CS19442 SOFTWARE ENGINEERING CONCEPTS LAB

LABORATORY RECORD NOTEBOOK

DISEASE PREDICTION SYSTEM

Submitted by

DIVYA SUNDHARI S M	**220701068**
HARIPRIYA P	**220701086**
HINDUSHA K	**220701092**
HARI BALAJI J C	**220701080**
GIRIDHAR M	**220701074**
JANANI P K	**220701098**

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

2023-24

RAJALAKSHMI ENGINEERING COLLEGE, THANDALAM

AN AUTONOMOUS INSTITUTE

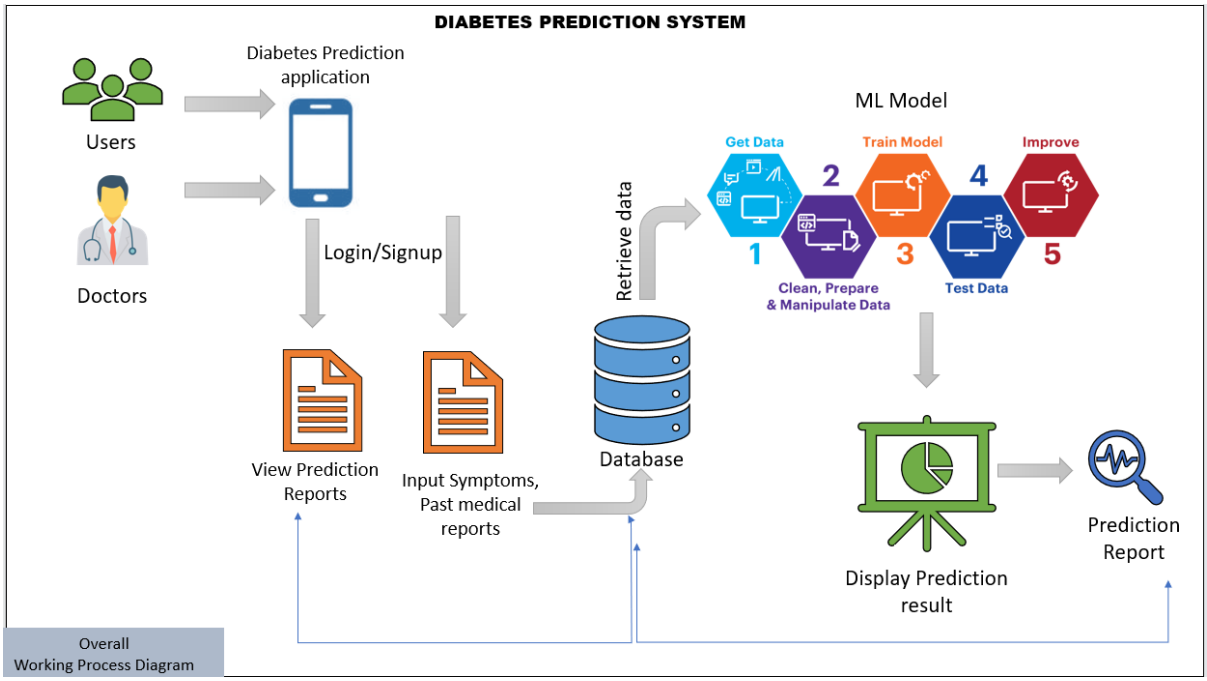
CHENNAI-602105

TABLE OF CONTENTS

CHAPTER 1

PROJECT OVERVIEW

- The Diabetes Prediction System uses machine learning to forecast an individual's risk of developing diabetes based on factors like age, weight, family medical history, diet, and exercise habits.
- It involves collecting, cleaning, and analyzing data, selecting relevant features, training models, evaluating performance, and deploying the system for user input and prediction.
- Diabetes Prediction System empower individuals with personalized risk assessments, enabling early intervention and preventive measures to mitigate the onset or progression of diabetes.



1.1 PROBLEM STATEMENT

- The Diabetes Prediction System offers a proactive approach to healthcare by leveraging machine learning to provide personalized risk assessments for diabetes.
- This system aims to enable early detection, personalize healthcare, and optimize resource allocation, ultimately improving patient outcomes and understanding of diabetes risk factors.
- It optimizes healthcare resources by targeting high-risk individuals for further testing and continuously improves through ongoing data inputs, making it more efficient and effective than traditional methods.

1.2 DATA PERSPECTIVE

- The project utilizes a diverse range of data sources to predict diabetes risk accurately. This includes demographic information, medical history (such as family history of diabetes, past diagnoses, medications), lifestyle factors (dietary habits, exercise routines, smoking status), physiological measurements (blood pressure, body mass index), and biomarker data (blood glucose levels, lipid profiles).
- Additionally, genetic data may be incorporated to assess genetic predisposition to diabetes and identify relevant genetic markers associated with the condition. This multi-dimensional approach to data collection ensures a comprehensive assessment of individual risk factors contributing to diabetes development.

1.3 USER BENEFITS

- **Early Detection and Intervention**: The system identifies individuals at high risk of diabetes before symptoms appear, enabling early intervention and preventive measures.
- **Personalized Risk Assessment**: Users receive personalized risk assessments based on their health profiles, allowing for tailored interventions and treatment plans.
- **Improved Patient Outcomes**: Timely identification of risk factors leads to better patient outcomes, reduced disease burden, and lower healthcare costs.
- **Data-Driven Decision Making**: Healthcare providers can make informed clinical decisions based on predictive insights, leading to more effective patient management and resource allocation.

CHAPTER 2

REQUIREMENTS AS USER STORIES

2.1 USER STORIES

2.1.1 **Signup page**: As a new user, I want to be able to sign up for an account on the disease prediction system so that I can access the system's features and receive personalized disease predictions.

Acceptance Criteria:

- The signup page should have a clear and concise form that prompts the user to enter their first name, last name, email address, and password.
- The system should validate the email address entered by the user to ensure it is in the correct format and that it is not already registered in the system.
- The system should enforce a minimum password length and require a mix of uppercase and lowercase letters and numbers.
- The system should display a confirmation message upon successful registration and send a confirmation email to the user's registered email address.

Poker point Estimation: 5 points

2.1.2 **Symptom input:** As a user, I want to be able to input my symptoms into the disease prediction system so that I can receive a personalized disease prediction.

Acceptance Criteria:

- The system should provide a clear and concise form for users to input their symptoms.
- The system should validate the input to ensure that it is in the correct format and that it is not empty.
- The system should display a confirmation message upon successful symptom input and provide a prediction based on the inputted symptoms.

Poker point Estimation: 3 points

2.2.3 Predict Diseases: As a user, I want the system to accurately predict diseases based on input data, so that I can take proactive measures for my health.

Acceptance Criteria:

- Prediction Algorithm: Generates list of potential diseases with confidence scores.
- Explanation: Provides detailed explanation for each prediction.
- Notification: Alerts user of new predictions.

Poker point Estimation: 8 points

2.1.4 Collect Health Data: As a user, I want to input and manage health data, so that the system can predict diseases accurately.

