Bilkent University

Department of Computer Engineering

Senior Design Project

T2410

DiabetAid

Analysis and Requirements Report

22003508, Çağatay Akpınar, cagatay.akpinar@ug.bilkent.edu.tr

22102677, Sude Topaloğlu, sude.topaloglu@bilkent.edu.tr

22103071, Osman Berkay Artaç, berkay.artac@ug.bilkent.edu.tr

22003595, Yamaç Yiğit Ozan, yigit.ozan@ug.bilkent.edu.tr

22103838, Begüm Kunaç, begum.kunac@ug.bilkent.edu.tr

Supervisor: Halil Altay Güvenir

Course Instructors: Atakan Erdem, Mert Bıçakçı

December 13, 2024

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/1.

Contents

Analysis and Requirement Report

DiabetAid

1. Introduction

Diabetes mellitus is a chronic disease that occurs when the pancreas produces insufficient insulin or when the body cannot effectively use the insulin it produces. Insulin is a vital hormone responsible for regulating blood sugar levels. Uncontrolled diabetes leads to elevated blood sugar, or hyperglycemia, which can cause long-term damage to various organs, including the nerves, blood vessels, and kidneys.

Diabetes is a growing health concern globally. In 2022, 14% of adults aged 18 and older were living with diabetes, and more than 59% of adults aged 30 and older with diabetes were not receiving medication. The disease remains a leading cause of death, directly causing 1.6 million deaths in 2021 alone, with almost half occurring before the age of 70 years. Additionally, diabetes is responsible for 11% of deaths due to cardiovascular disease and 530000 deaths due to kidney diseases [1].

Our project, DiabetAid, is being developed as an innovative solution to improve diabetes care through machine learning (ML) techniques. This project, in collaboration with Ankara Bilkent City Hospital, aims to provide personalized recommendations for medication and dosage to support doctors in making accurate treatment decisions. By analyzing patient data, such as age, gender, BMI, laboratory results, current medical conditions, and current treatments, DiabetAid uses ML algorithms to deliver precise, real-time suggestions during patient visits. DiabetAid is a user-friendly application that integrates seamlessly with hospital systems and supports doctors in making accurate treatment decisions more efficiently. This accelerates the treatment stage, reduces the workload on healthcare professionals, and enhances the overall quality of diabetes management.

2. Current System

Diabetes management systems currently in use are majorly based on blood glucose monitoring and recording of health data by patients. These tools mostly do not integrate with advanced machine learning models that can offer actionable, personalized treatment recommendations. Most existing systems, including common electronic health record (EHR) systems, offer only general guidelines or standardized options of treatments that do not consider variability in individual patients, including comorbid conditions, laboratory results, and personal medical history.

On the other hand, DiabetAid tries to bridge this gap by utilizing XAI techniques to deliver personalized medication recommendations and dosage, supported by explainability. Though other systems may also give similar recommendations, they are usually not explainable, hence leaving doctors with little understanding of how the AI arrived at such a decision.

DiabetAid's innovation is that it can dynamically analyze patient data, generate insights based on the data, and provide recommendations along with their rationale in real time during a patient visit. This system is designed to directly support doctors in improving the efficiency and accuracy of diabetes management while maintaining trust through transparent explainability. Unlike existing solutions, DiabetAid puts the doctor’s role first, assisting rather than replacing medical expertise, while also offering a scalable framework that is easily deployable across a number of healthcare institutions.

3. Proposed System

3.1 Overview

DiabetAid is a pioneering project that puts into use some of the advanced ML techniques in order to change the way diabetes is managed and treated. The application is designed to make personalized, real-time recommendations of medication and dosage during a patient visit, greatly enhancing efficiency and accuracy in diabetes care. Seamlessly integrating with hospital systems, DiabetAid supports doctors in making informed clinical decisions, thereby streamlining the treatment process and enhancing patient outcomes.

The system analyzes patient data: age, gender, body mass index, laboratory test results, current health conditions, and treatments. This information goes through strong ML algorithms to output recommendations with high precision and personalized to the needs of a particular patient. One of the most important novelties of DiabetAid is its focus on XAI, making recommendations with clear and understandable explanations. This feature enhances transparency, builds trust, and empowers doctors to validate the system's suggestions confidently.

DiabetAid also introduces a user-friendly interface, which can be managed by health professionals with any level of computer experience. This application not only accelerates the treatment phase but also reduces the workload on medical staff so that doctors can focus more on patient care. By helping in clinical decisions, DiabetAid addresses some of the key challenges in diabetes management, including personalization of treatment and efficient use of resources.

In cooperation with Ankara Bilkent City Hospital, the solution shall be scalable and easy to deploy in several healthcare institutions. It will first be implemented as a web application for doctors and should iteratively improve through integration with more functionalities. These will involve feedback mechanisms that increase model performance over time and adaptability for new datasets or medical environments.

