Hamdard University
Department of Computing
Final Year Project

GlucoPredict: AI-Powered Diabetes Predictor (FYP-017/FL24)

Software Design Specifications

Submitted by:
M. Moiez Siddiqui (2270-2021)
M. Hassan bin Sabih (2023-2021)

Supervisor:
Dr. Khurram Iqbal

Fall 2024

Document Sign off Sheet

Document Information

Project Title	GlucoPredict – AI Powered Diabetes Predictor
Project Code	FYP-017/FL24
Document Name	Software Design Specifications
Document Version	1.0
Document Identifier	FYP-017/FL24-SRS
Document Status	Draft
Author(s)	Hassan Moiez
Approver(s)	Dr. Khurram Iqbal
Issue Date	15/08/2024

Name	Role	Signature	Date
M. Moiez Siddiqui	Team Lead		
M. Hassan bin Sabih	Team Member 2		
Dr. Khurram Iqbal	Supervisor		
-	Co-Supervisor		
-	Project Coordinator		

Revision History

Date	Version	Description	Author
16/08/2024	1.0	Prepared Draft of SDS	Hassan
24/12/2024	1.1	Made changes to original draft	Hassan

Definition of Terms, Acronyms, and Abbreviations

Term	Description
Architectural Layer	Refers to the conceptual design that divides the system into layers such as presentation, business logic, and data.
Module	A self-contained unit of software designed to perform a specific functionality within the system.
Component	A part of the system that implements specific services or functionalities.
Interface	A boundary that defines how different components or systems communicate with each other.
Algorithm	A step-by-step procedure or formula for solving a problem, particularly those used for AI and prediction modeling.
Data Flow	The path data follows within the system, showing interactions between modules and components.
Use Case Scenario	A detailed narrative explaining how a user or system interacts with the software in a specific situation
System Dependency	Any external system or library that the software relies on for operation.
REST	Representational State Transfer, an architectural style for APIs.
API	Application Programming Interface

Table of Contents

1 Introduction

1.1 Purpose of Document

The purpose of this document is to provide a comprehensive overview of the software design for the GlucoPredict system. It details the architecture, modules, data flow, and algorithms used to implement the diabetes risk prediction system. This document serves as a blueprint for developers, testers, and stakeholders to ensure a shared understanding of the system's design and functionality.

1.2 Intended Audience

This document is intended for the following audiences:

- ***Development Team:*** *To guide the implementation process by outlining the system architecture and module design.*
- ***Quality Assurance Team:*** *To validate the system design against requirements and plan appropriate test cases.*
- ***Project Managers:*** *To oversee development and ensure alignment with the project goals.*
- ***Stakeholders:*** *To gain insights into the technical design and implementation of the system.*
- ***Researchers and Data Scientists:*** *To understand the integration of machine learning models and data workflows.*

1.3 Document Convention

The following conventions are used throughout this document:

- ***Font Styles:***
  - *Italics: Used for emphasis and definitions.*
  - *Bold: Used for key headings and terms.*
  - ***`Code Blocks:`*** *Represent code snippets or system commands.*
- ***Diagrams:*** *Unified Modeling Language (UML) diagrams are used to illustrate architecture, data flow, and sequence designs.*
- ***References:*** *External resources and documentation are cited using [bracketed hyperlinks] or footnotes where applicable*

1.4 Project Overview

GlucoPredict is an AI-powered system designed to predict diabetes risk in individuals by analyzing user-provided health data, such as glucose levels, age, weight, and lifestyle factors. The project integrates machine learning models, a user-friendly interface, and a secure backend to provide reliable predictions. Key features include:

- Personalized risk assessment reports.
- Visualizations of historical and predicted trends.
- Data integration from wearables and electronic health records (EHR).
- Real-time insights for healthcare providers.
  The system is aimed at empowering users and healthcare providers to take proactive measures for diabetes prevention and management.

1.5 Scope

In Scope:

- **Prediction of Diabetes Risk**: Implementing AI-driven models to predict an individual's diabetes risk based on provided data.
- **Data Collection and Integration**: Handling data from user inputs, wearables, and EHR systems.
- **User Interface Design**: Providing a user-friendly interface for at-risk individuals and healthcare providers.
- **Reporting and Visualization**: Generating and presenting insights in easy-to-understand formats, including graphs and reports.
- **Secure Data Management**: Ensuring data security, encryption, and compliance with healthcare data regulations (e.g., HIPAA).
- **APIs for External Integration**: Developing APIs for third-party integration, such as with healthcare provider systems or wearable devices.

Not In Scope:

- **Medical Diagnosis**: The system will not provide formal medical diagnoses or treatment recommendations.
- **Data Collection Hardware**: The project does not include the development or provisioning of wearable devices or medical hardware.
- **Manual Data Analysis**: The system relies entirely on automated machine learning algorithms and does not support manual data analysis by users.
- **Support for Rare Diseases:** Predictions will be focused solely on diabetes and related risks, excluding other medical conditions.

2 Design Considerations

By addressing important issues that need to be fixed to guarantee a successful implementation, this section lays the foundation for the GlucoPredict system design. These factors are unique to the design stage and don't restate topics covered in the SRS.

2.1 Assumptions and Dependencies

Assumptions:

- **Data Availability**: Users will provide accurate and sufficient health data (e.g., glucose levels, weight, age) for predictions.
- **System Access**: Users and healthcare providers will have access to stable internet connections to interact with the application.
- **Regulatory Compliance**: The system will operate within jurisdictions that allow the use of AI for health-related applications and comply with relevant data protection laws.
- **Hardware Requirements**: Users will utilize compatible devices, such as modern smartphones, tablets, or computers.
- **Integration Feasibility**: Wearables and EHR systems will support standard APIs or formats for seamless data integration.

Dependencies:

- **Third-Party APIs**: Dependence on external APIs for accessing wearable data and integrating with EHR systems.
- **Machine Learning Libraries**: Relying on libraries such as TensorFlow or PyTorch for implementing predictive models.
- **Database Management System**: Dependence on a relational or NoSQL database (e.g., PostgreSQL or MongoDB) for secure data storage.
- **Cloud Infrastructure**: Using cloud services like AWS or Azure for hosting and scaling the application.
- **Regulatory Standards**: Compliance with healthcare regulations (e.g., HIPAA, GDPR) that may influence system design and data handling.

2.2 Risks and Volatile Areas

Risks:

- **Data Privacy Breaches**: Unauthorized access or misuse of sensitive health data could harm user trust and violate regulations.
- **Machine Learning Bias**: The predictive model may produce biased results if the training data is unbalanced or insufficient.
- **Integration Challenges**: Difficulties in integrating with third-party APIs or wearable devices due to non-standardized formats.
- **System Downtime**: Risks of service unavailability due to cloud infrastructure failures or maintenance issues.
- **User Non-Compliance**: Users may provide incomplete or incorrect data, impacting prediction accuracy.

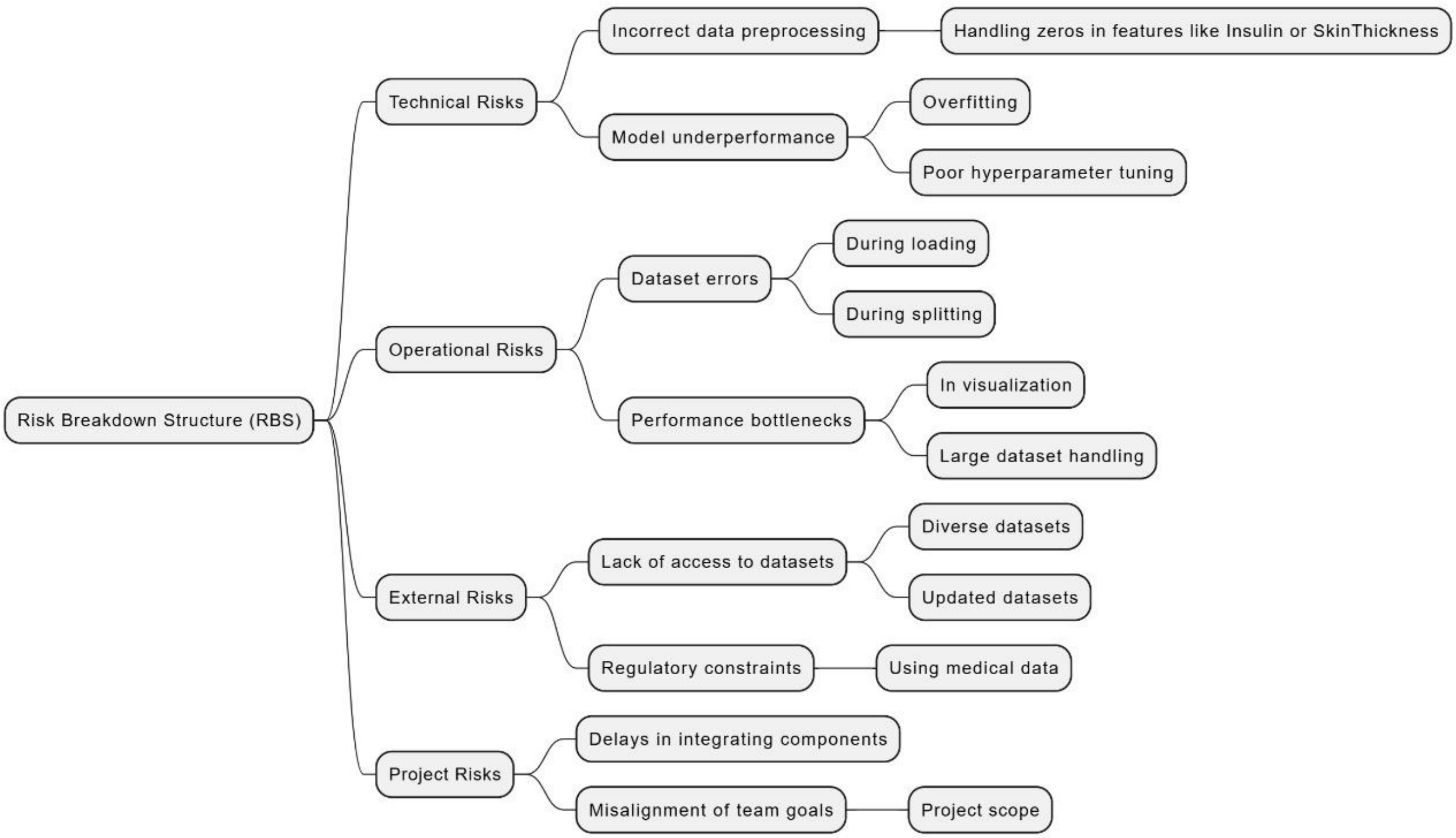
Volatile Areas:

- **Regulatory Changes**: Updates to healthcare regulations (e.g., stricter data handling laws) could require significant system modifications.
- **Technology Evolution**: Rapid advancements in AI or cloud computing may necessitate updates to maintain system efficiency.
- **API Dependencies**: External API changes or deprecations could disrupt functionality and require rework.