Acceptance Criteria:

- Form: Includes personal, medical, and lifestyle data.
- Validation: Ensures completeness and correctness of data.
- Storage: Securely stores encrypted data.
- Management: Allows viewing, updating, and deleting of data.

Poker point Estimation: 13 points

2.1.5 Personalized Health Assessment: As a user, I want personalized health assessments to understand my health status better, so that I can make informed decisions for my well-being.

Acceptance Criteria:

- Customization: Tailors assessments based on user data and preferences.
- Insights: Provides detailed insights and recommendations for health improvement.
- Progress Tracking: Allows users to track their progress over time.

Poker point Estimation: 8 points

2.1.6 Display Prediction Results: As a user, I want to see disease predictions clearly presented, so that I can understand my health risks easily.

Acceptance Criteria:

- Clarity: Predictions are presented in a clear and understandable format.
- Accuracy: Predictions are accurate and based on reliable data.
- Accessibility: Predictions are accessible to users with disabilities.

Poker point Estimation: 5 points

2.1.7 Remedies Database: As a user of the disease prediction system, I want to be able to access a remedies database so that I can view potential treatments for my predicted diseases.

Acceptance Criteria:

- The remedies database should be easily accessible from the main menu of the disease prediction system.
- The remedies database should provide a clear and concise list of potential treatments for each predicted disease.
- Each treatment option in the remedies database should include a brief description, potential side effects, and any necessary precautions.

Poker Point Estimation: 5 points

2.1.8 System input: As a user of the disease prediction system, I want to be able to save my favorite treatments in the remedies database so that I can easily access them later.

Acceptance Criteria:

- The remedies database should have a "favorites" feature that allows users to save their preferred treatments.
- Users should be able to easily access their saved treatments from the main menu of the remedies database.
- Users should be able to remove treatments from their favorites list at any time.

Poker Point Estimation: 5 points

2.2 Functional Requirements:

1. **Signup Page:**

- The system shall provide a signup page with fields for first name, last name, email address, and password.
- Upon submission, the system shall validate the email address format and check if it's already registered.
- After successful registration, the system shall display a confirmation message and send a confirmation email to the user.

2. **Symptom Input:**

- The system shall offer a form for users to input their symptoms.
- Input validation shall ensure the correctness of format and non-emptiness.
- Upon successful submission, the system shall display a confirmation message and provide a disease prediction based on the inputted symptoms.

3. **Predict Diseases:**

- The system shall accurately predict diseases based on input data.
- Prediction algorithm shall generate a list of potential diseases with confidence scores.
- The system shall provide detailed explanations for each prediction and alert the user of new predictions.

4. **Collect Health Data:**

- The system shall provide forms for users to input and manage personal, medical, and lifestyle data.
- Data validation shall ensure completeness and correctness.
- Secure storage of encrypted data shall be provided, allowing viewing, updating, and deletion of data.

5. **Personalized Health Assessment:**

- The system shall offer personalized health assessments tailored based on user data and preferences.
- Detailed insights and recommendations for health improvement shall be provided.
- Progress tracking functionality shall allow users to monitor their health status over time.

6. **Display Prediction Results:**

- The system shall present disease predictions in a clear and understandable format.
- Predictions shall be accurate and based on reliable data.
- Accessibility standards shall be followed to ensure predictions are accessible to users with disabilities.

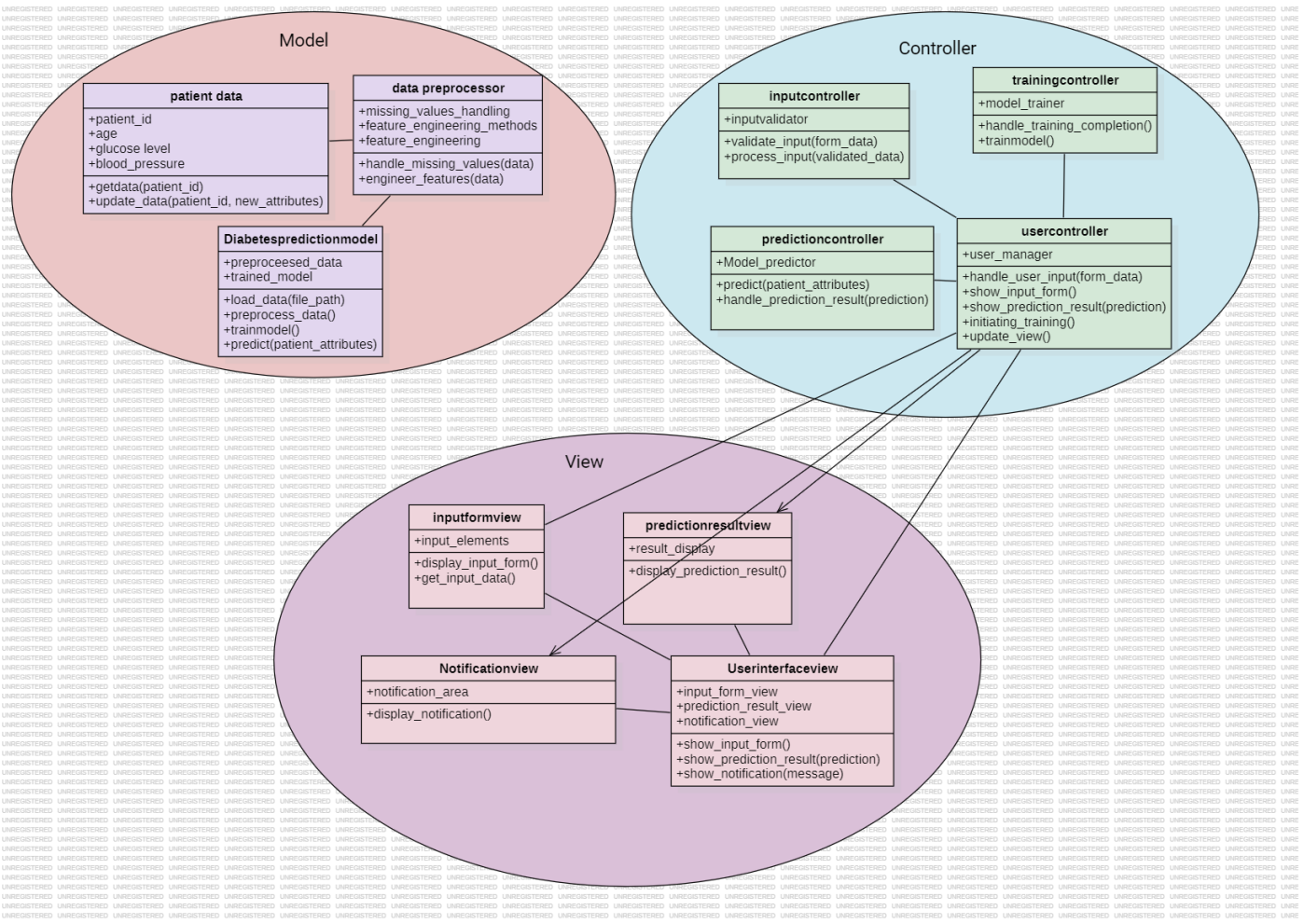
2.3 Non-Functional Requirements:

- **Accuracy:** Achieve a prediction accuracy of at least 90% on a standardized dataset, benchmarked against medical expert diagnoses.
- **Response Time**: Provide predictions within 100 milliseconds (ms) for individual requests and maintain an average response time of 500 ms under peak load conditions.
- **Reliability**: Ensure system uptime of at least 99.9% over a rolling 30-day period, with automated monitoring and failover mechanisms in place to mitigate service disruptions.
- **Scalability**: Scale horizontally to support a user base of up to 100,000 concurrent users and handle a daily prediction workload of 1 million requests, with linear scalability beyond this threshold.
- **Security:** Implement end-to-end encryption using industry-standard algorithms (e.g., AES-256) for data in transit and at rest, enforce multi-factor authentication for user access, and conduct regular penetration testing to identify and remediate vulnerabilities.