The most important feature of DiabetAid is that it is capable of bringing a revolution in the healthcare system by the betterment of critical challenges like under-treatment and mismanagement of diabetes. Doctors can log in, introduce the patient, go through recommendations, and may even update the treatment history. This application guarantees patients' data privacy and safety through anonymization and in accordance with international standards, among which KVKK and ISO feature.
The scope of the treatment also goes beyond diabetes, with DiabetAid using machine learning to determine the most efficient treatments for other chronic diseases. Since this project follows ethical considerations with AI and also the existing standards within healthcare, the shift to personalized, transparent, and efficient medicine is obvious here. Ultimately, DiabetAid makes a statement on the transforming role of AI in contemporary medicine and sets a new standard for personalized healthcare tools.

3.2 Functional Requirements

- The doctor must be able to securely log in to the system using hospital-provided credentials.
- The doctor should be able to input patient data such as demographics, lab results, and medical history into the application.
- The doctor should be able to receive personalized treatment recommendations, including medication and dosage, based on the patient's data.
- The doctor should be able to view and update treatment history for each patient through the system.
- The doctor should be able to access support or help documentation through the application interface.
- The doctor should be able to generate and download a detailed report summarizing patient data, recommendations, and treatment history.
- The doctor should be able to provide feedback on the system's recommendations to improve machine learning model performance over time.
- The system should analyze patient data using machine learning models to generate personalized treatment recommendations, including medication and dosage.
- The system must store treatment logs and recommendations securely to allow for future reference and follow-up.

3.3 Nonfunctional Requirements

3.3.1 Usability

- The application should be easy to use, with a simple and clean design to help doctors of all technical levels.
- Clear instructions, tooltips, and error messages should guide users and make the system easy to navigate.
- Data entry should be quick and efficient so that it doesn't slow down doctors during patient care..

3.3.2 Reliability

- The system must work well even if some patient data is missing, by providing alerts or making reasonable assumptions.

- It should be available and ready to use at all times since it supports important medical decisions.
- If the VPN connection fails, the system should save any entered or processed data and try to reconnect automatically.
- The recommendations must be accurate, using well-tested machine learning models to ensure reliability.

3.3.3 Performance

- The system should provide recommendations quickly after the doctor enters patient data, ensuring it does not slow down the workflow.
- The system must handle large patient datasets quickly and efficiently.

3.3.4 Supportability

- Updates should be simple to apply without interrupting hospital operations.
- The system should connect smoothly with hospital systems like electronic health records (EHRs) or lab reports.

3.3.5 Scalability

- The system should handle more users, patient data, and hospitals as it grows without performance issues.
- The system should support multiple doctors using it simultaneously without performance issues or slowing down.

3.4 Pseudo Requirements

- Python will be used as the programming language for machine learning model development and data processing tasks.
- React and Flask frameworks will be used to develop the front-end and back-end of the system, which will ensure a smooth user interface and server functionality.
- TensorFlow and Scikit-learn libraries will be used to train and test machine learning models.
- LIME (Local Interpretable Model-Agnostic Explanations) will be used to ensure explainability in the model's recommendations.

- The patient data will be stored and preprocessed in CSV and Excel formats before feeding into the system.
- Git with GitHub will be used as a version control system to track the development progress and to share the work within the team.
- Jira will be used for project management, including task tracking to ensure smooth development processes and teamworks and that deadlines are met.
- SSL/TLS protocols will be implemented to secure communication between the application, hospital servers, and doctor devices.
- For the secure access of patient data from the hospital database, a Hospital VPN will be utilized.
- Data will be anonymized and temporarily stored in structured formats according to KVKK and GDPR regulations.
- Integration of the EHR will be according to HL7 standards in order to ensure compatibility with existing systems in the hospital.
- The system will initially be deployed on the computers in the hospital at Ankara Bilkent City Hospital and will be designed for easy scaling to other institutions.

3.5 System Models

3.5.1 Scenarios

Login:

- Use-case Name: Login
- Actor: Doctor
- Entry Condition: The doctor has the system credentials and opens the DiabetAid system.
- Exit Condition: The doctor successfully logs in into the system and accesses its functionality.
- Flow of Events:
    1. The doctor navigates to the login page of the DiabetAid system.
    2. The doctor enters their username and password provided by the hospital.
    3. The system verifies the username and the password against the database.

4. If the credentials are correct, the doctor is directed to the main dashboard.
5. If the credentials are incorrect, an error message prompts the doctor to re-enter valid credentials.

- Alternative Flow: The system, upon detecting suspicious login attempts (like multiple failed attempts), temporarily locks the account and sends a notification to the IT administrator at the hospital.

