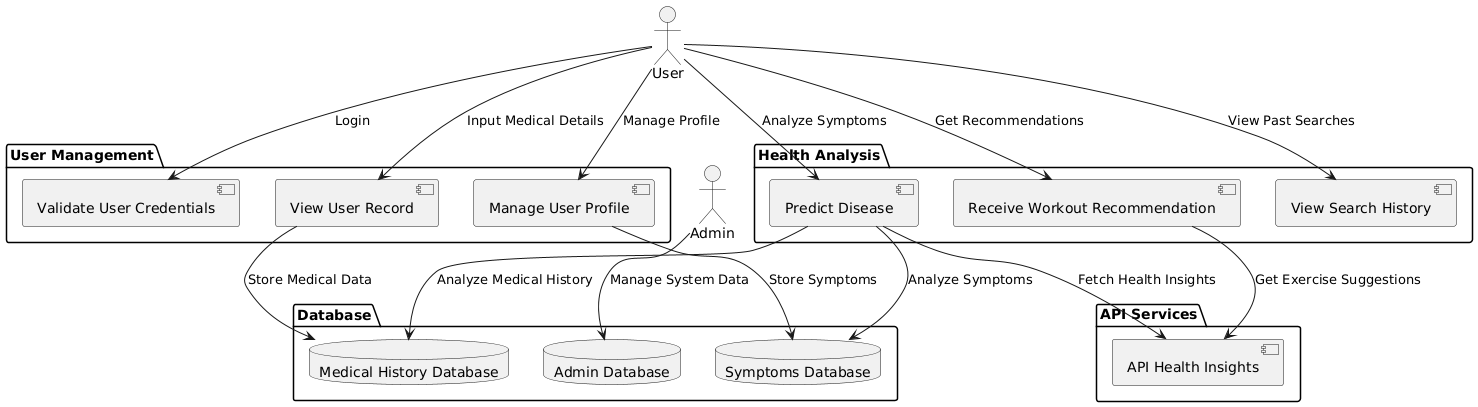
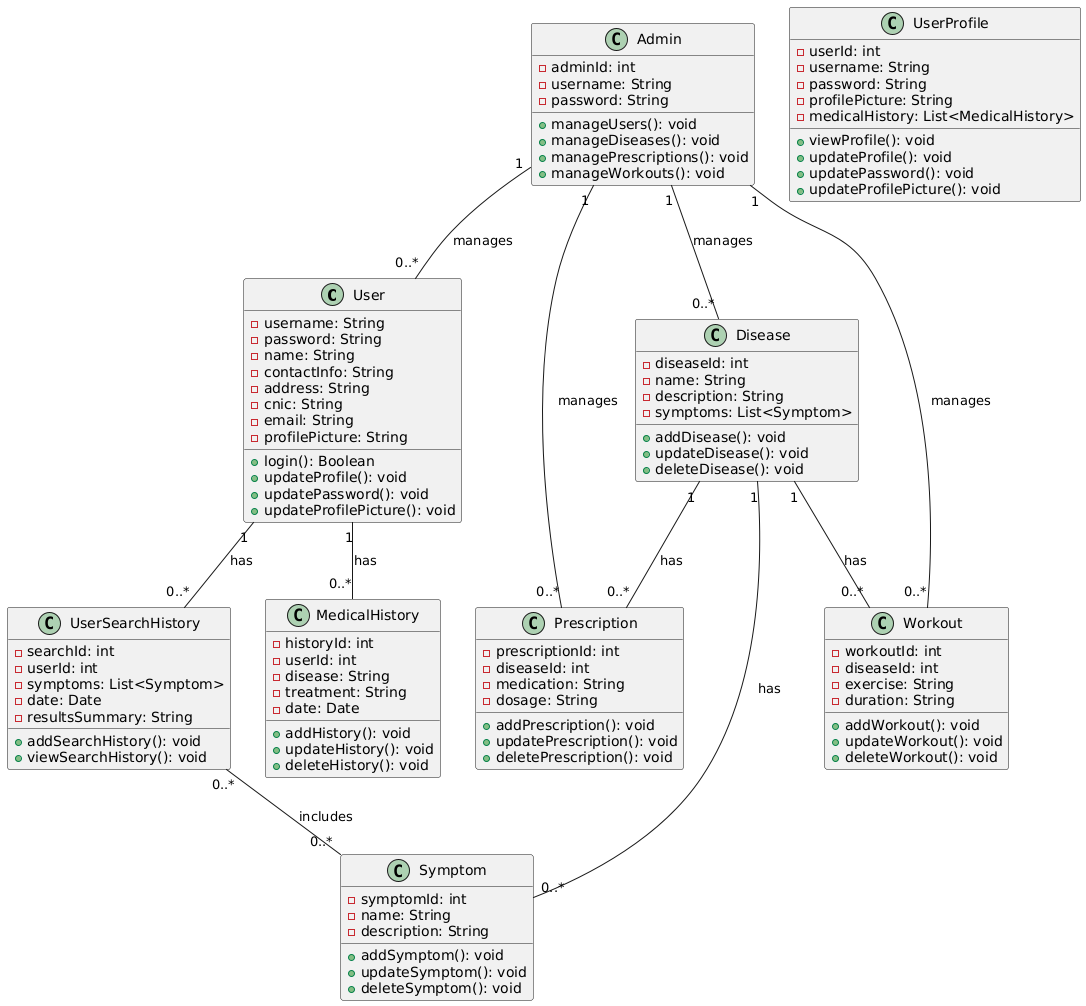