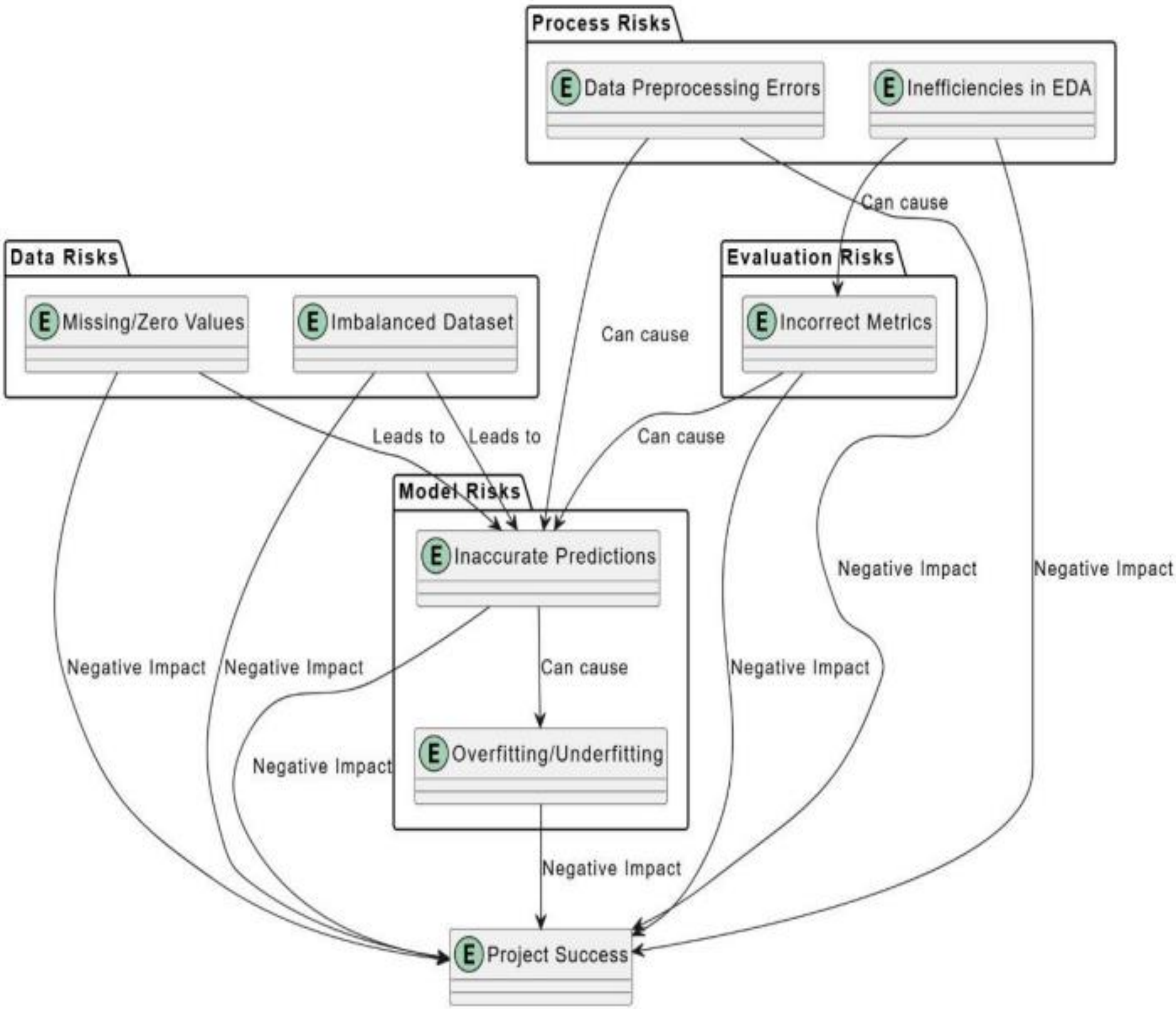
- **Market Dynamics**: Emergence of competitive solutions or changes in user expectations could impact project relevance.
- **Data Security Protocols**: Increasing cybersecurity threats may demand continual updates to safeguard user data.

Risk Breakdown Structure (RBS):

Technical Risks

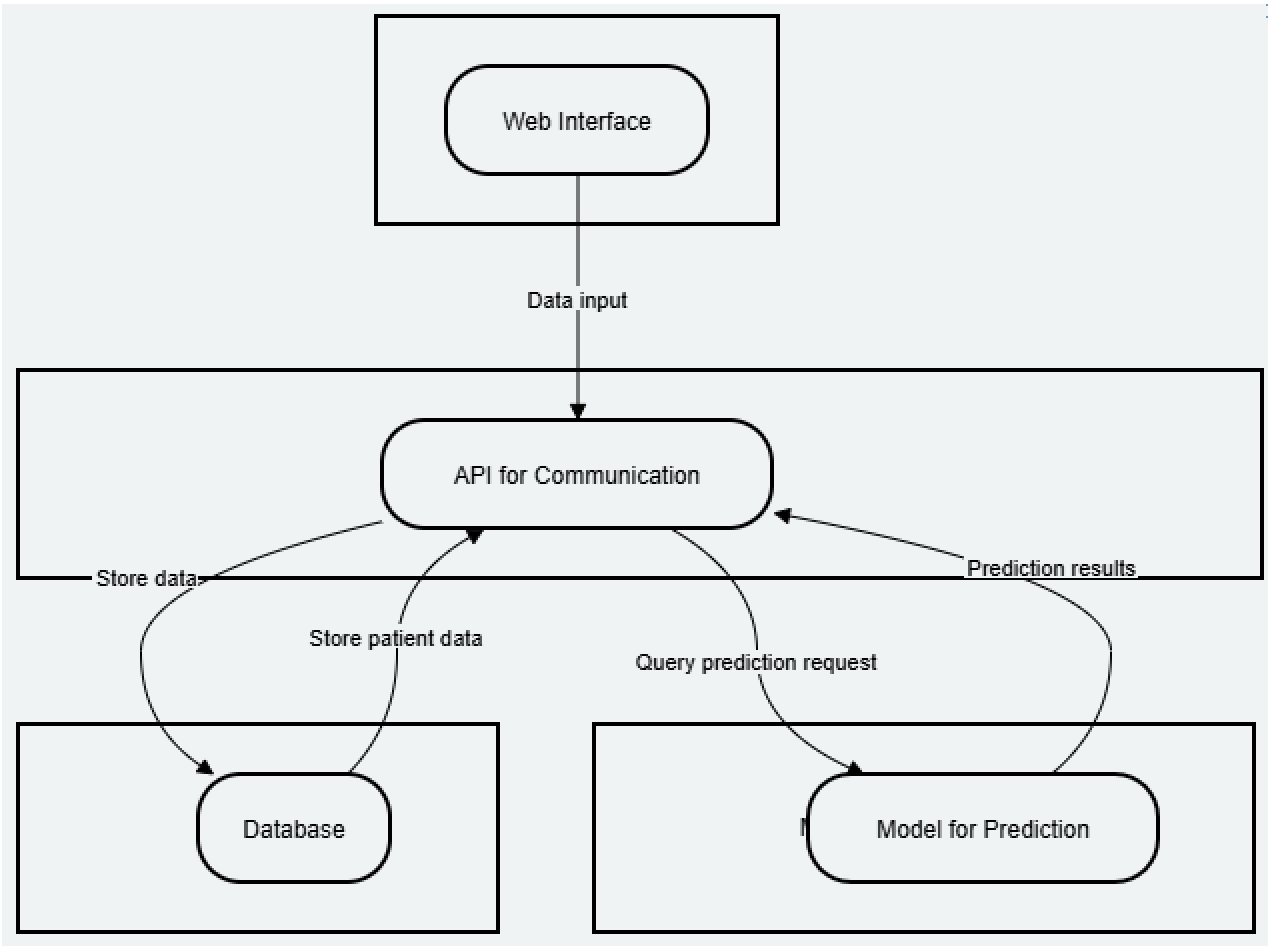


Operational Risks



3 System Architecture

The GlucoPredict system architecture is designed using a modular and layered approach to enhance scalability, maintainability, and ease of integration. The system is decomposed into components that interact through well-defined interfaces to deliver the desired functionalities. Below is a high-level overview of the system's architecture:



3.1 System Level Architecture

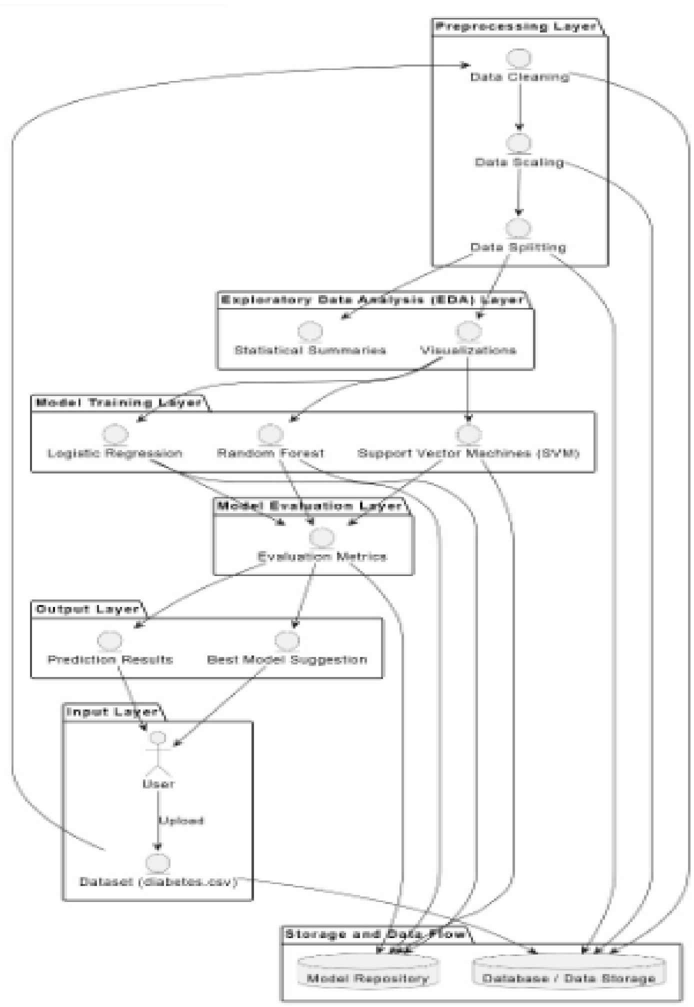
The GlucoPredict system-level architecture is made to be modular, scalable, and maintainable. Each of the discrete subsystems that make up the system has well defined functions and interfaces. An outline of the top-level breakdown and the connections between the subsystems may be found below.