Enter New Patient:

- Use-case Name: Enter New Patient
- Actor: Doctor
- Entry Condition: The doctor is logged into the system and has a new patient's information to record.
- Exit Condition: The new patient's data is successfully stored in the system.
- Flow of Events:
    1. The doctor selects the "Enter New Patient" option from the dashboard.
    2. The system shows a form for input regarding patient details: name, age, gender, body mass index, laboratory results, and current medical conditions.
    3. The doctor fills in the provided information and clicks "Save".
    4. The system validates the correctness of the input fields such as no missing mandatory field.
    5. The data gets saved in the system database upon successful validation of the input fields.
    6. A confirmation message is displayed.
- Alternative Flow: In case the incomplete record is attempted to be saved by the doctor, the system highlights the missing fields and prompts for correction.

Search Patient:

- Use-case Name: Search Patient
- Actor: Doctor
- Entry Condition: The doctor has logged in and wants to retrieve details of an already existing patient's records.

- Exit Condition: The doctor views the records of the patient successfully.
- Flow of Events:
    1. The doctor chooses the "Search Patient" option on the dashboard.
    2. The system displays a search bar where the doctor enters the patient’s ID.
    3. The system searches the records in the database.
    4. If a match is found, the system will display the patient information along with the treatment history and medications.
    5. The doctor selects the patient record to view additional details.
- Alternative Flow: If no matching record is found, a message is displayed: "Patient not found. Please verify the entered information."

Update Patient Records:

- Use-case Name: Update Patient Records
- Actor: Doctor
- Entry Condition: The doctor decides to update a patient's record, possibly to modify the treatment plan.
- Exit Condition: The modification of the patient record in the system is successfully reflected.
- Flow of Events:
    1. The doctor accesses the patient record through the use of the "Search Patient" feature.
    2. The system displays the details and treatment history of the patient.
    3. The doctor modifies fields like medication, dosage, or other medical information.
    4. The doctor clicks the "Update" button to save changes.
    5. The system validates the updated fields and confirms the changes.
    6. A success message appears, and the updated record is stored in the database.
- Alternative Flow: If validation fails, for example due to an invalid data format, the system highlights the errors and prompts the doctor to correct them.

Submit Feedback:

- Use-case Name: Submit Feedback
- Actor: Doctor
- Entry Condition: The doctor has reviewed a treatment recommendation and wants to provide feedback.
- Exit Condition: Feedback is successfully submitted to the system for model refinement.
- Flow of Events:
    1. The doctor accesses the "Submit Feedback" option on the treatment recommendation page.
    2. The doctor fills out the text form and clicks "Submit".
    3. The system validates the input fields and stores the feedback.
    4. A confirmation message is displayed to the doctor.
- Alternative Flow: If the doctor submits incomplete feedback, the system prompts them to complete all required fields.

Logout:

- Use-case Name: Logout
- Actor: Doctor
- Entry Condition: The doctor has completed their session and selects the logout option.
- Exit Condition: The doctor is logged out of the system and the session is terminated.
- Flow of Events:
    1. The doctor clicks the "Logout" button on the dashboard.
    1. The system prompts the doctor to confirm the logout action.
    2. If confirmed, the system logs out the doctor and redirects to the login page.
    3. The system terminates the session and clears any cached data.
- Alternative Flow: If the logout process is interrupted (e.g., network failure), the system ensures the session is terminated securely upon reconnection.

View Patient Data:

- Use-case Name: View Patient Data
- Actor: Doctor
- Entry Condition: The doctor has searched for a patient wanting to check patient data.
- Exit Condition: The doctor is done with the patient data.
- Flow of Events:
    1. The doctor clicks the “View Patient Data” button after a successful search result of an existing patient.
    2. The system retrieves the data from the database and shows the results to the doctor.
    3. If the doctor wants to go back, the doctor clicks the “Back” button.
    4. The system returns the user back to the search page.
- Alternative Flow: If the data cannot be fetched from the database, an error message is displayed.

Receive Treatment Recommendation:

- Use-case Name: Receive Treatment Recommendation
- Actor: Doctor
- Entry Condition: The doctor wants to take treatment advice from the system for a patient whose data exists in the database.
- Exit Condition: The doctor reads the recommendation and explanations and wants to go back to the dashboard.
- Flow of Events:
    1. The doctor searches for a patient whose data is in the database from the “Search Patient” tab.
    2. Doctor clicks on the requested patient data from the results obtained from the search.
    3. Doctor clicks the button “Get Treatment Recommendation” to learn what the system recommends for the selected patient.
    4. After reading all the explanations and recommendations, the doctor goes back to the dashboard or searches for another patient.

- Alternative Flow: If recommendation or explanation cannot be made because of either missing data or error running the AI module, a descriptive error message is displayed to the doctor.