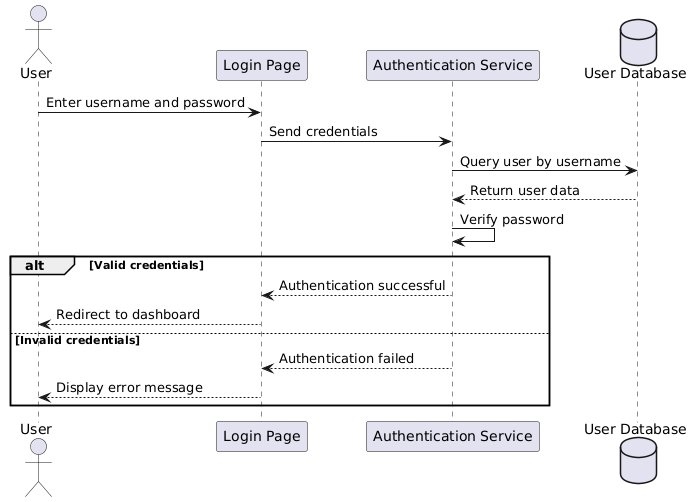
Figure 2: System Architecture

## Class Diagram

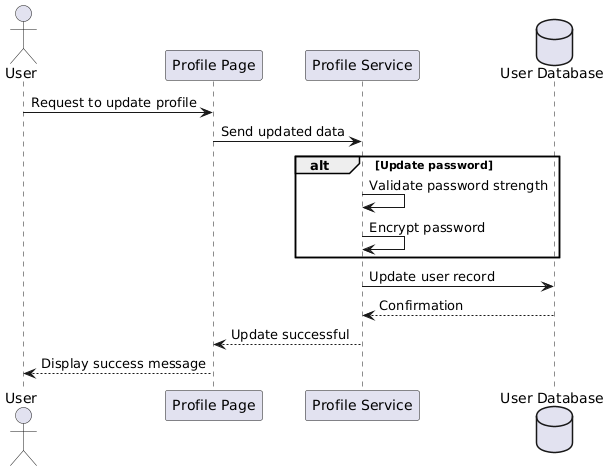


## Sequence Diagrams

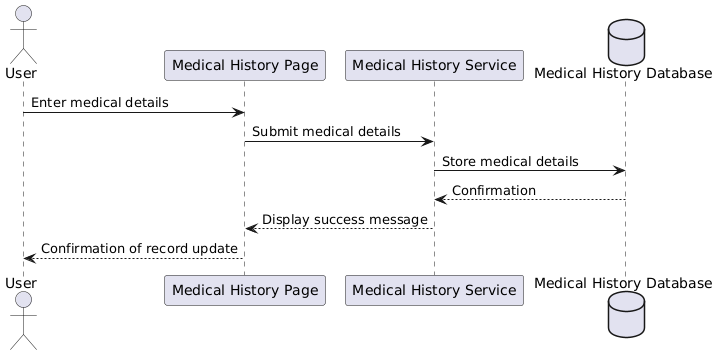
* **Validate User credential:**



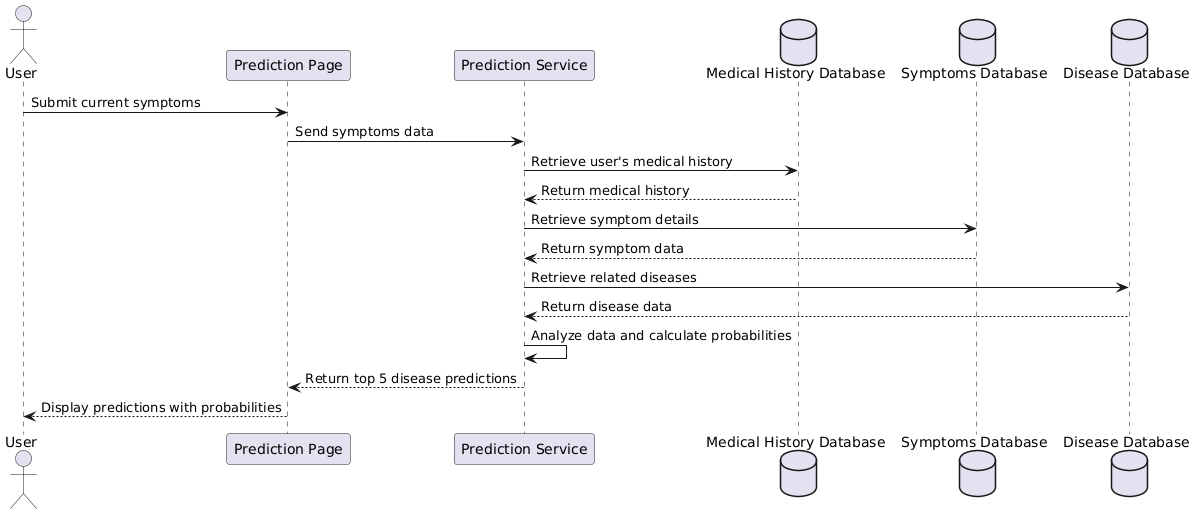
* **Manage User Profile:**



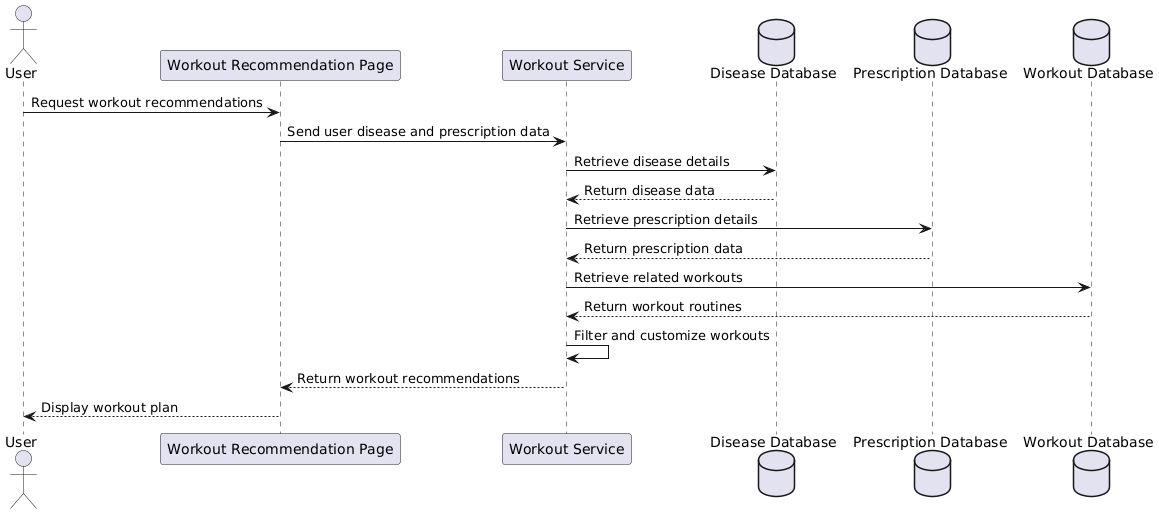
* **View User Record:**



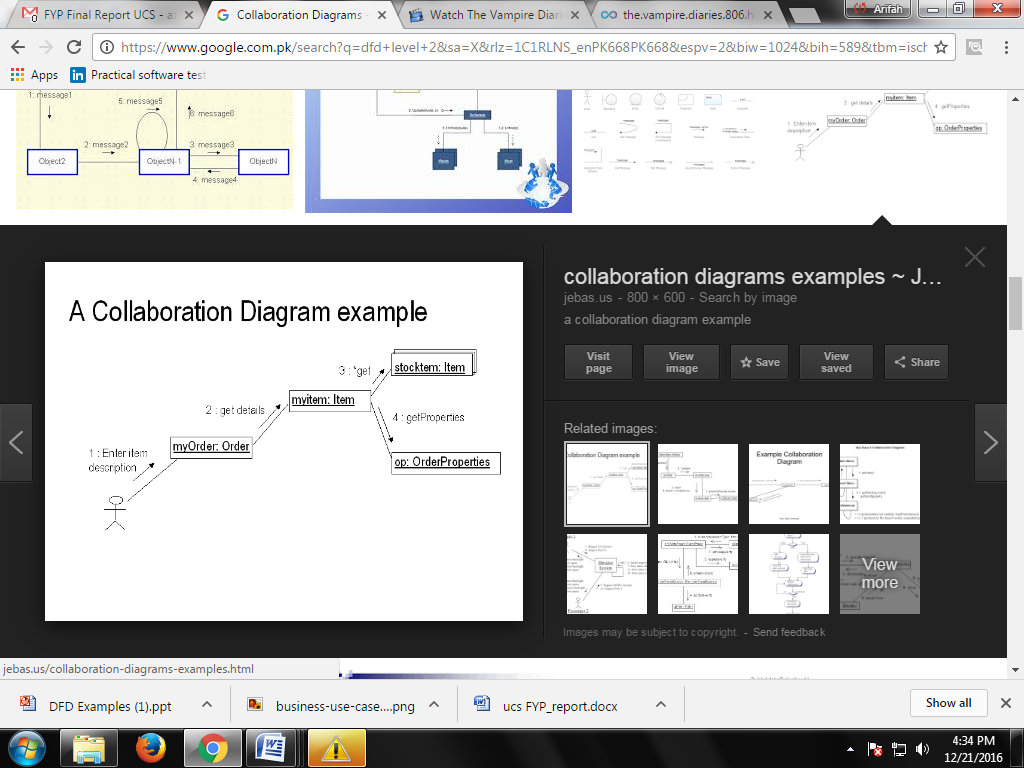
* **Predict Disease:**



* **Receive Workout Recommendation:**



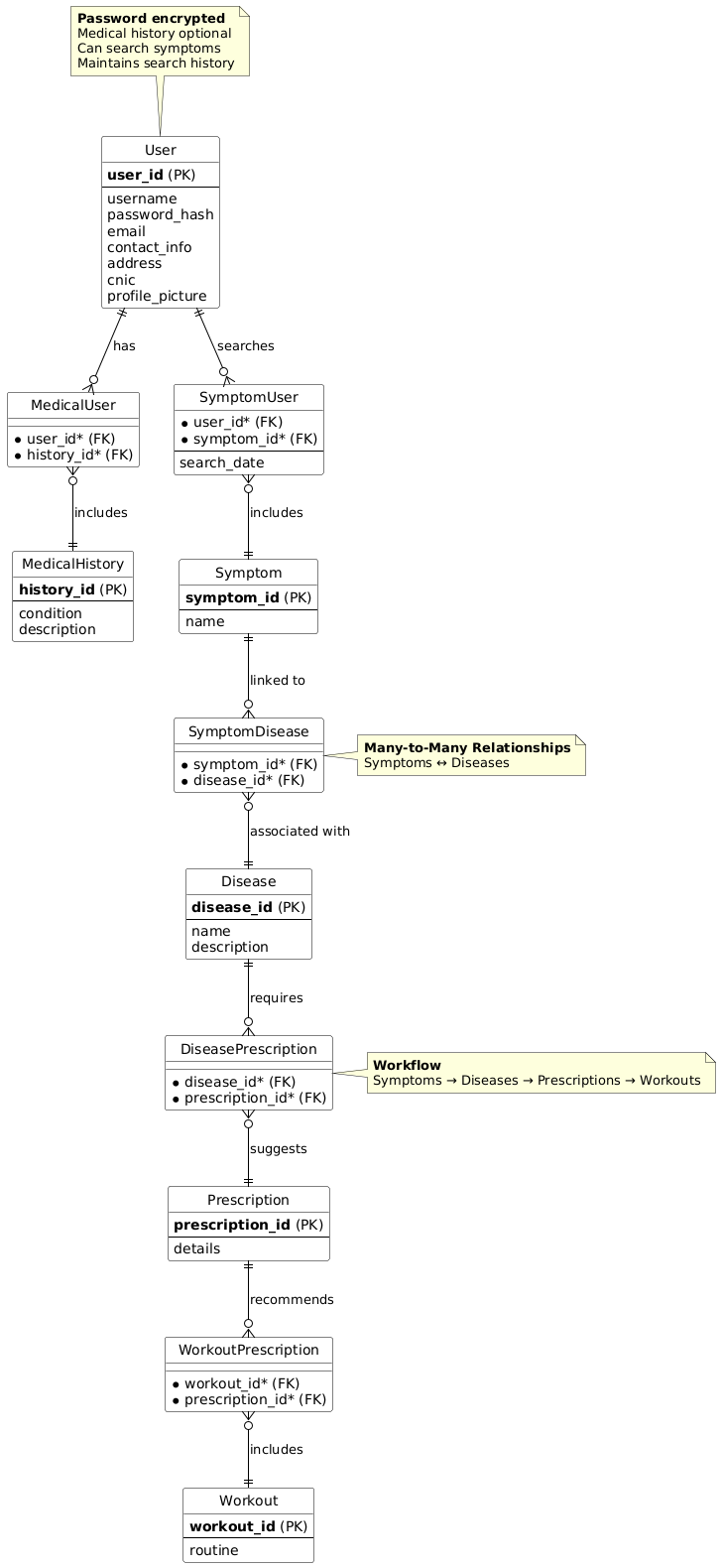
## llaboration Diagrams



## Other UMLs

This is optional. You may include any other UML to support your system.