CHAPTER 3: ARCHITECTURE DIAGRAM

3.1 MVC ARCHITECTURE:

In the Model-View-Controller (MVC) architecture, the Model manages the data and business logic, the View displays information to the user, and the Controller handles user input and updates the Model or View accordingly. This division of responsibilities simplifies application development and maintenance by separating concerns and promoting modularity.



MVC Architecture Diagram for Disease Prediction System

Model Component:

- **PatientData:** Manages patient-specific data and provides it to the DataPreprocessor.
- **DataPreprocessor:** Handles data preprocessing and passes preprocessed data to the DiabetesPredictionModel.
- **DiabetesPredictionModel:** Manages the machine learning model, receives preprocessed data, and interacts with controllers for training and prediction.

View Component:

- **InputFormView:** Collects user input and sends it to the UserController.
- **PredictionResultView:** Displays prediction results received from the UserController.
- **NotificationView:** Displays notifications/error messages from the UserController.
- **UserInterfaceView**: Orchestrates the display of input form, prediction results, and notifications.

Controller Component:

- **InputController:** Handles user input validation and processing, communicates with the UserController.
- **TrainingController:** Manages model training, interacts with the UserController.
- **PredictionController**: Manages model prediction, interacts with the UserController.
- **UserController:** Coordinates interactions between views and controllers, updates views with relevant data.

Workflow between Model, View and Controller

- In the Diabetes Prediction System, user interaction begins as users input their data via the InputFormView. The InputController validates and processes this input, ensuring its accuracy and relevance.
- Once validated, the data is sent to the DiabetesPredictionModel for further processing. The Model component handles the training of the machine learning model or makes predictions based on the received data.
- Upon completion of model processing, the PredictionController communicates the results back to the PredictionResultView, where users can view the predicted outcomes. Additionally, any notifications or error messages generated during the process are displayed in the NotificationView, providing users with feedback on the system's status.
- Throughout this process, the Views update the user interface to reflect the outcomes of the Controller actions, creating a smooth and intuitive user experience. This interconnected flow enables users to input data, receive predictions, and stay informed about the system's status in a coherent and organized manner.

Architecture Pattern

- The chosen architecture pattern is the Microservices Architecture. It's selected due to its ability to modularize the system into **independent** services, each responsible for a specific domain, such as user management, data collection, and prediction.

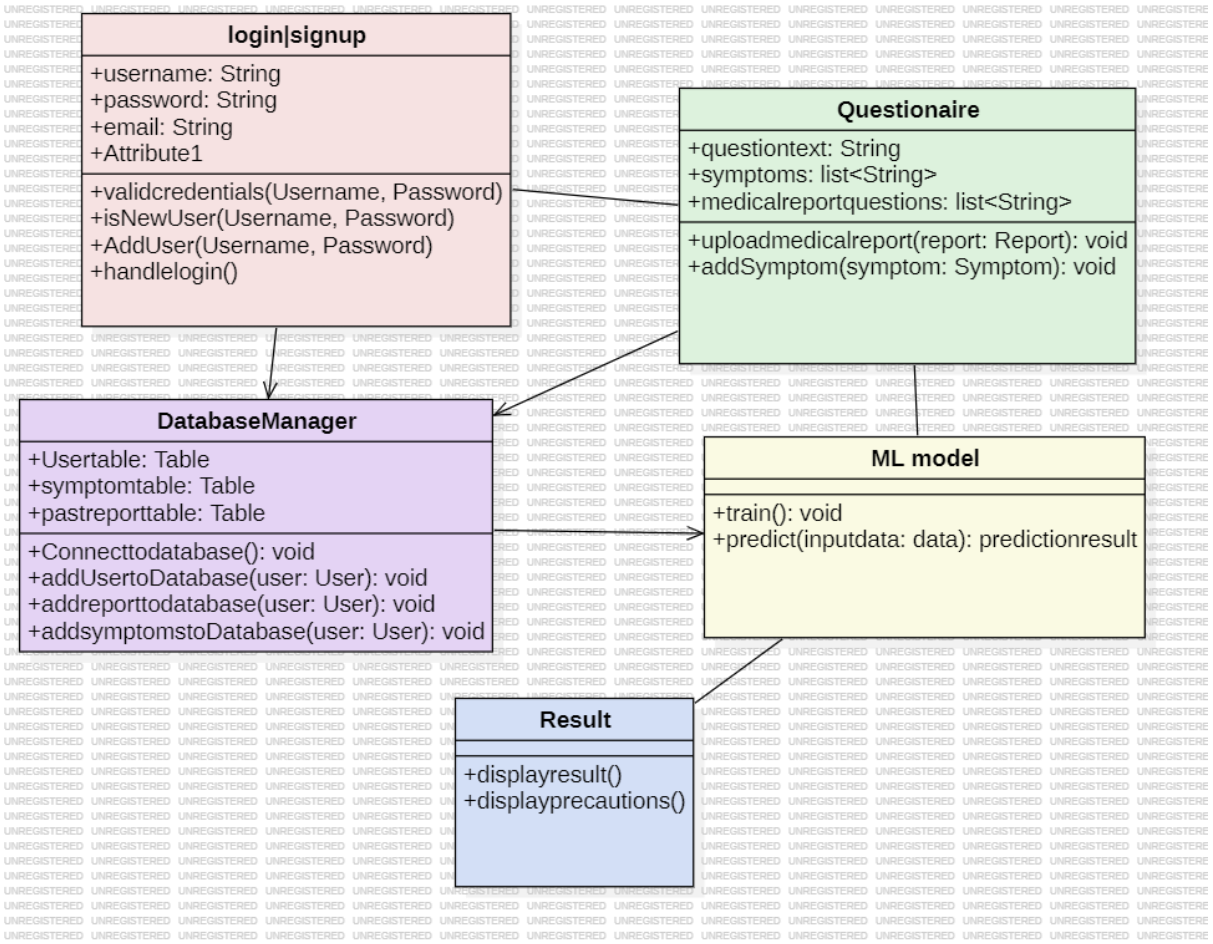
- This approach facilitates scalability, flexibility, and resilience, allowing individual services to be developed, deployed, and scaled independently. Microservices also promote agility, as they enable teams to work on different services concurrently, reducing dependencies and speeding up the development process.