Receive Explanation:

- Use-case Name: Receive Explanation
- Actor: Doctor
- Entry Condition: The doctor wants to take treatment advice from the system for a patient whose data exists in the database.
- Exit Condition: The doctor reads the recommendation and explanations and wants to go back to the dashboard.
- Flow of Events:
    1. Doctor clicks the button "Get Treatment Recommendation" to learn what the system recommends for the patient.
    2. For each recommendation, the doctor will be informed about which medication is recommended for which reason by the AI explanation system.
    3. After reading all the explanations for each medication recommended, the doctor either returns to the dashboard and searches for another patient, or chooses to give feedback for the explanation.
- Alternative Flow: If recommendation or explanation cannot be made because of either missing data or error running the AI module, a descriptive error message is displayed to the doctor.

Generate Recommendation:

- Use-case Name: Generate Recommendation
- Actor: AI Recommendation Model
- Entry Condition: Patient data arrives at the recommendation system.
- Exit Condition: Prediction of the model is returned back to the doctor who requested the recommendation.
- Flow of Events:
    1. Patient data arrives at the recommendation system.

2. System gives the data as input to the AI model that is trained to make treatment predictions.
3. The AI model generates treatment recommendations for the given data.
4. The recommendation information is sent to the frontend and shown to the doctor.

- Alternative Flow: If an internal error occurs while sending the data to the AI recommendation model or while running the AI recommendation model, a descriptive error message is displayed.

Generate Explanation:

- Use-case Name: Generate Explanation
- Actor: AI Recommendation Model
- Entry Condition: Patient data arrives at the recommendation system.
- Exit Condition: Prediction of the model is returned back to the doctor who requested the recommendation.
- Flow of Events:
    1. Model recommends a treatment for the given patient data.
    2. These recommendations and the patient data are given to the explanation module.
    3. The explanation module generates explanations using the training data, patient data and the treatment prediction.
    4. The explanations are sent to the frontend and shown to the doctor.
- Alternative Flow: If an internal error occurs inside the AI prediction model, a descriptive error message is displayed and sent to the frontend.

Handle Missing Data:

- Use-case Name: Handle Missing Data
- Actor: AI Recommendation Model
- Entry Condition: The doctor requests for a treatment recommendation.
- Exit Condition: Missing data is handled and the patient data is ready to be sent to the recommendation module.
- Flow of Events:

1. The doctor clicks on the button “Get Treatment Recommendation” and the system recognizes that there are missing values to be handled.
2. The missing values are handled by copying from the data of the patient from the past.
3. The resulting data is sent back to the AI recommendation module.

- Alternative Flow: If the missing values can not be filled from past data, the doctor is prompted to fill by hand or the data is filled automatically.