System Decomposition

The GlucoPredict system is divided into the following subsystems:

1. **User Interface Subsystem**: Manages interaction with end-users, providing input forms, dashboards, and visualizations.
2. **Machine Learning Subsystem**: Responsible for diabetes risk prediction using trained models.
3. **Data Management Subsystem**: Handles data storage, retrieval, and integration from various sources (e.g., wearables, EHR).
4. **API Gateway Subsystem**: Facilitates communication between internal and external systems using RESTful APIs.

Package Diagram



Interfaces to External Systems

- **Wearables and Health Devices**: Interfaces via standardized APIs (e.g., Fitbit, Apple Health).
- **Electronic Health Records (EHR)**: Integration with EHR systems using HL7 or FHIR standards.
- **Cloud Services**: Utilizes AWS or Azure for hosting, storage, and ML model deployment.
- **Third-Party APIs**: Dependency on external APIs for auxiliary functionalities like geolocation or notification services.

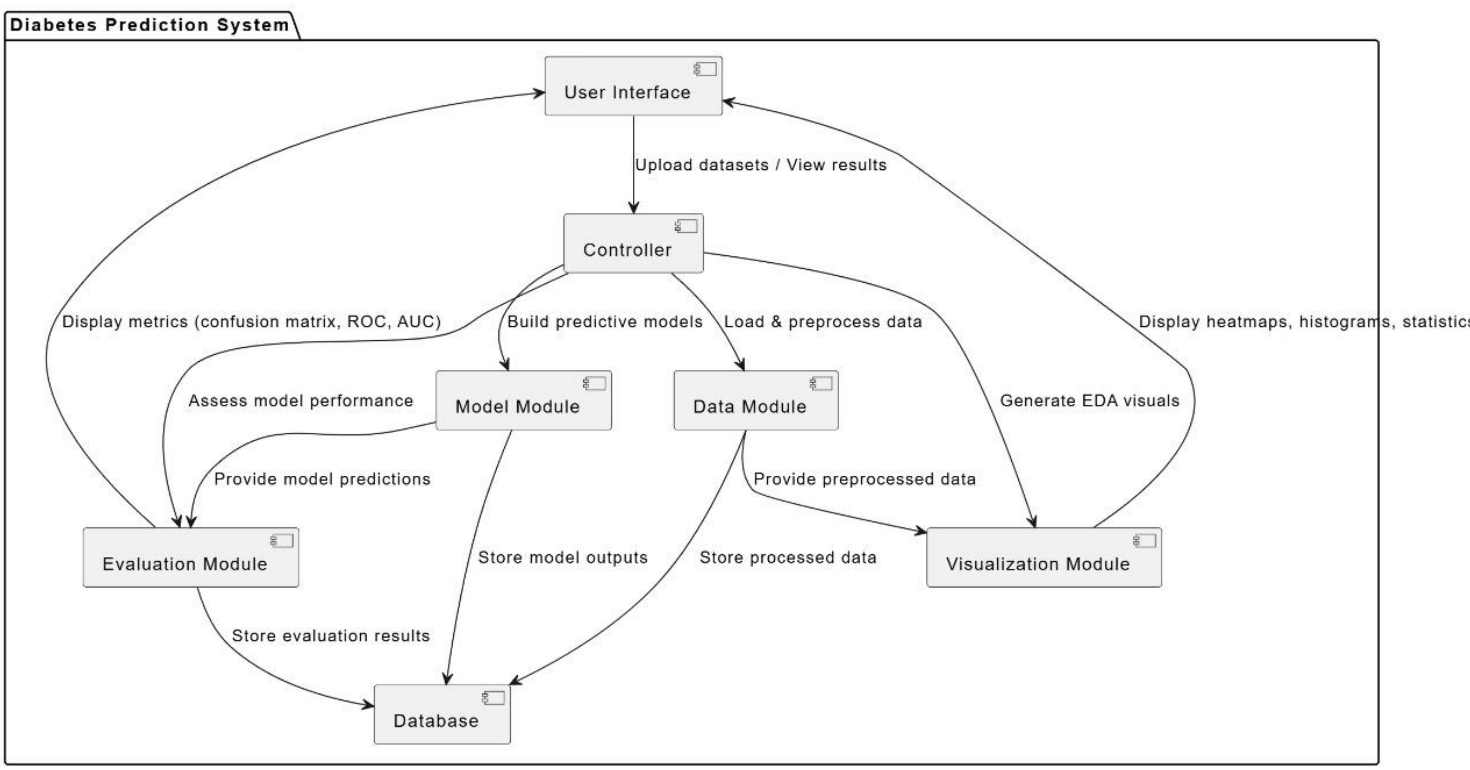
Major Physical Design Issues

- **Scalability**: Ensuring the system can handle increased users and data volume.
- **Latency**: Minimizing response time, especially for real-time predictions.
- **Fault Tolerance**: Designing for high availability, with redundancy in cloud infrastructure.
- **Data Security**: Implementing robust encryption and access control mechanisms.

Global Design Strategies

- **Error Handling**: Implementing robust mechanisms for detecting, logging, and gracefully recovering from errors to maintain system stability.
- **Scalability**: Designing the system to handle growth in user base and data volume through load balancing, horizontal scaling, and distributed databases.
- **Security**: Ensuring data protection through encryption, secure authentication mechanisms, and compliance with healthcare regulations (e.g., HIPAA, GDPR).

Component Diagram



3.2 Software Architecture

User Interface Layer

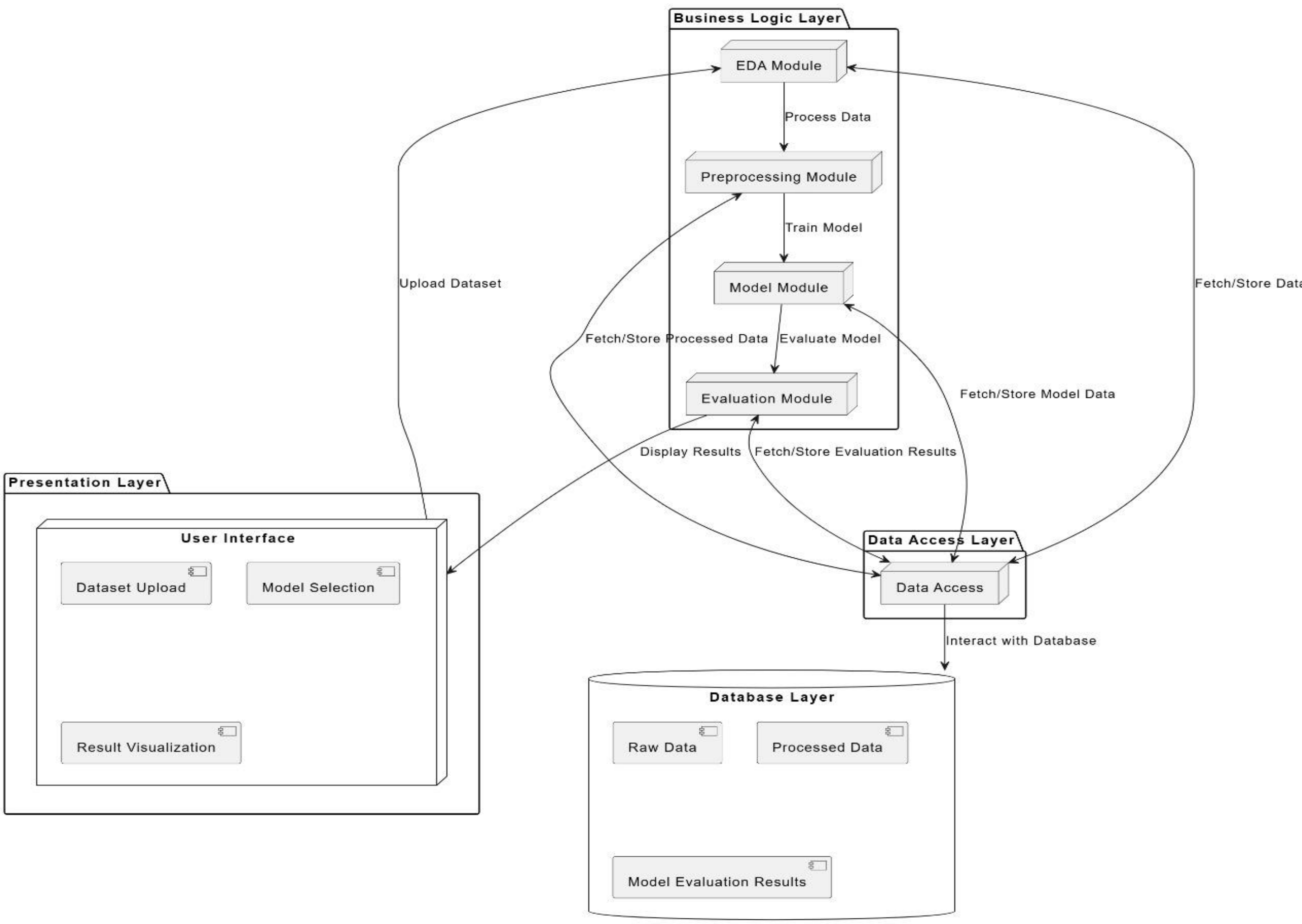
- Provides an intuitive and interactive platform for users.
- Key components:
  - **Frontend Framework**: React.js or Angular for web-based applications.
  - **Mobile Compatibility**: Responsive design or dedicated mobile apps (Flutter/React Native).
  - **Features**: Data input forms, real-time dashboards, and visualization tools for trends and predictions.

Middle Tier (Business Logic Layer)

- Orchestrates data processing, prediction, and interaction between the UI and data layers.
- Key components:
  - **API Gateway**: Handles communication between subsystems.
  - **Prediction Engine**: Implements ML models and provides diabetes risk scores.
  - **Validation Services**: Ensures data integrity and validation before processing.

Data Access Layer

- Manages secure access to the database and ensures data consistency.
- Key components:
  - **Database Management**: Use of RDBMS (PostgreSQL) or NoSQL (MongoDB) depending on data type.
  - **Data Integration**: ETL processes for wearable and EHR data synchronization.
  - **Data Security**: Encryption of sensitive user data at rest and in transit.



4 Design Strategy

GlucoPredict's design approach is centered on developing a modular, scalable, and maintainable system that supports the project's objectives of effectively generating diabetes prediction. To guarantee seamless operation in real-world situations, the approach places a strong emphasis on extensibility, reusability, user-friendly interaction, robust data management, and concurrency handling.