Design Principles

- In designing the system, several design principles are adhered to, including SOLID (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) and DRY (Don't Repeat Yourself).

- These principles promote modularity, maintainability, and extensibility of the system. By adhering to SOLID principles, the system's components are designed to be loosely coupled and highly cohesive, facilitating easier testing, debugging, and evolution. DRY principle ensures code reusability and reduces redundancy, resulting in a cleaner, more concise, and less error-prone codebase

3.2 CLASS DIAGRAM



CLASS DIAGRAM EXPLANATION:

Login Signup:

Purpose: The **Login Signup** class is responsible for managing user authentication and registration. It handles the processes of validating user credentials, checking if a user is new, adding a new user to the system, and managing the login process.

Database Manager:

Purpose: The **Database Manager** class handles interactions with the database. It manages the user table, symptom table, and past report table. This class is responsible for connecting to the database, adding users, adding reports, and storing symptoms.

Questionnaire:

Purpose: The **Questionnaire** class collects user input related to symptoms and medical history. It includes methods to upload medical reports and add symptoms to the user's profile.

ML Model:

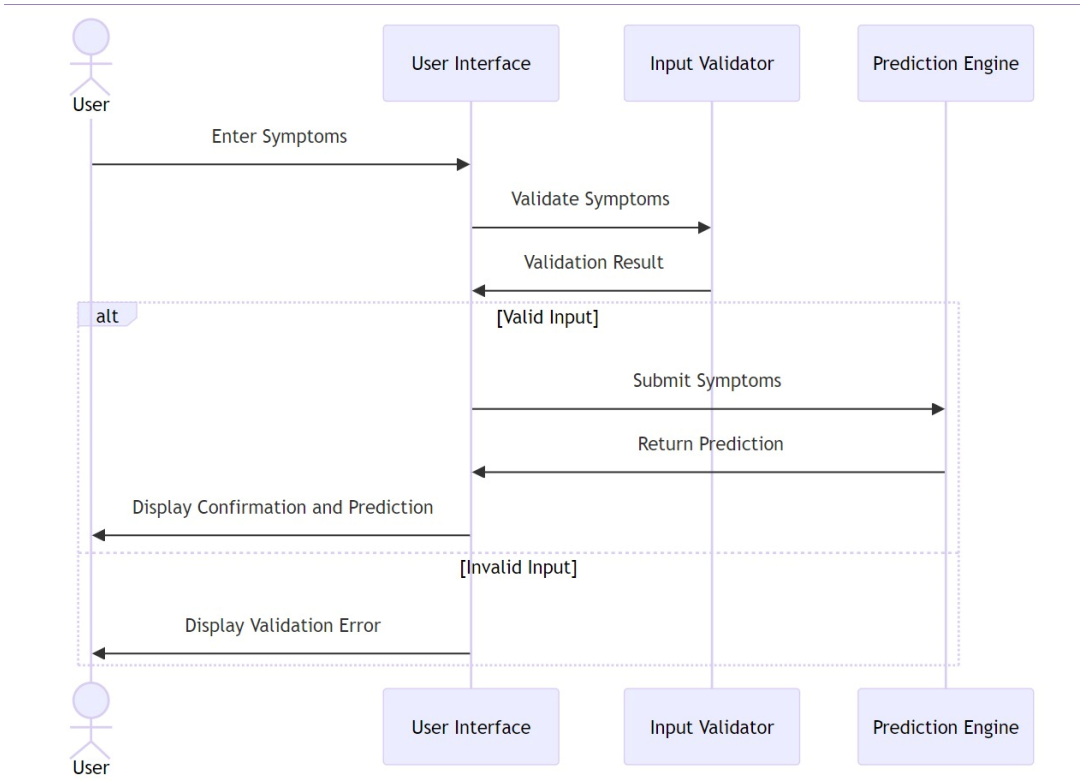
Purpose: The **ML Model** class represents the machine learning model used for predicting diseases. It includes functionality to train the model and make predictions based on input data.

Result:

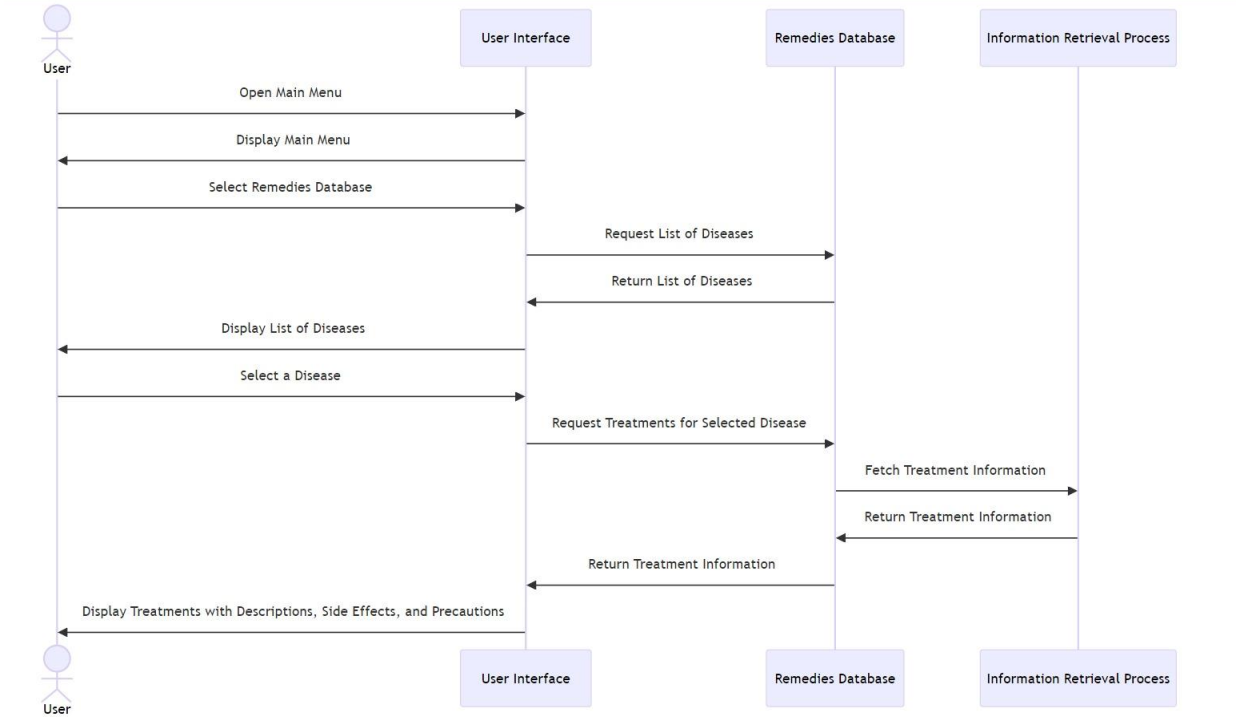
Purpose: The **Result** class is responsible for displaying the outcomes of the disease predictions. It shows the results and any precautionary measures that should be taken based on the prediction.