3.5.2 Use-Case Model

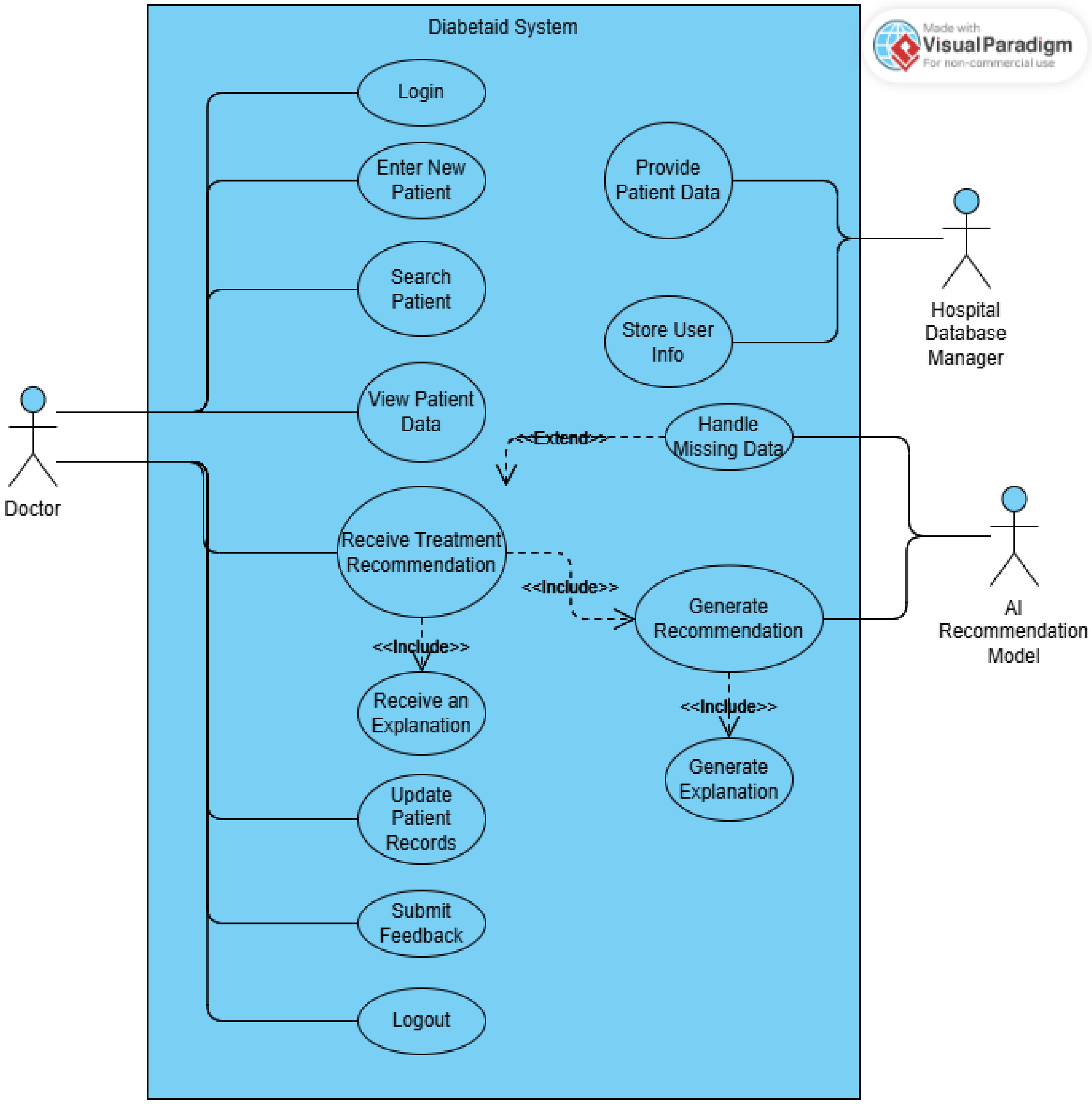

Figure 1: Use Case Model.