## ERD



## Data Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element Name** | **Type** | **Validation** | **Mandatory** | **Remarks** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Username* | *String* | *Min 5, Max 50 characters* | *Yes* | *Unique identifier for user login* |
| *Password* | *String* | *Min 8 Characters, Must Include Special Character* | *Yes* | *Encrypted for security* |
| *Workout Recommendation* | *Text* | *System Generated* | *No* | *Personalized workout plan* |
| *Symptom* | *text* | *Free Text, Max 500 Characters* | *Yes* | *Used for disease prediction* |
| *Medical Record* | *text* | *Stored for 6 Months* | *No* | *Used for tracking past symptom searches* |

# Implementation details

## Development Setup

List your tools and technologies and their role in development.

## Deployment setup

How and where was your software deployed? Did you face any problems, how did you overcome these problems.

## Algorithms

Entire code of software is not required. Just highlight your important (user defined/ improved) algorithms.

## Constraints

### Assumptions

Things we assume will be true.

e.g.:

* *We will receive all necessary technical support from the engineers at cMeRun, Select and Mellon Bank to help design the interfaces between their systems and enGyro.*
* *All database maintenance will be handled by the client.*
* *There will be no real-time interfacing with any accounting systems.*

### System constraints

 A constraint specifies how the system must operate or how it must be built

### Restrictions

Constraints applied on the system by the client

### Limitations

Services your software is unable to provide

# Testing

## Extended Test Cases

## ****Test Case 1: Validate User Credentials (Login Process)****

| **Test Case ID: 1** | **Test Designed By: [Eimaan Fatima]** |
| --- | --- |
| **Test Module Name:** User Authentication | **Test Design Date:**  [2-2-2025] |
| **Test Priority:** High | **Test Executed By:** [Tester Name] |
| **Test Title/Name:** Validate User Login & Passwor | **Test Executed Date:** [Execution Date] |
| **Description:** Validate the login functionality by checking correct and incorrect user credentials. |  |
| **Pre-Condition:** The user must be registered and have valid login credentials. |  |