3.3 SEQUENCE DIAGRAM

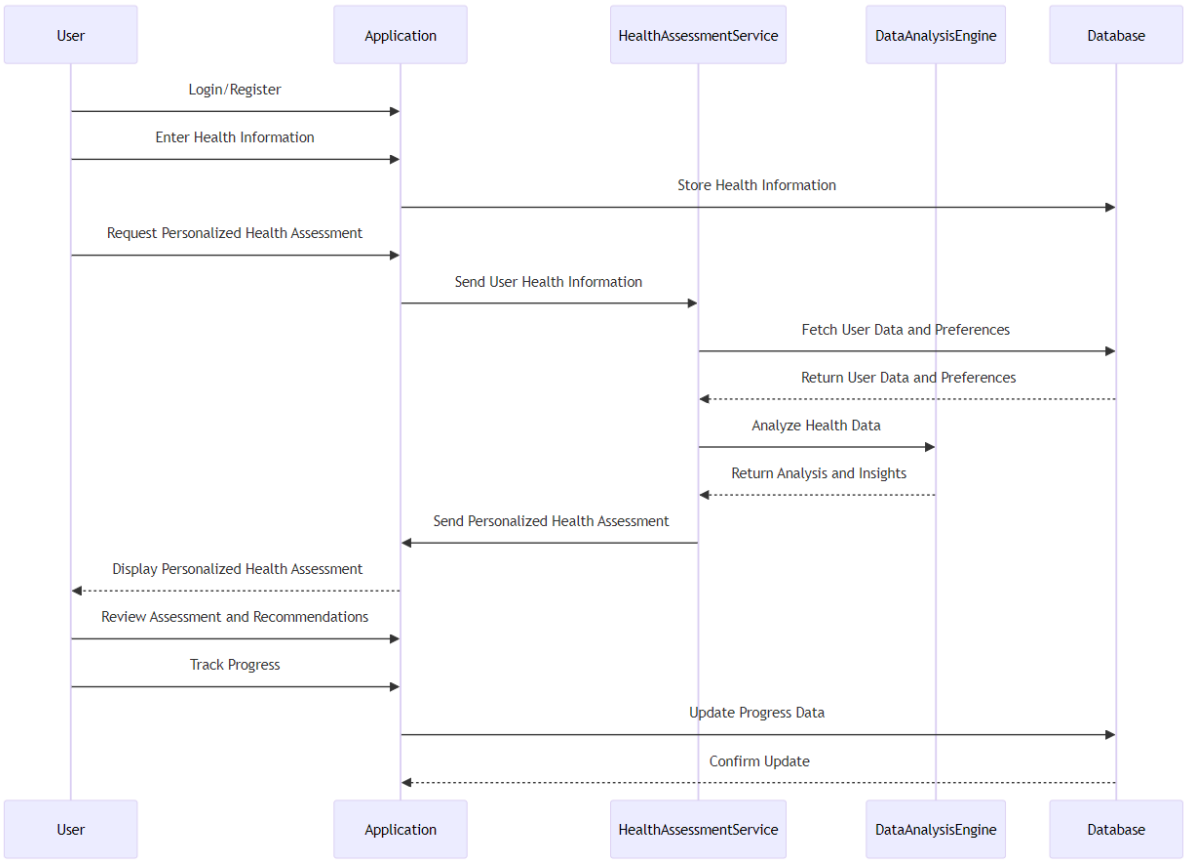
1. User Story : Symptom Input



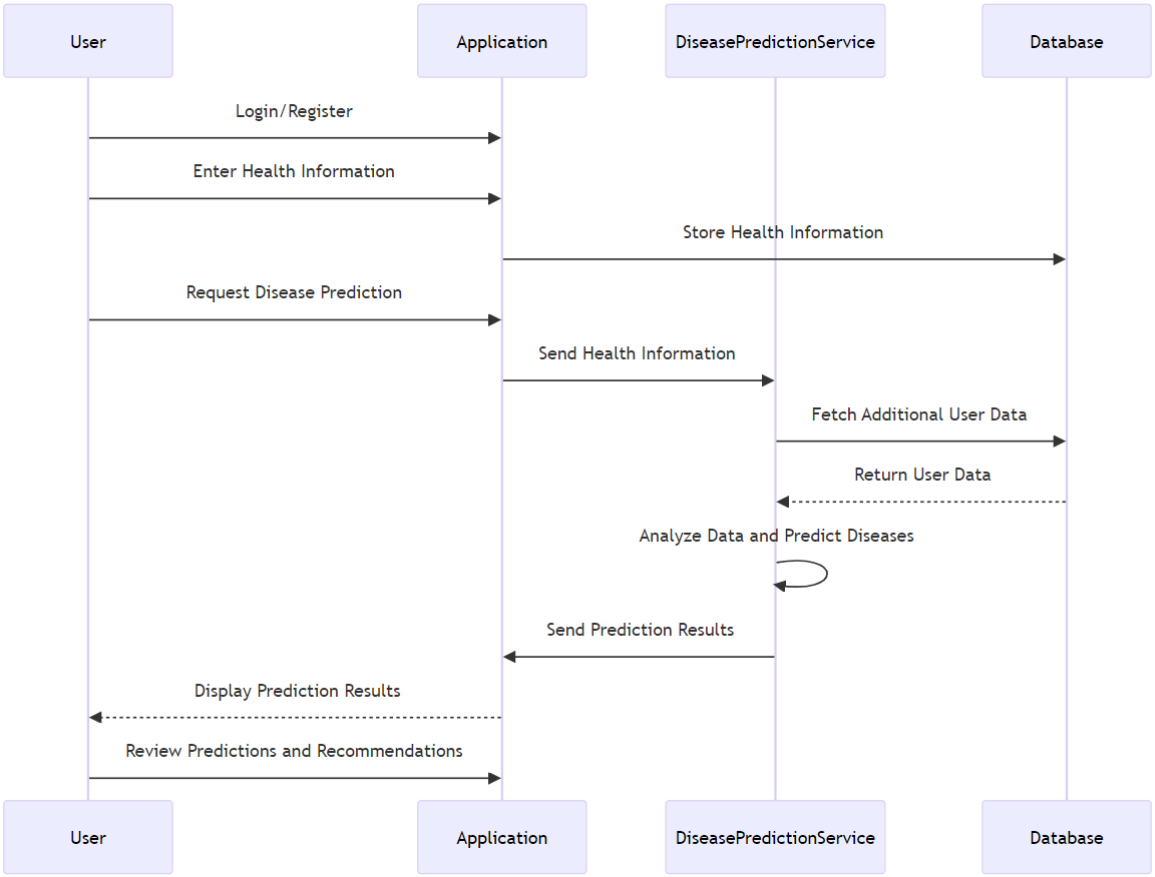
2. User Story: Remedies Database



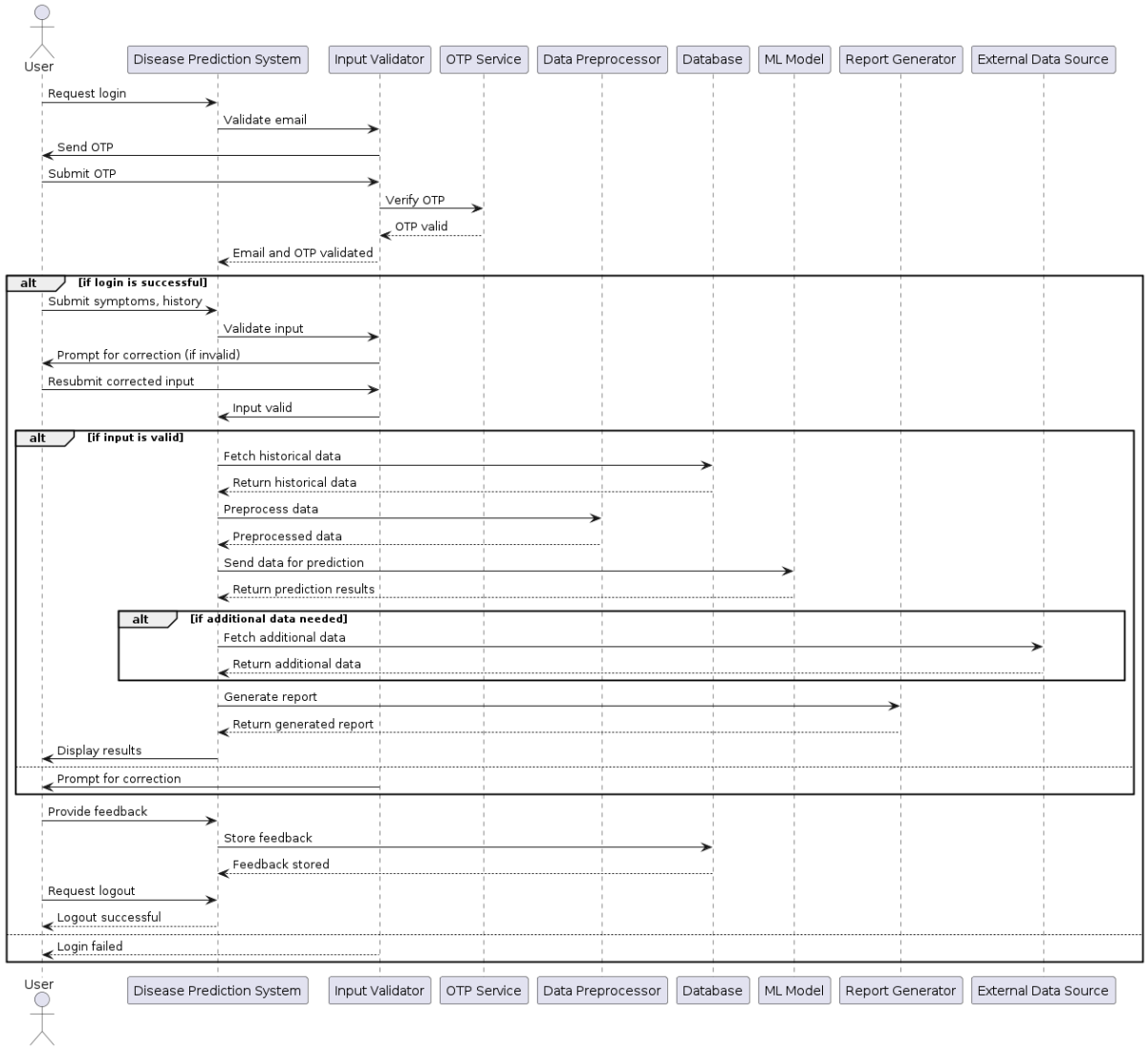
3. User Story: Personalized Health Assessment



4. User Story: Predict Diseases



Sequence Diagram for Diabetes Prediction System Workflow



SEQUENCE DIAGRAM EXPLANATION

Entities:

- **User**: The person interacting with the disease prediction system.
- **Disease Prediction System**: The central system that orchestrates the entire process.
- **Input Validator**: Validates user inputs such as email, symptoms, and medical history.