3.5.3 Object and Class Model

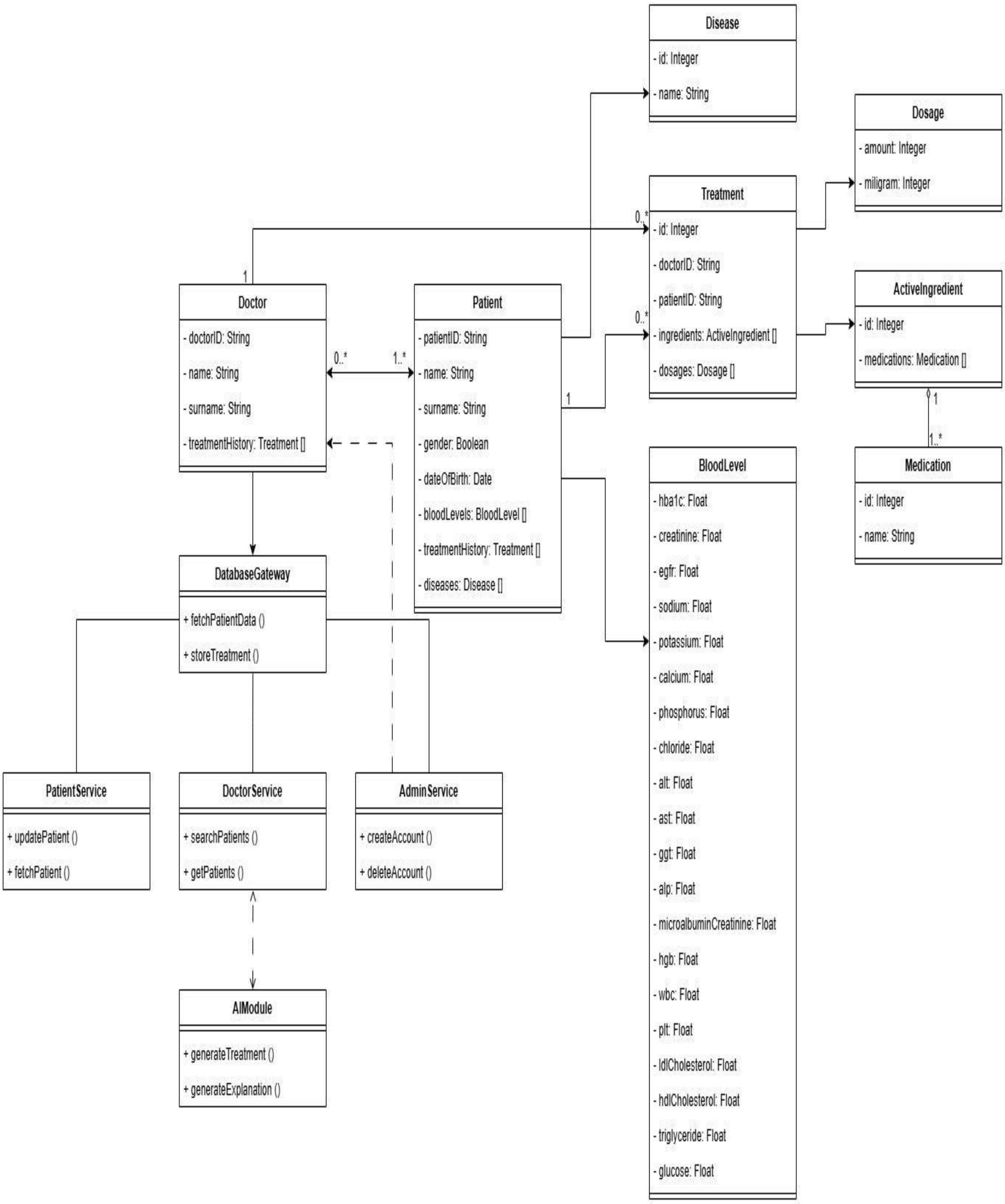

Figure 2: Class and Object Model.

[Link For the Class Diagram]

3.5.4 Dynamic Models

3.5.4.1 Activity Diagrams

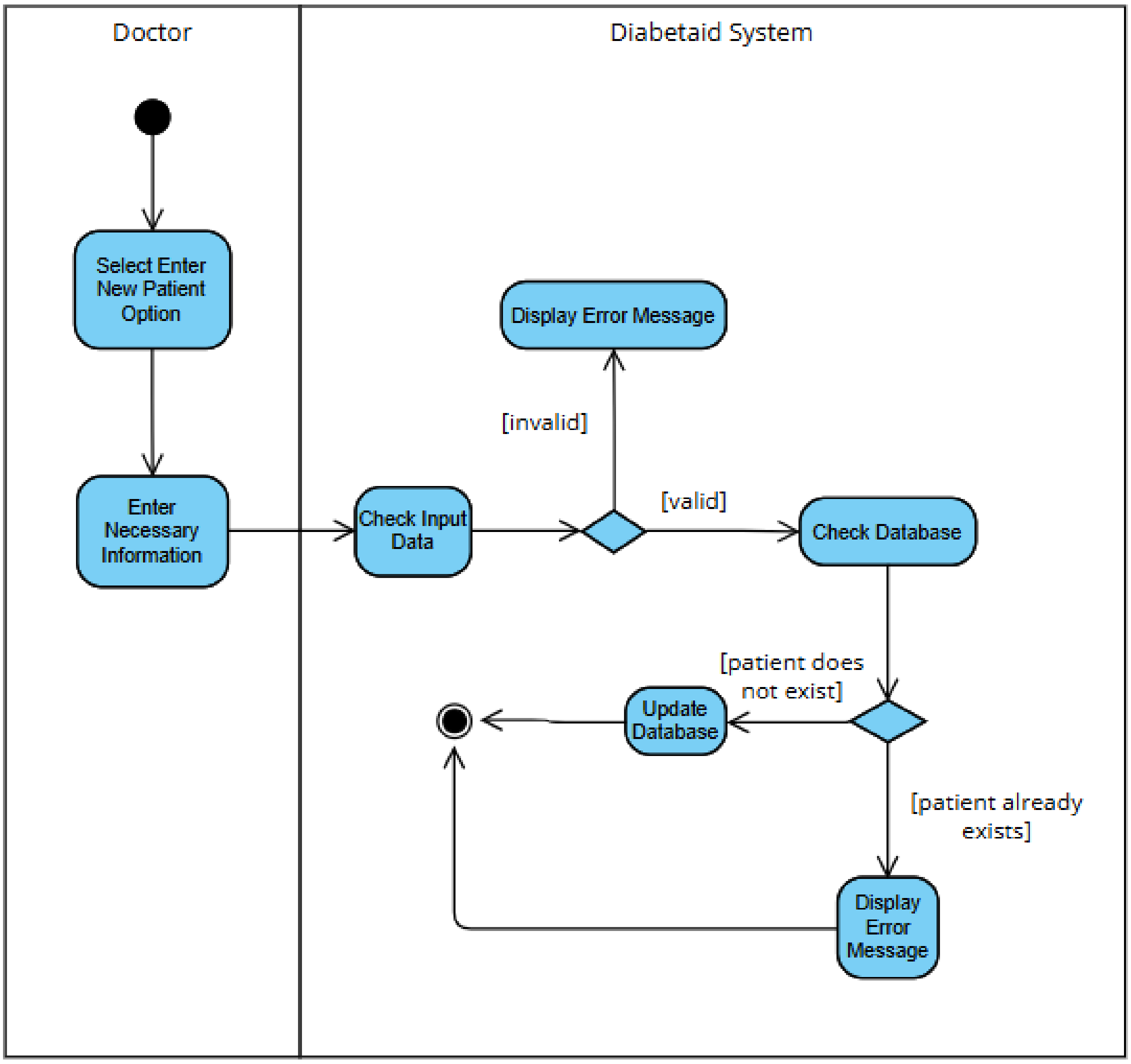

Figure 3: Enter New Patient Data Activity Diagram.