### ****Dependencies****

| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Open Login Page | - | Login page should load successfully | [Result] | [Pass/Fail] |  |
| 2 | Enter Username & Password | Username: testuser Email:  **eimaanfatima208@gmail.com**  Password: **Honda123** | System should validate credentials | [Result] | [Pass/Fail] |  |
| 3 | Click on Login Button | - | System should authenticate user and redirect to dashboard | [Result] | [Pass/Fail] |  |
| 4 | Enter Invalid Credentials | Username: invaliduser Password: WrongPass | System should show an error message | [Result] | [Pass/Fail] |  |

|  |
| --- |
| **Post Condition:** If valid credentials are used, the user is logged in successfully and redirected to the dashboard. If invalid credentials are used, an error message is displayed. |

|

## ****Test Case 2: Predict Disease Based on Symptoms****

| **Test Case ID: HRP-02** | **Test Designed By: [Mariam anwar]** |
| --- | --- |
| **Test Module Name:** Disease Prediction | **Test Design Date:**  [2-2-2025] |
| **Test Priority:** Critical | **Test Executed By:** [Tester Name] |
| **Test Title/Name:** Symptom-based Disease Prediction | **Test Executed Date:** [Execution Date] |
| **Description:** Verify that the system correctly predicts diseases based on user symptoms. |  |
| **Pre-Condition:** The user must be logged in and have access to the symptom input field. |  |

### ****Dependencies****

| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Navigate to User Dashboard | - | Dashboard should load with symptom input fields | [Result] | [Pass/Fail] |  |
| 2 | Enter Symptoms | Fever, Headache, Fatigue | System should accept input | [Result] | [Pass/Fail] |  |
| 3 | Click on Confirm Button | - | System should process symptoms and search the database | [Result] | [Pass/Fail] |  |
| 4 | Display Predicted Diseases | - | System should show top 5 diseases with probabilities | [Result] | [Pass/Fail] |  |

|  |
| --- |
| **Post Condition:** The system should display accurate disease predictions based on the entered symptoms. |

## ****Test Case 3: Receive Workout Recommendation****

| **Test Case ID: HRP-03** | **Test Designed By: [Farheen Azhar]** |
| --- | --- |
| **Test Module Name:** Workout Suggestions | **Test Design Date:** [Date] |
| **Test Priority:** Medium | **Test Executed By:** [Tester Name] |
| **Test Title/Name:** Workout Recommendations | **Test Executed Date:** [Execution Date] |
| **Description:** Verify that the system suggests appropriate workouts based on diagnosed diseases. |  |
| **Pre-Condition:** The user must have a diagnosed disease or enter a known disease manually. |  |