- **OTP Service**: Handles the generation and verification of One-Time Passwords (OTPs) for user authentication.
- **Data Preprocessor**: Prepares data for analysis by cleaning and transforming it.
- **Database**: Stores and retrieves user data, historical data, and other relevant information.
- **ML Model**: The machine learning model that performs disease predictions based on input data.

- **Report Generator**: Generates detailed reports based on the prediction results.
- **External Data Source**: Provides additional data required for more accurate predictions.

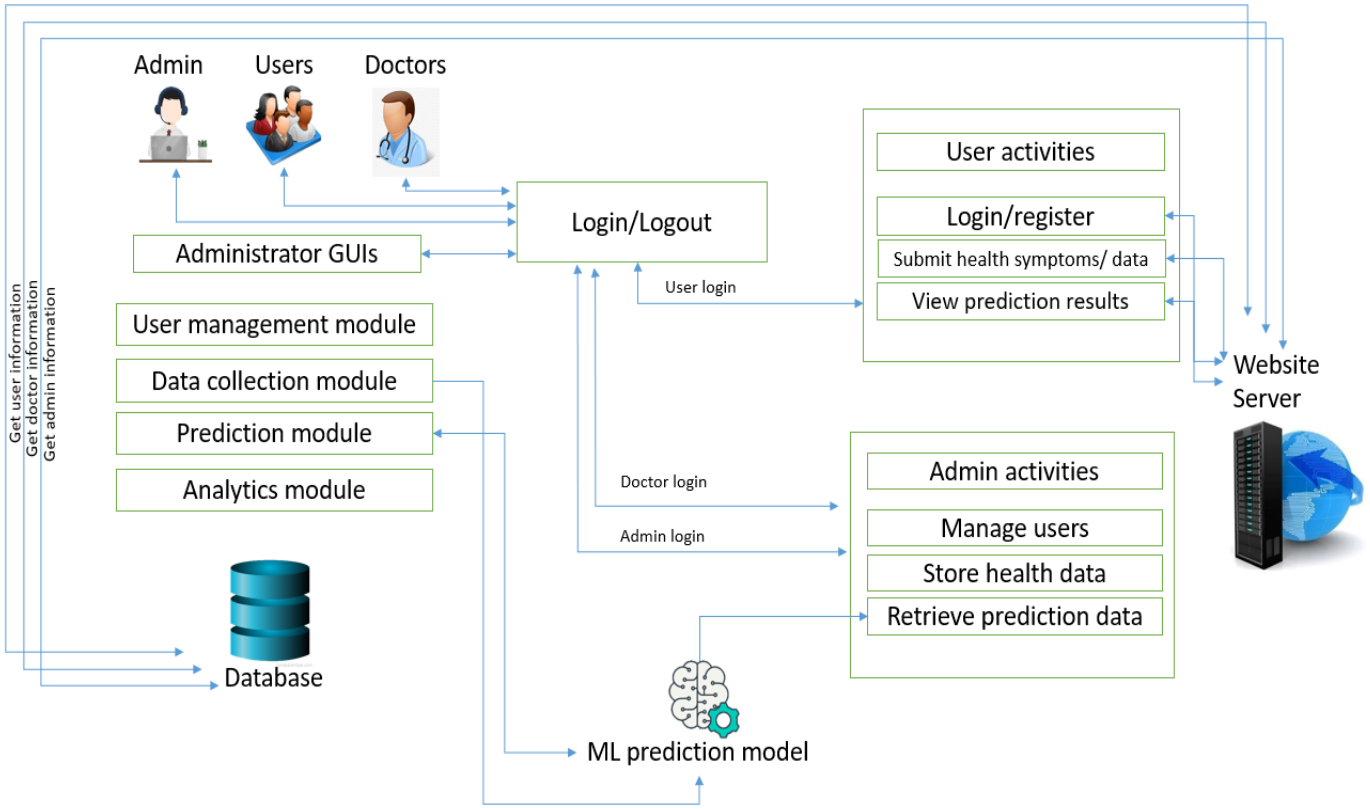
Interaction Steps:

1. **Request Login**:
    - The **User** requests to log in.
    - The **Disease Prediction System** initiates email validation through the **Input Validator**.
2. **OTP Verification**:
    - The **OTP Service** sends an OTP to the user's email.
    - The **User** submits the OTP.
    - The **OTP Service** verifies the OTP. If valid, the process continues.
3. **Successful Login**:
    - If the login is successful, the **User** submits symptoms and medical history.
    - The **Input Validator** checks the validity of the input.
4. **Input Validation**:
    - If the input is invalid, the system prompts the user for correction.
    - The user resubmits the corrected input.
    - Once the input is valid, the system proceeds to fetch historical data.
5. **Data Fetching and Preprocessing**:
    - The **Disease Prediction System** fetches historical data from the **Database**.
    - The **Data Preprocessor** processes this data.
    - The preprocessed data is sent to the **ML Model** for prediction.
6. **Prediction and Additional Data**:
    - The **ML Model** returns prediction results.
    - If additional data is needed, the system fetches it from the **External Data Source**.
    - The **Report Generator** creates a detailed report based on the results and any additional data.
7. **Result Display**:
    - The system displays the prediction results to the user.
    - If necessary, it prompts the user for any corrections.
8. **Feedback and Logout**:

- The **User** provides feedback.
- The **Disease Prediction System** stores this feedback.
- The user requests to log out.
- The system processes the logout request and logs out the user.

CHAPTER 4

BUSINESS ARCHITECTURE



The business architecture consists of interconnected components designed to facilitate the operation of a disease prediction system:

Login/Sign Up: This component enables users to create accounts and authenticate themselves within the system. Users provide credentials (such as email and password) during sign-up and use them to log in subsequently.

Administrator GUI: The administrator graphical user interface (GUI) provides tools and functionalities for system administrators to manage user accounts, configure system settings, and monitor system performance. Admin activities may include user management, data management, Result Prediction and system configuration.

User Activities: This component encompasses functionalities available to regular users within the system. Users can input their health data, view prediction results, receive personalized recommendations, and track their health progress over time.

Admin Activities: Admin activities involve functionalities available to system administrators. Admins can manage user accounts, monitor system performance, store health data, configure system settings, and access administrative reports and analytics.

ML Prediction Model: The machine learning (ML) prediction model is the core component responsible for analyzing user input data and generating disease risk predictions. The model leverages historical data and algorithms to predict disease likelihood based on user profiles and input data.

Website Server: The website server hosts the system's web application, providing access to users via web browsers. It serves web pages, handles user requests, and communicates with the backend components to retrieve and process data.

Database: The database component stores and manages various types of data used by the system, including user accounts, health data, prediction results, and system configuration settings. It ensures data integrity, reliability, and accessibility for the system's functionalities.

4.1 CURRENT PROCESS

The current process for disease prediction may vary depending on whether it's manual or automatic.

Manual Process: In a manual process, individuals may visit healthcare providers for regular check-ups, where their medical history, lifestyle habits, and family history of diseases are recorded.

Healthcare professionals may use standardized risk assessment tools and guidelines to estimate the risk of developing diseases like diabetes. However, this process heavily relies on subjective interpretation and may lack real-time data analysis capabilities.

Automatic Process: In an automatic process, if any, individuals may use digital health platforms or mobile applications to input their health data, such as symptoms, medical history, and lifestyle factors.

These platforms may employ algorithms or predictive models to analyze the data and generate personalized disease risk assessments

4.2 DIFFERENT PERSONAS AND THEIR USAGE

Individuals/Patients:

- Individuals typically engage in the process by providing personal health data through various channels, such as mobile apps or web platforms.
- They rely on the system to provide accurate disease risk assessments based on their input data.
- Individuals may use the system to monitor their health status over time and receive personalized recommendations for preventive measures.

Healthcare Providers:

- Healthcare providers utilize the system to assess patients' disease risk based on their health records and input data.
- They may integrate the system into their clinical workflow to support decision-making and provide personalized care plans to patients.
- Healthcare providers may also use the system for population health management and disease surveillance purposes.

Researchers:

- Researchers leverage the system to analyze population-level health data trends, identify risk factors for diseases, and develop predictive models.
- They use the system to conduct epidemiological studies, evaluate the effectiveness of interventions, and contribute to evidence-based healthcare practices.

4.3 BUSINESS PROBLEMS

- **Limited Accessibility to Healthcare Services**: Many individuals, especially in underserved areas, may face challenges accessing healthcare services for regular check-ups and disease risk assessments.
- **Subjectivity and Inconsistency**: Manual disease prediction processes may suffer from subjectivity and inconsistency in risk assessment, leading to variability in outcomes and recommendations.
- **Data Silos and Fragmentation**: Health data may be fragmented across multiple sources, such as electronic health records, wearable devices, and patient-reported data, making it difficult to obtain a comprehensive view of an individual's health status.
- **Delayed Intervention and Diagnosis**: Without timely disease prediction and intervention, individuals may remain unaware of their disease risk until symptoms appear, leading to delayed diagnosis and treatment initiation.
- **Resource Constraints in Healthcare Systems**: Healthcare systems may face resource constraints, including limited healthcare personnel and infrastructure, impacting their capacity to provide preventive care and disease management services.

CHAPTER 5

TEST STRATEGY

1. User Story 1: User Registration

Happy Path:

- Verify that users can register with valid credentials (email, password).
- Ensure that upon successful registration, users receive a confirmation email.
- Confirm that users can log in with the registered credentials.

Error Scenarios:

- Attempt to register with an invalid email format.
- Try to register with a password that does not meet the minimum requirements (e.g., length, complexity).
- Verify that users receive an appropriate error message if registration fails due to network issues or server errors.