This activity diagram describes the activity of the "Enter New Patient" in the DiabetAid system. The doctor selects the option and then inputs the details of the patient. Following that, the system checks whether the input is valid. In the case of invalid data, an error message appears. If the input is valid, it checks the database for a patient with such details. If the patient already exists, it displays an error message; otherwise, it updates the database with the new patient information, completing the process. This ensures that there is accurate data entry without duplications.

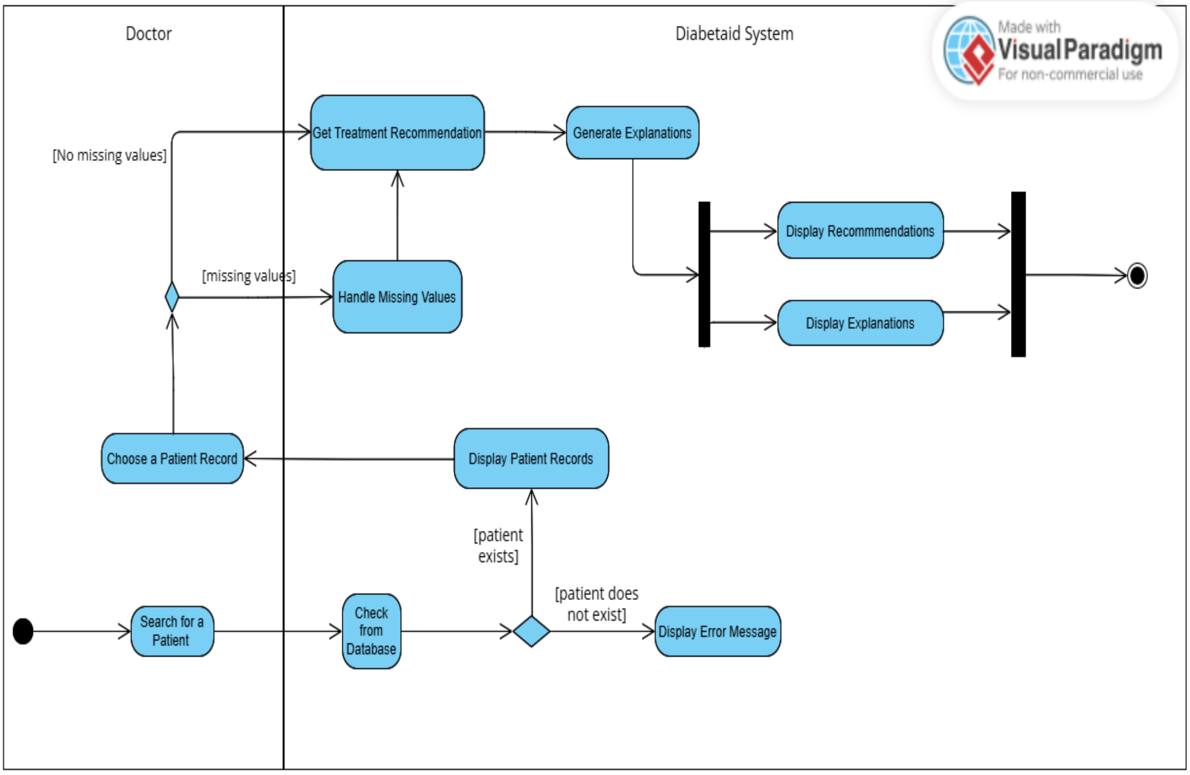

Figure 4: Get Treatment Recommendations Activity Diagram.

This activity diagram shows the process of generating treatment recommendations and explanations in the DiabetAid system. It starts with the doctor searching for a patient, after which the system checks the database. In case the patient exists, it displays the records; otherwise, it shows an error message. If the patient data has missing values, then these are handled first. The system then generates treatment recommendations with explanations to be presented to the doctor once all data is complete. This ensures accurate and transparent treatment recommendations while handling incomplete data.