4.1 Future System Extension or Enhancement

- **Advanced Models**: Incorporate real-time glucose data from wearables and more sophisticated prediction models.
- **Healthcare Integration**: Enable sharing predictions with healthcare providers.
- **Mobile App**: Develop a mobile version for real-time tracking.
- **Global Expansion**: Include multilingual support and broaden to other health conditions.

4.2 System Reuse

- **Libraries:** Utilize existing frameworks like **TensorFlow**, **Keras**, and **scikit-learn**.
- **Cloud: Use** services like **AWS** or **Google Cloud** for data storage and scalability.
- ***APIs:*** Integrate with third-party systems (wearables, health apps).

4.3 User Interface Paradigms

- **Minimalistic & Responsive**: Simple, clear design adaptable for both desktop and mobile.
- **Personalized Dashboards**: Display glucose trends and health insights.
- **Accessibility**: Compliant with WCAG for visually impaired users.

4.4 Data Management (Storage, Distribution, Persistence)

- **Storage**: Use **cloud-based databases** (e.g., **AWS RDS**, **MongoDB**).
- **Persistence**: Ensure secure data retention with regular backups.
- **Distribution**: Microservices architecture for scalability and flexibility.
- **Security**: Implement **end-to-end encryption** and comply with **HIPAA/GDPR**.

4.5 Concurrency and Synchronization

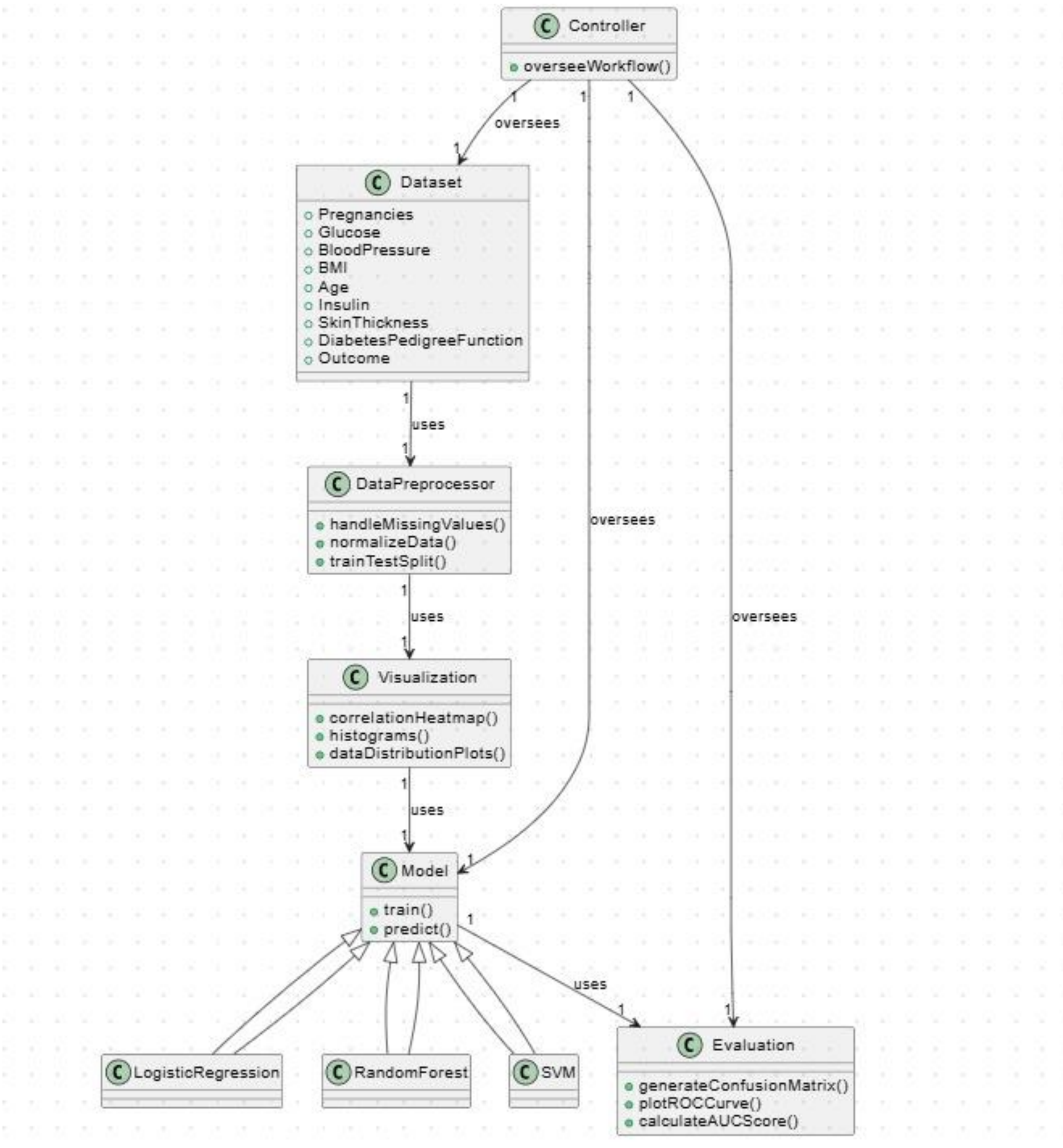
- **Concurrency**: The system will use **asynchronous processing** and **multi-threading** for tasks such as data collection, risk prediction, and report generation. This ensures that the system can handle concurrent requests without blocking or slowing down user interactions.
- **Synchronization**: For data integrity and consistency, the system will utilize **database transactions** to handle updates to user data, ensuring atomicity and preventing race conditions. In case of concurrent data access, **locks** or **optimistic concurrency control** will be employed to ensure consistent results.

4.6 Design Trade-offs

- **Accuracy vs. Performance**: More complex machine learning models may provide higher prediction accuracy but require more computational resources. A balance will need to be struck to ensure predictions are accurate while maintaining reasonable response times, especially for mobile users.
- **Development Time vs. Feature Set**: The initial release of **GlucoPredict** will prioritize core features (glucose risk prediction and user profiles) to ensure a quick market launch. More advanced features, such as personalized health recommendations or AI-powered dietary suggestions, will be introduced in future versions.
- **Cost vs. Scalability**: Implementing a cloud-based infrastructure for scalability will incur additional operational costs. However, the scalability and flexibility benefits will be essential for accommodating future growth and ensuring high availability.

5 Detailed System Design

5.1 Design Class Diagram



Logical Data Model (E/R Model)

- **Entities**:
- **Raw Data**: Represents user inputs such as health metrics (e.g., glucose levels, age, BMI).
- **Features**: Derived or selected attributes like age, glucose, BMI, or lifestyle factors.
- **Model**: Represents machine learning models like XGBoost and Logistic Regression.
- **Prediction**: The result of applying the model, such as diabetes risk prediction.
- **Metrics**: Evaluation metrics for model performance, e.g., accuracy, precision.

- **Relationships**:
- Raw Data is **processed into** Features (via cleaning, extraction, and scaling).
- Features are **used by** Models for training and evaluation.
- Models **generate** Predictions.
- Metrics are **calculated** for Models based on their Predictions.

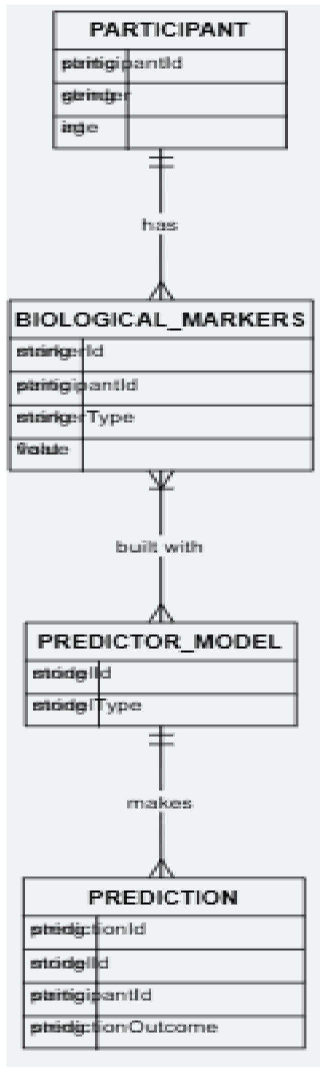
Detailed GUIs

- **Data Input Screen**:
  - A form where users input raw data such as glucose levels, BMI, and activity levels.
  - Option to upload medical records or wearable device data.
- **Data Cleaning and Feature Display**:
  - Display raw data and allow users to verify or adjust inputs before processing.
  - Show selected features after cleaning and scaling.
- **Model Selection and Results**:
  - A panel for choosing the prediction model (e.g., XGBoost or Logistic Regression).
  - Display metrics such as accuracy or error rates in a dashboard.
- **Prediction Results Screen**:
  - Display the predicted diabetes risk in a simple format (e.g., “High Risk” or “Low Risk”).
  - Include actionable recommendations for users (e.g., consult a doctor, lifestyle changes).
- **Evaluation Metrics Dashboard**:
  - A visualization of model performance (charts or graphs for accuracy, precision, etc.).
  - Option to compare models side-by-side.

5.2 Database Design

5.2.1 ER Diagram

Below is the Entity Relationship (ER) diagram for the system. It represents the relationships between entities such as Participants, Biological Markers, Predictor Model and Prediction. Each entity is described with its attributes and relationships to other entities.



5.2.2 Data Dictionary

A thorough explanation of data components, including their names, aliases, uses, and structures, is given in this section. Every data element is documented using the template that is supplied.