2. User Story 2: Data Input

Happy Path:

- Validate that users can input their health and lifestyle data accurately.
- Confirm that all required fields (e.g., age, weight, family medical history) are properly validated and saved.

Error Scenarios:

- Attempt to input invalid data (e.g., non-numeric age, negative weight).
- Verify that users receive error messages for missing or invalid input fields.
- Test data input under poor network conditions to ensure data integrity and error handling.

3. User Story 3: Prediction Generation

Happy Path:

- Ensure that users can request a diabetes risk prediction based on their input data.

- Validate that the prediction is generated accurately and promptly.
- Confirm that users receive a clear explanation of their risk level and any recommendations for further action.

Error Scenarios:

- Test prediction generation with incomplete or inconsistent input data.
- Verify that users receive an appropriate error message if prediction generation fails due to technical issues.
- Test prediction generation under high server load to ensure scalability and performance.

4. User Story 4: Viewing Predictions

Happy Path:

- Confirm that users can view their previous diabetes risk predictions.
- Ensure that predictions are displayed accurately, including risk level and date/time stamp.

Error Scenarios:

- Test viewing predictions for a user with no previous prediction history.
- Verify that users receive an error message if there are issues retrieving prediction data from the server.

5. User Story 5: Account Management

Happy Path:

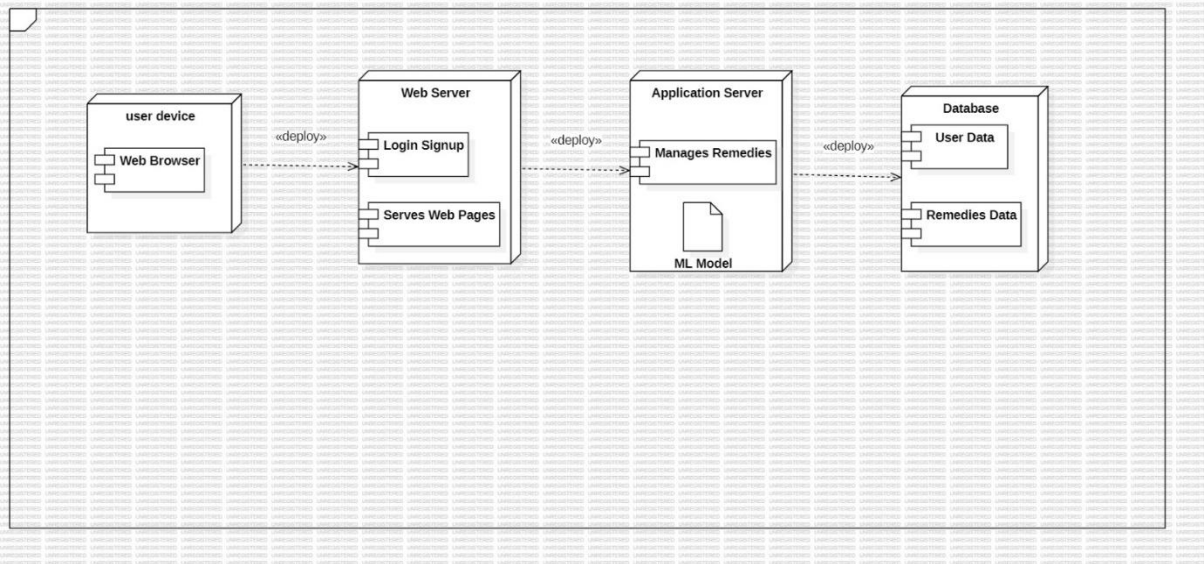
- Validate that users can update their account information (e.g., password, email).
- Confirm that users can delete their account if needed.

Error Scenarios:

- Attempt to update account information with invalid or unauthorized credentials.
- Verify that users receive confirmation prompts and warnings before deleting their account.
- Test account management functionalities under various edge cases (e.g., concurrent updates, network interruptions).

CHAPTER 6

DEPLOYMENT ARCHITECTURE



User Device (Web Browser):

- **Function**: This is the front-end interface through which users interact with the system. It allows users to register, log in, input data, request predictions, and view results.
- **Components**:
  - **Login/Signup Pages**: Web pages that facilitate user registration and authentication.
  - **Data Input Forms**: Interfaces for users to enter their health and lifestyle information.
  - **Prediction Display**: Sections to display diabetes risk predictions and recommendations.

Web Server:

- **Function**: This server handles all user requests related to accessing web pages, such as login, signup, and data input forms. It serves as the intermediary between the user device and the application server.
- **Components**:
  - **Authentication Module**: Manages user login and signup processes.
  - **Session Management**: Maintains user sessions for secure and seamless navigation.
  - **Request Handling**: Routes user requests to the appropriate components.

Application Server:

- **Function**: The core logic of the system resides here. It manages user data, processes input information, and interacts with the machine learning model to generate predictions. It also handles the application logic for managing remedies and recommendations.
- **Components**:
    - **Data Processing Module**: Processes and validates input data from users.
    - **Prediction Engine**: Interacts with the machine learning model to generate diabetes risk predictions.
    - **Recommendation System**: Provides personalized health recommendations based on prediction results.
    - **Business Logic**: Manages the flow of data and ensures that the correct operations are performed in response to user actions.

Database:

- **Function**: Stores all necessary data for the system, including user information, health data inputs, prediction results, and remedy recommendations.
- **Components**:
    - **User Data Storage**: Securely stores user credentials, personal information, and input data.
    - **Prediction Data Storage**: Maintains records of all predictions made, including input data, results, and timestamps.
    - **Remedies Data Storage**: Stores information about possible remedies and recommendations that can be provided to users based on their predictions.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

Conclusion

In conclusion, the Diabetes Prediction System presents a proactive approach to healthcare by leveraging machine learning to provide personalized risk assessments for diabetes. Throughout this project, we have developed and deployed a robust system capable of accurately forecasting an individual's risk of developing diabetes based on various health and lifestyle factors. By empowering individuals with early insights into their susceptibility to diabetes, the system encourages timely interventions and lifestyle adjustments, potentially averting the onset of diabetes-related complications.

Future Enhancements

While the current iteration of the Diabetes Prediction System offers valuable insights and functionality, there are several avenues for future enhancements and refinements:

Enhanced Prediction Models: Continuously refine and optimize the machine learning models used for diabetes prediction to improve accuracy and reliability.

Integration of Additional Data Sources: Explore the integration of additional data sources such as wearable devices, electronic health records, and genetic data to further enhance prediction capabilities.

User Feedback and Iterative Improvements: Gather user feedback and usage data to iteratively improve the user experience, feature set, and performance of the system.

Advanced Remedies and Recommendations: Expand the system to provide more advanced and personalized health recommendations, including diet plans, exercise routines, and medication suggestions.

Real-time Monitoring and Alerts: Implement real-time monitoring capabilities to track changes in user health metrics and provide timely alerts or interventions when necessary.

Integration with Healthcare Providers: Explore partnerships with healthcare providers to integrate the Diabetes Prediction System into clinical workflows and enable seamless communication between patients and healthcare professionals.