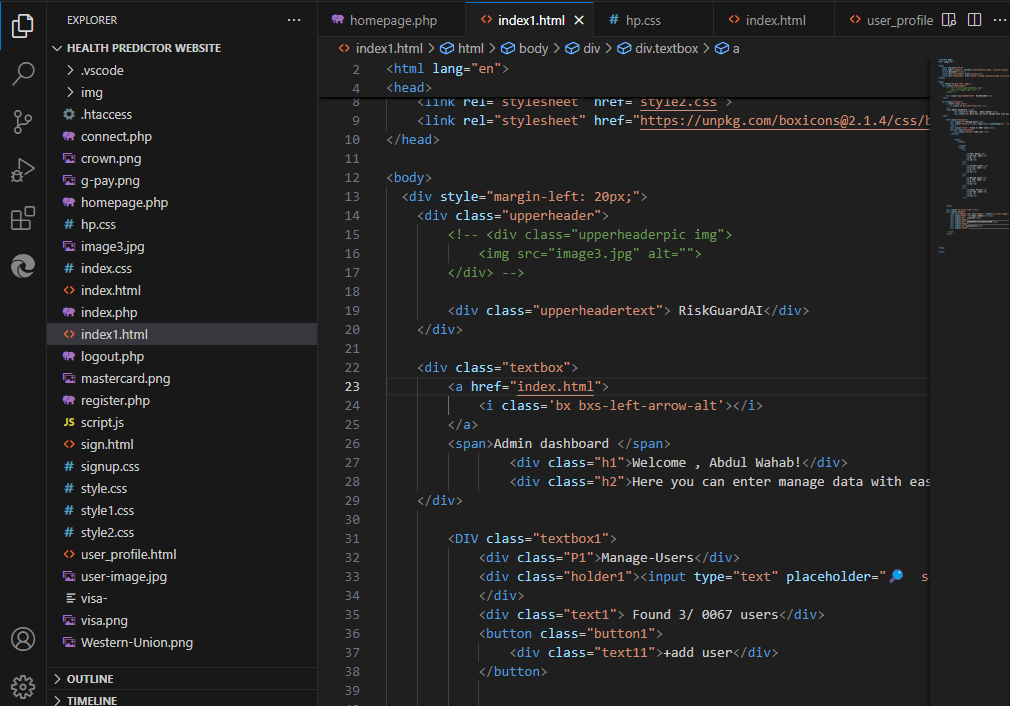
### ****Dependencies****

| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Navigate to User Dashboard | - | Dashboard should load properly | [Result] | [Pass/Fail] |  |
| 2 | Enter a Disease | Arthritis | System should accept input | [Result] | [Pass/Fail] |  |
| 3 | Click on Get Workout Plan | - | System should retrieve workout recommendations | [Result] | [Pass/Fail] |  |
| 4 | Display Workouts | - | System should display a list of recommended exercises | [Result] | [Pass/Fail] |  |

|  |
| --- |
| **Post Condition:** The user receives a workout routine based on their diagnosed disease |

## Decision Table

### Code snippet

****

### Decision coverage table

**User Login Validation:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition** | **R1** | **R2** | **R3** | **R4** |
| **Username** | F | F | T | T |
| **Password** | F | T | F | T |
| **Actions** | **R1** | **R2** | **R3** | **R4** |
| Access Dashboard  (both user and password were correct) | F | F | F | T |
| Invalid Username | T | T | F | F |
| Invalid Password | T | F | T | F |
| Error Message (both User and Password were wrong) | T | T | T | F |

**Disease Prediction Based on Symptoms:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Condition** | **Rule 1** | **Rule 2** | **Rule 3** | **Rule 4** | **Rule 5** | **Rule 6** | **Rule 7** | **Rule 8** |
| **Valid symptoms** | **Y** | **Y** | **Y** | **F** | **Y** | **F** | **F** | **F** |
| **Disease Exists in Database** | **Y** | **Y** | **F** | **Y** | **F** | **F** | **Y** | **F** |
| **User Medical History** | **Y** | **F** | **Y** | **Y** | **F** | **Y** | **F** | **F** |
| **Action** |  |  |  |  |  |  |  |  |
| **Display Disease Predictions(top 3 disease)** | **Y** | **Y** | **F** | **F** | **F** | **F** | **F** | **F** |
| **Provide Prescriptions(if user provide medical history)** | **Y** | **F** | **F** | **F** | **F** | **F** | **F** | **F** |
| **Suggest Workouts** | **Y** | **Y** | **F** | **F** | **F** | **F** | **F** | **F** |
| **Error Message** | **F** | **F** | **Y** | **Y** | **Y** | **Y** | **Y** | **Y** |

## Traceability Matrix

### RID vs UCID (requirements vs use cases)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UCID/RID** | **R**  **1** | **R**  **2** | **R**  **3** | **R**  **4** | **R**  **5** | **R**  **6** | **R**  **7** | **R**  **8** | **R**  **9** | **R**  **10** | **R**  **11** | **R**  **12** | **R**  **13** | **R**  **14** | **R**  **15** | **R**  **16** | **R**  **17** | **R**  **18** | **R**  **19** | **R**  **20** | **R**  **21** |
| UC 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 9 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 11 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 12 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 19 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 20 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 21 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 22 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 23 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 24 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 26 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 27 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### Prototypes (RID vs PID)

### Test Cases (RID vs TID)

### Coverage (UCID vs TID)

# Results/Output/Statistics

## %completion

Use the matrix & values from 7.3.1 to show that all requirements are being fulfilled.

## %accuracy

Use the matrix & values from 7.3.3 to show that all requirements have been implemented correctly.

## %correctness

Use the matrix & values from 7.3.4 to show that all requirements have been tested to be conforming to requirements.

# Conclusion

# Future work

# Bibliography

Use IEEE or ACM format for citations

## Books

## Journals

## Articles

## Research papers

## Other References

# Appendix

## Glossary of terms

## Pre-requisites

Must use contents of development/ deployment setup & external system dependencies