5.2.2.1 Data 1

Raw Data						
Name	Raw Data					
Alias	Input Data					
Where-used/how-used	• Input to: Data Cleaning, Feature Extraction • Stored in: Database for historical data tracking					
Content description	Raw Data = User Demographics + Health Metrics					
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
User ID	Unique identifier for user	INT	10	No	Auto increment	PK
Age	Age of the user	INT	3	No	NULL	
Glucose Level	Fasting or postprandial glucose levels	FLOAT	6	No	NULL	
BMI	Body Mass Index of the user	FLOAT	5	No	NULL	
Activity Level	Physical activity level (low/medium/high)	VARCHAR	10	Yes	“low”	

5.2.2.2 Data 2

Features						
Name	Features					
Alias	Processed Attributes					
Where-used/how-used	• Input to: Model Training, Prediction Engine • Output from: Data Cleaning, Feature Selection					
Content description	Features= Selected Attributes [|] Scaled Data					
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
Feature ID	Unique identifier for the feature	INT	10	No	Auto-increment	PK
Feature Name	Name of the feature (e.g., glucose level)	VARCHAR	50	No	NULL	
Value	Processed value for the feature	FLOAT	10	No	NULL	
Scale Factor	Scaling factor applied	FLOAT	5	Yes	1.0	

5.2.2.3 Data 3

Model						
Name	Model					
Alias	Prediction Algorithm					
Where-used/how-used	• Input to: Evaluation • Output from: Training Processes					
Content description	Model = Features + Algorithm					
Column Name	**Description**	**Type**	**Length**	**Nullable**	**Default Value**	**Key Type**
Model ID	Unique Identifier for the ML model	INT	10	No	Auto-Increment	PK
Algorithm	Algorithm used (e.g., XGBoost, Logistic	VARCHAR	50	No	NULL	
Accuracy	Date of quiz creation	FLOAT	5	Yes	NULL	
Last Trained	Timestamp of the latest training	DATETIME	-	Yes	NULL	

5.2.2.4 Data 4

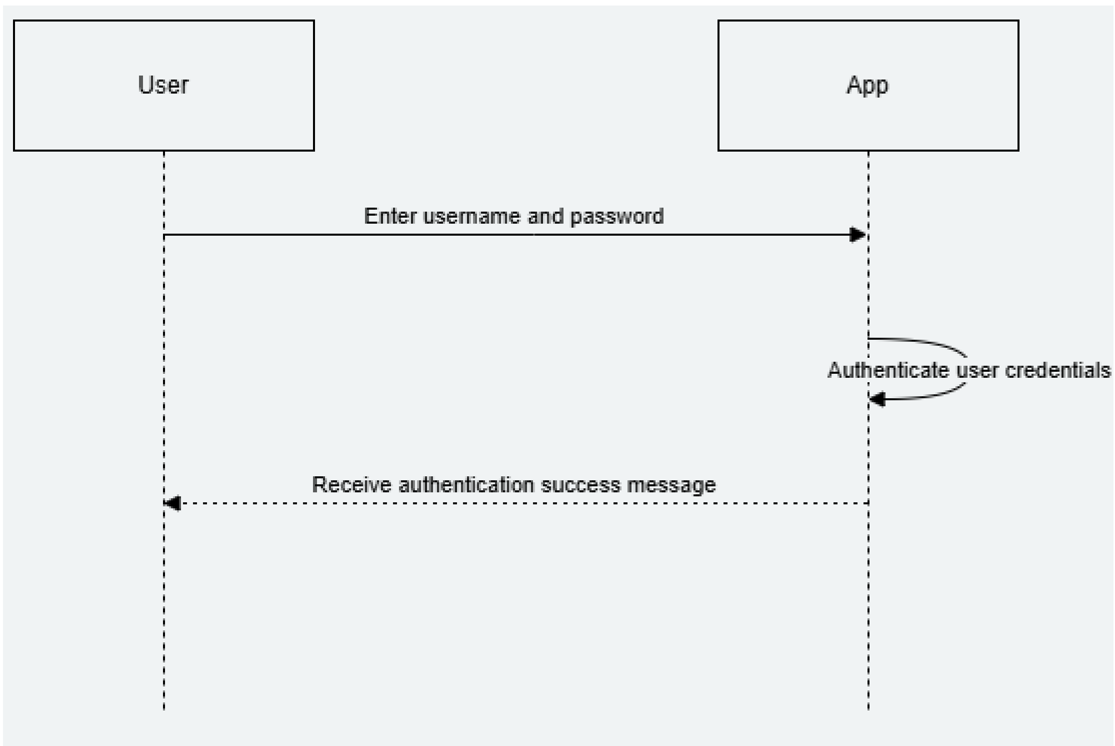
Prediction						
Name	Prediction					
Alias	Diabetes Risk Assessment					
Where-used/how-used	• Input to: Reporting, Feedback Mechanisms • Output from: Prediction Engine					
Content description	Prediction = Model Output					
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
Prediction ID	Unique identifier for the prediction	INT	10	No	Auto-increment	PK
User ID	Identifier for the user	INT	10	No	NULL	FK
Risk Level	Predicted diabetes risk (e.g., high/low)	VARCHAR	10	No	NULL	
Confidence Score	Confidence in the prediction (%)	FLOAT	5	Yes	NULL	
Timestamp	Prediction timestamp	DATETIME	-	No	Current time	

5.3 Application Design

The sequence diagrams that show how various system components interact are shown in this section. With a description of the interaction processes, each sequence graphic depicts a certain system functionality.

5.3.1 Sequence Diagram

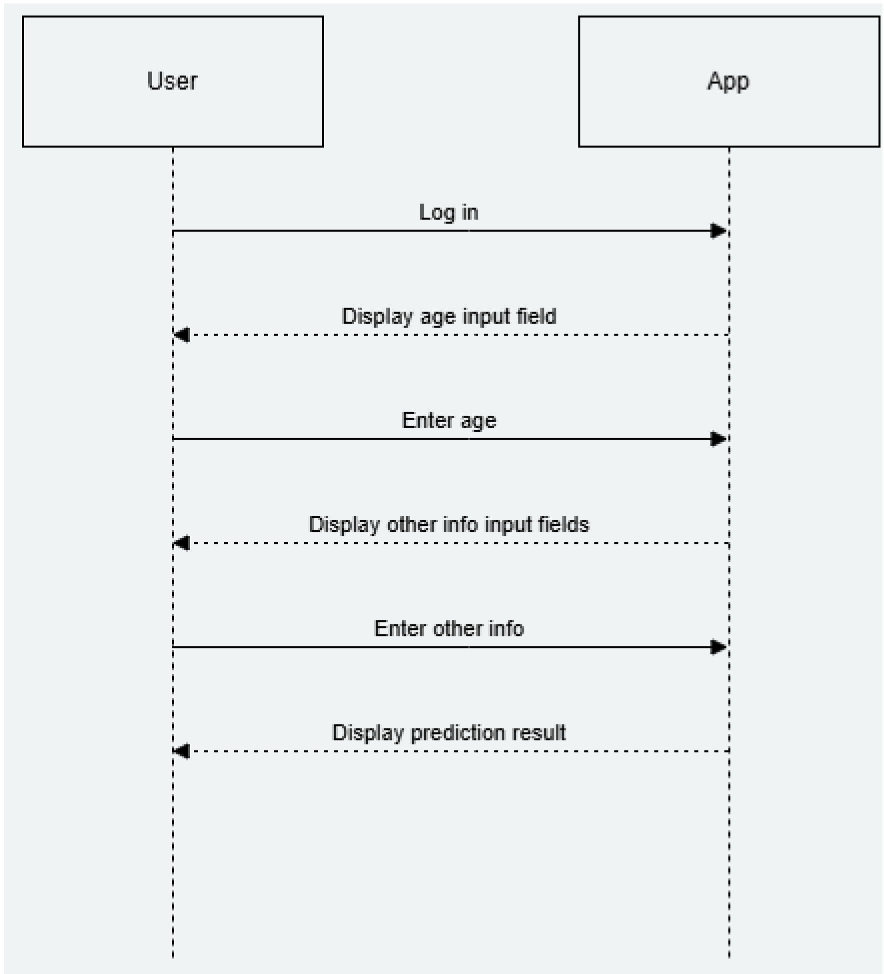
5.3.1.1 Sequence Diagram 1: User Login



Description

- Through the frontend, the user enters their login information.
- To verify credentials, the frontend requests information from the backend.
- The database is queried by the backend to obtain user data.
- The system either shows an error notice or logs the user in based on the outcome.

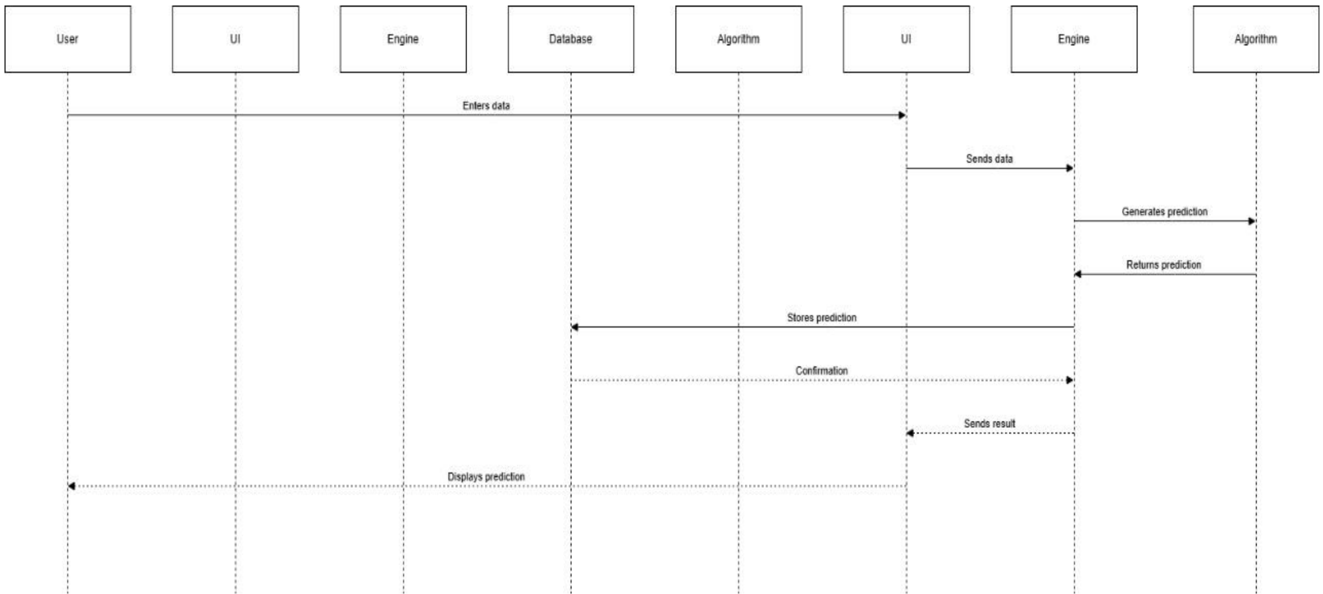
5.3.1.2 Sequence Diagram 2: Data Entering



Description

- The user after successfully entering in the app
- They would be required to enter their age
- They also need to enter other medical requirements
- The user is then notified with whether diabetes is detected or not

5.3.1.3 Sequence Diagram 3: Prediction



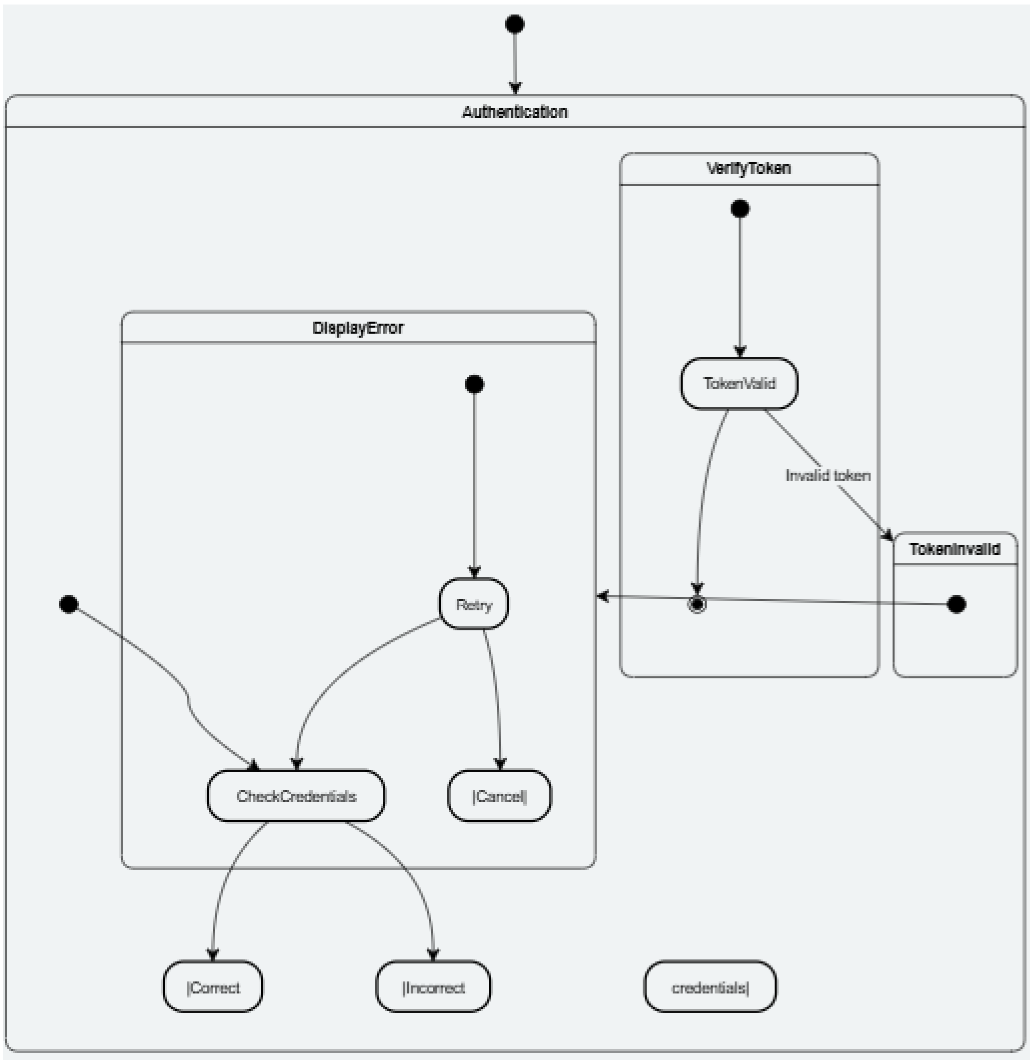
Description:

- After all the data are given as per the requirement
- The data is analyzed
- After the analysis the result is given to the user that whether it has detected diabetes or not

5.3.2 State Diagram

The different states that a system or component can occupy and the transitions between them depending on circumstances or events are shown in state diagrams. State diagrams for the various system components are shown below.

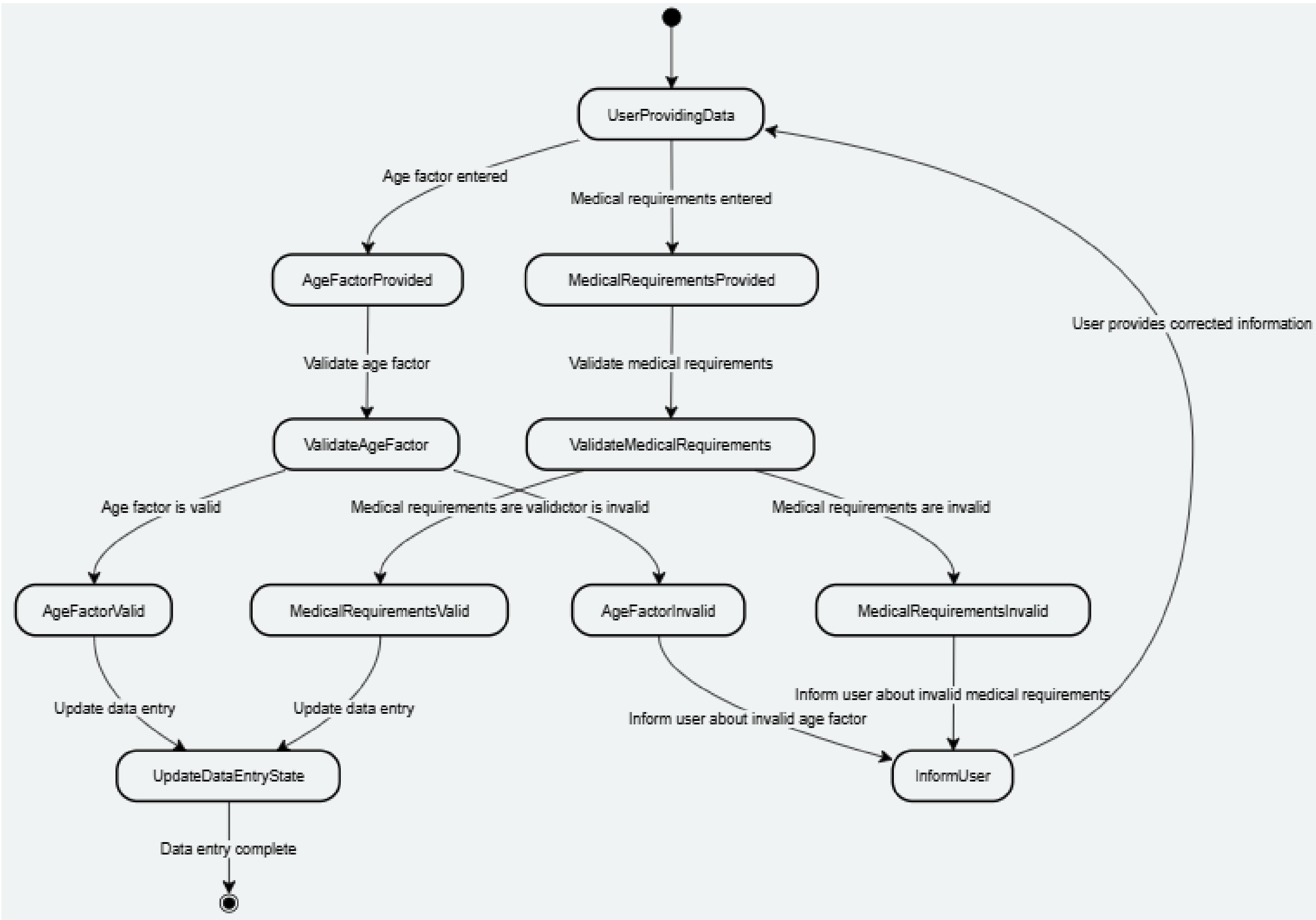
5.3.2.1 State Diagram 1: User Authentication



Description

- The initial state is the start of the app.
- The system transitions to the login screen where the user enters credentials.
- If valid credentials are provided, the system transitions to the logged-in state.
- If invalid credentials are provided, the system remains on the login screen.
- Logging out transitions the system back to the initial state.

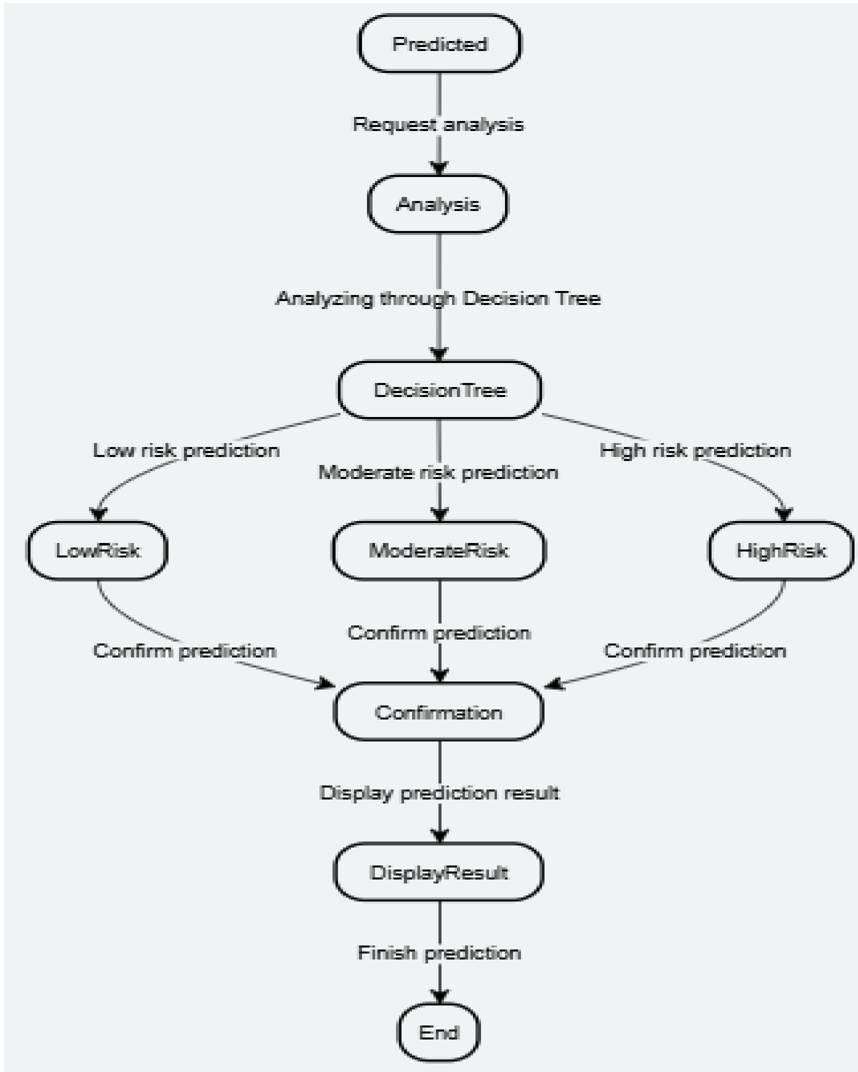
5.3.2.2 State Diagram 2: Data Entering Stage



Description

- The system begins in an idle state.
- The user uploads age and other medical requirements.
- Once uploaded, the data is analyzed by the Decision Tree.
- Errors prompt a retry or cancel option, bringing the system back to idle.

5.3.2.3 State Diagram 3: Prediction Process

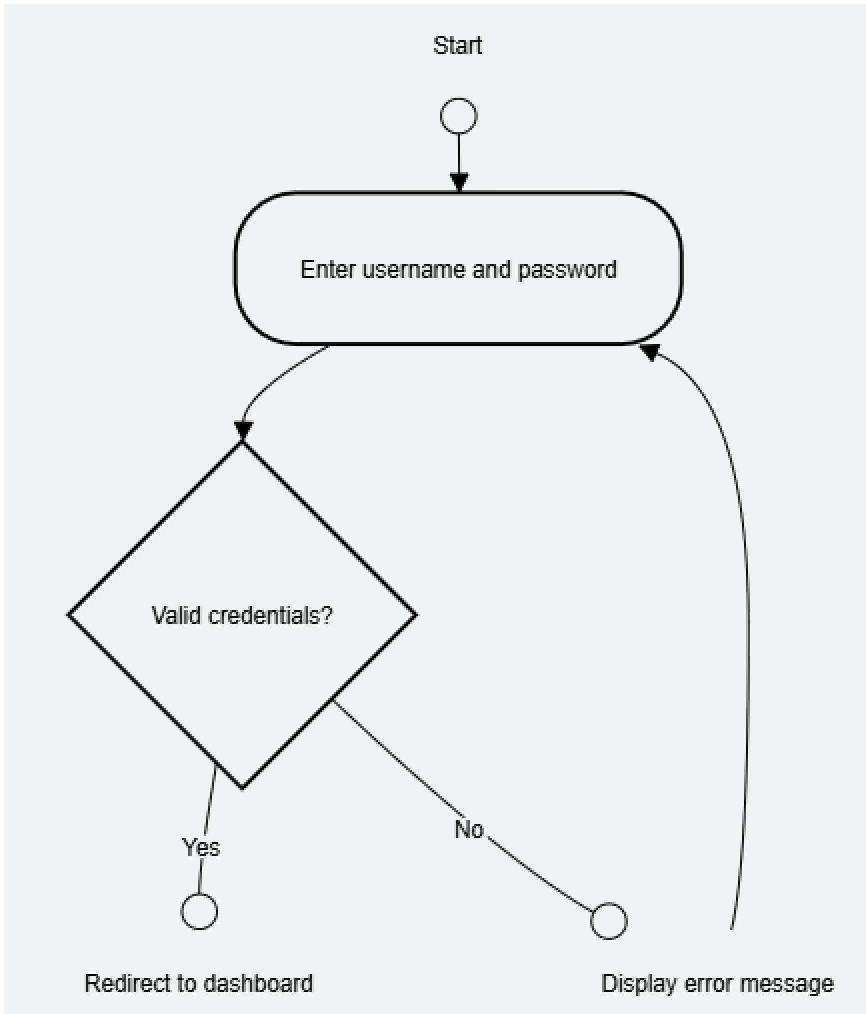


Description

- The system starts in an idle state.
- The user uploads the age and other medical factors.
- The system processes the input, leading to either a success or an error.
- On success, the system transitions back to idle after the user receives the diabetes prediction.
- On error, the user can retry or exit, returning to the idle state.

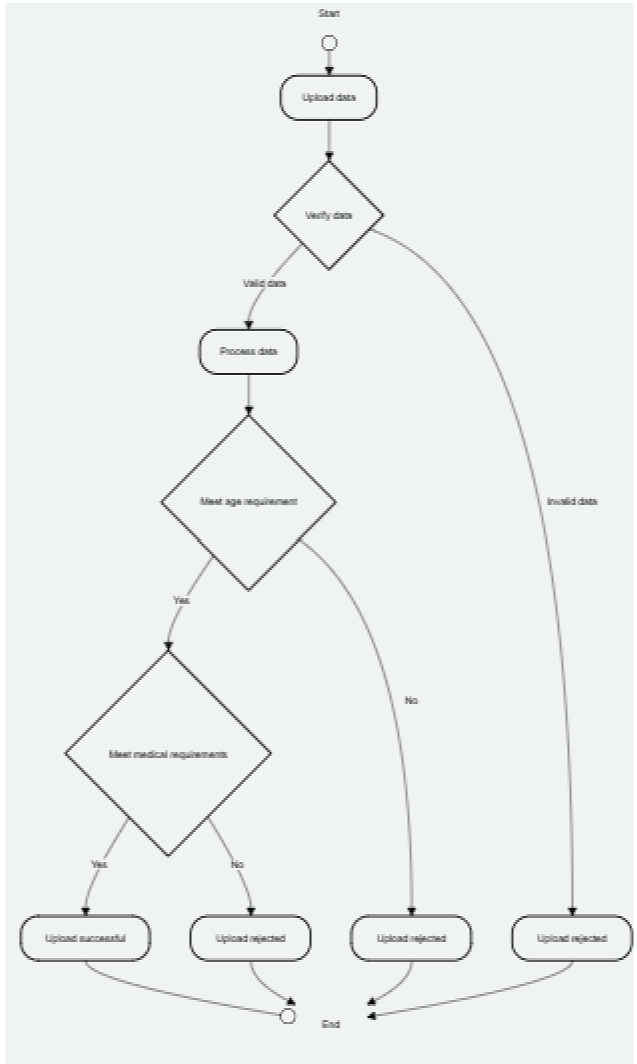
5.3.3 Activity Diagram
Activity diagrams display the system's activities or control flow. Activity diagrams for the various system components are shown below.

5.3.3.1 Activity Diagram 1: User Authentication



Description
- The user launches the application, which displays the login screen.
- If valid credentials are entered, the system authenticates the user and redirects them to the dashboard.
- If credentials are invalid, the system displays an error message and awaits user input.

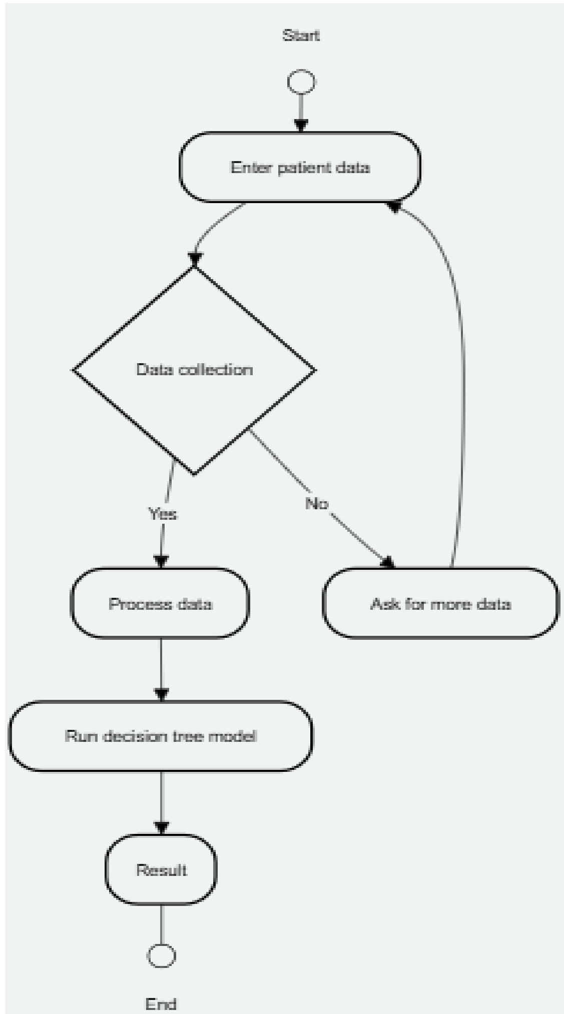
5.3.3.2 Activity Diagram 2: Data Upload Process



Description

- The system begins in an idle state.
- The user uploads age and other medical requirements.
- Once uploaded, the data is analyzed by the Decision Tree.
- Errors prompt a retry or cancel option, bringing the system back to idle.

5.3.3.3 Activity Diagram 3: Prediction Process



Description

- The system starts in an idle state.
- The user uploads the age and other medical factors.
- The system processes the input, leading to either a success or an error.
- On success, the system transitions back to idle after the user receives the diabetes prediction.
- On error, the user can retry or exit, returning to the idle state.

5.4 GUI Design

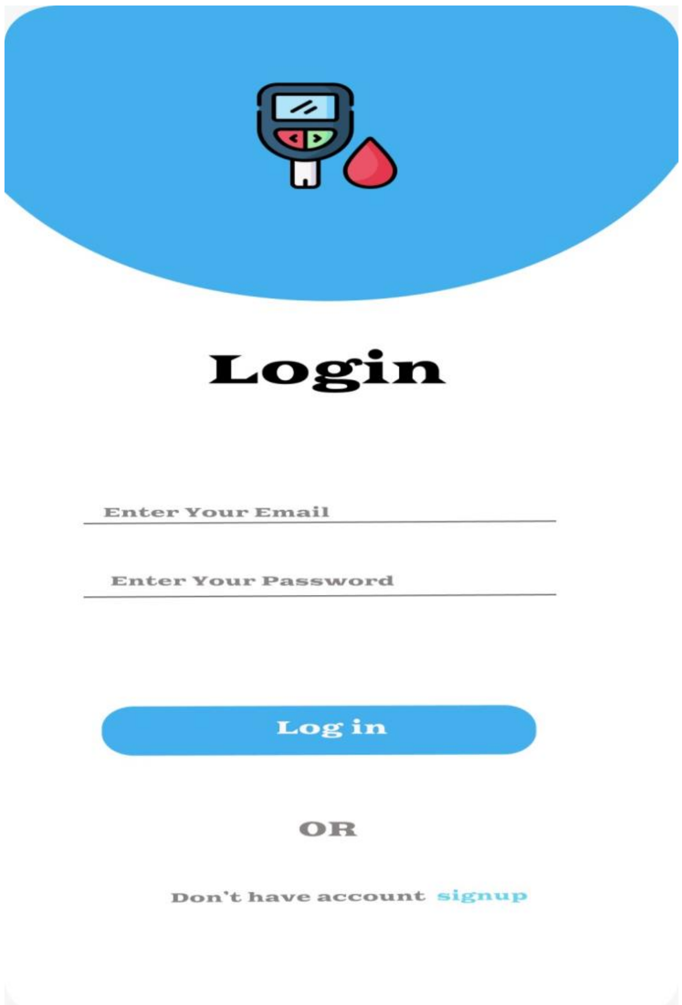
5.4.1 Home Page - Mock Screen 1

Purpose:
As the application's entry point, the Home Page introduces users to the system
Elements and Functions:
Contains the welcoming message for the Diabetes App with the option given to create account or to login in to the existing account.

5.4.2 Login Page - Mock Screen 2

Purpose:
Allows registered users to authenticate themselves and access the system's features.
Elements and Functions:

- Input field for the user's Email ID.
- Input field with masking for user passwords.
- Submits credentials for verification.

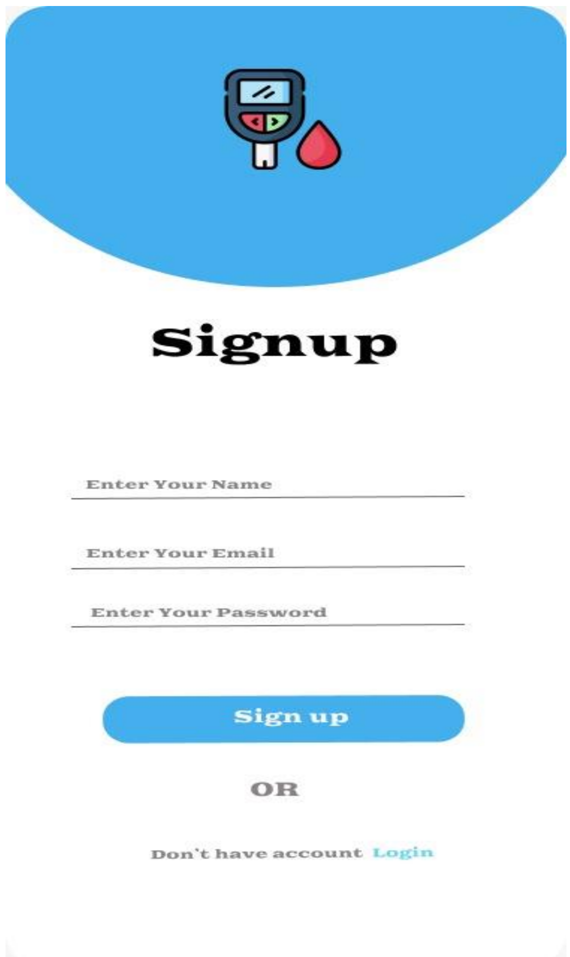


5.4.3 Sign Up Page - Mock Screen 3

Purpose:
Allows new users to create an account by providing their details.

Elements and Functions:

- Name
- Email Address
- Password
- Submits user details to create an account.
- Redirects existing users back to the Login Page.



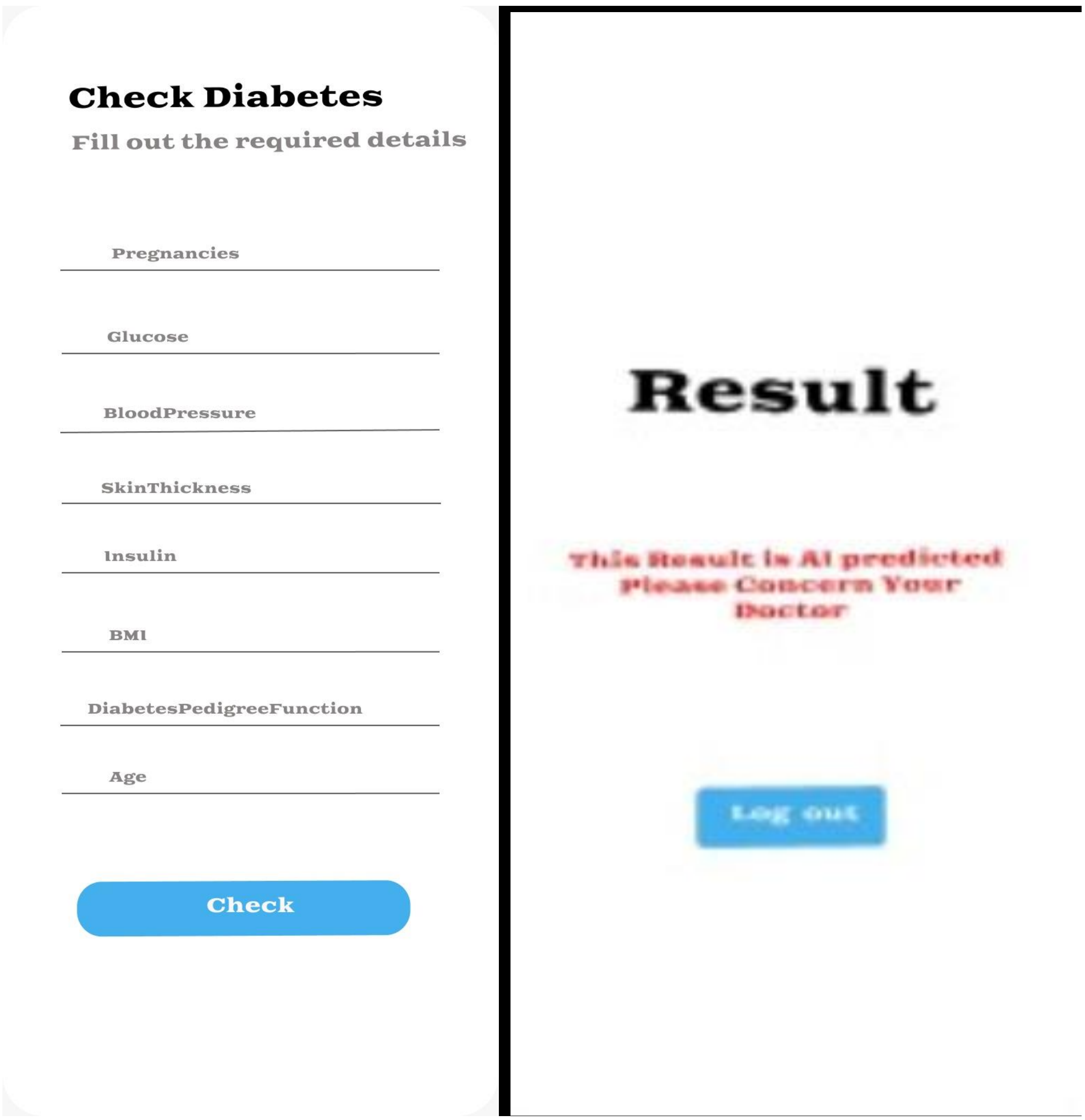
5.4.4 Data Collection Screen - Mock Screen 3

Purpose:

Gathers the information from the user to detect the diabetes.

Elements and Functions:

- The field are given for the user to enter the required information
- If any data isn't given to the AI it will not proceed further and display an error
- After the submission of the data, "Check" option is selected and the data is analyzed to give the prediction.
-



6 References

1. **IEEE Standards for Software Engineering**
    - **Title**: IEEE Standard for Information Technology - Systems Design - Software Design Descriptions
    - **Report Number**: IEEE Std 1016-2009
    - **Date**: 2009
    - **Publishing Organization**: IEEE
    - **Source**: IEEE Xplore Digital Library
2. **HIPAA Compliance Guide**
    - **Title**: Health Insurance Portability and Accountability Act (HIPAA) Compliance Guide
    - **Date**: 2023
    - **Publishing Organization**: U.S. Department of Health & Human Services
    - **Source**: HHS Official Website
3. **GDPR Compliance Handbook**
    - **Title**: General Data Protection Regulation (GDPR) Compliance Handbook
    - **Date**: 2022
    - **Publishing Organization**: European Union
    - **Source**: EU GDPR Official Website
4. **Machine Learning Textbook**
    - **Title**: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow
    - **Authors**: Aurélien Géron
    - **Date**: 2023
    - **Publishing Organization**: O'Reilly Media
    - **Source**: O'Reilly Media
5. **Data Science Best Practices**
    - **Title**: Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking
    - **Authors**: Foster Provost, Tom Fawcett
    - **Date**: 2020
    - **Publishing Organization**: O'Reilly Media
    - **Source**: O'Reilly Media
6. **Cloud Services Documentation**
    - **Title**: Amazon Web Services (AWS) Documentation
    - **Date**: 2024
    - **Publishing Organization**: Amazon Web Services, Inc.
    - **Source**: AWS Documentation
7. **Medical Guidelines for Diabetes**
    - **Title**: Standards of Medical Care in Diabetes—2024
    - **Date**: 2024
    - **Publishing Organization**: American Diabetes Association
    - **Source**: American Diabetes Association

7 Appendices

This section provides supplementary information that supports the main content of the document but would be too detailed or distracting if included in the main sections.

7.1 Appendix A: Glossary of Terms

A comprehensive list of terms and their definitions used throughout the document.

- **AI (Artificial Intelligence)**: Technology enabling machines to perform tasks that typically require human intelligence.
- **Machine Learning (ML)**: A subset of AI focused on the development of systems that learn from data.
- **Feature Extraction**: The process of transforming raw data into structured features for machine learning.
- **XGBoost**: An optimized gradient boosting framework for machine learning.
- **HIPAA**: Health Insurance Portability and Accountability Act, ensuring the protection of sensitive patient data.

7.2 Appendix B: User Interface Mockups

Detailed screenshots or wireframes for:

- **Login Page**: Interface for user authentication.
- **Data Entry Page**: Form for inputting user health data.
- **Dashboard**: Overview of user metrics and insights.
- **Prediction Results Page**: Display of diabetes risk assessment.
- **Settings Page**: User preferences and account settings.

7.3 Appendix C: Database Schema

A detailed schema of the database including:

- Table names
- Column names and data types
- Relationships and foreign key constraints

Example:

Table Name	Description
users	Stores user credentials and personal information
health_data	Stores user-submitted health metrics
Predictions	Stores diabetes risk predictions
Model_parameters	Stores parameters of machine learning models

7.4 Appendix D: System Requirements

A detailed list of system requirements for the GlucoPredict application

Requirement	Description
Hardware	Minimum 8GB RAM, 256GB SSD, Dual-core CPU
Software	Python 3.x, TensorFLow, sckit-learn, MySQL
Storage	Stable internet connect (for cloud sync)
Operating System	HIPPA and GDPR compliance

7.5 Appendix E: Test Cases

A summary of key test cases used for validating the system.

Test Case ID	Test Description	Expected Result	Status
TC001	Login with valid credentials	Redirects to home page	Passed
TC002	Generate a quiz from sample notes	Successfully creates a quiz file	Passed
TC003	Submit health data	Data is stored in the diabetes	Passed
TC004	View diabetes prediction results	Prediction results displayed correctly	Passed

7.6 Appendix F: Error Handling Strategy

A detailed guide on how the system handles different errors.

Error Type	Handling Strategy
Input Validation Error	Display user-friendly error message, highlight the incorrect fields, and prompt user to correct the data.
Database Connection Error	Log the error, retry the connection, and if it fails, notify the user with an appropriate message.
Model Prediction Error	Log the error details, display a fallback message, and prompt the user to retry or contact support.
Authentication Failure	Notify the user of incorrect credentials, provide options for password reset or account recovery.
Server Timeout	Inform the user of the delay, suggest retrying, and log the issue of system monitoring.