3.5.4.2 Sequence Diagrams

Sequence Diagram of Searching for a Patient Record and Getting Recommendations:

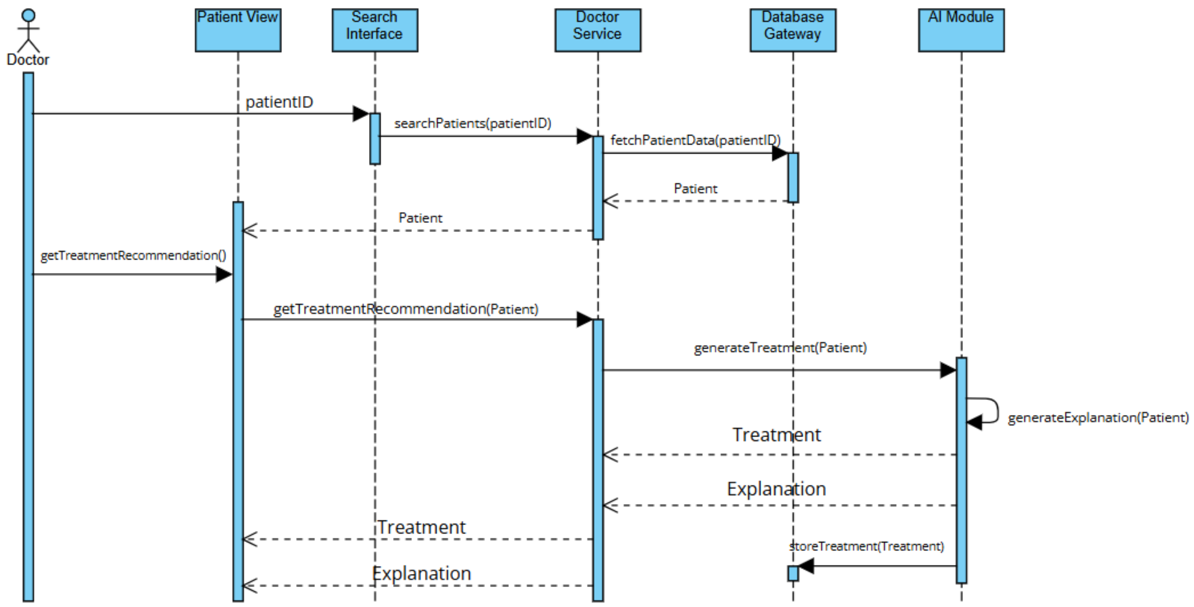

Figure 5: Get Treatment Recommendation from Existing Data Sequence Diagram.

Sequence Diagram of Entering New Patient Data and Getting Recommendations:

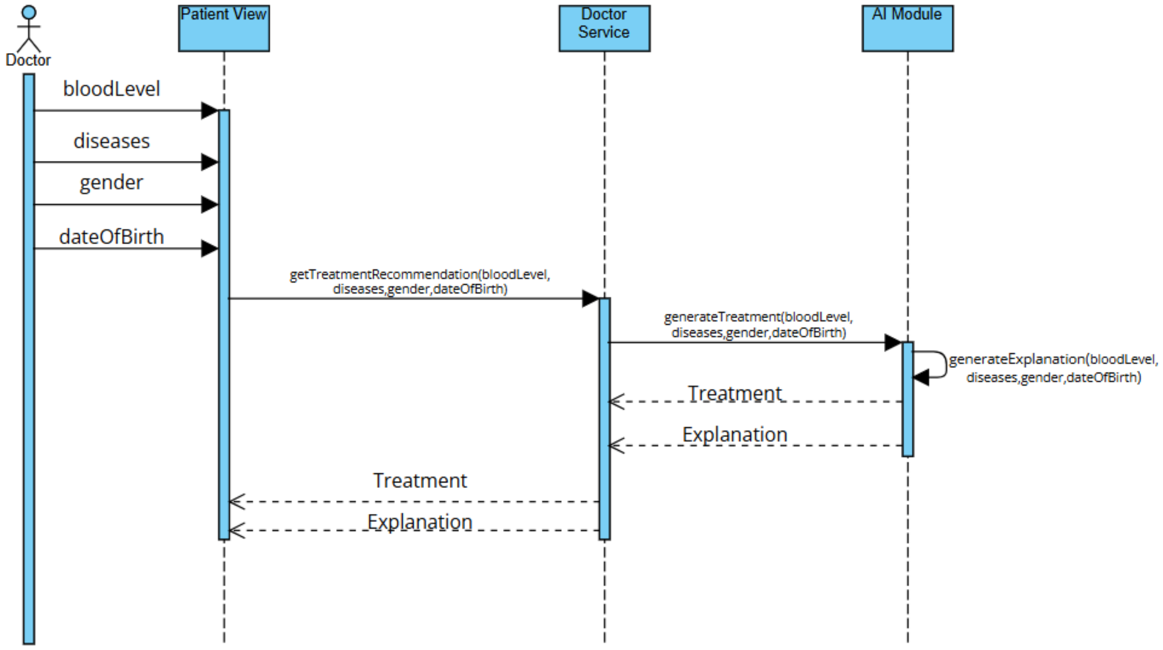

Figure 6: Get Treatment Recommendation from New Data Sequence Diagram.

3.5.5 User Interface

Login Screen:

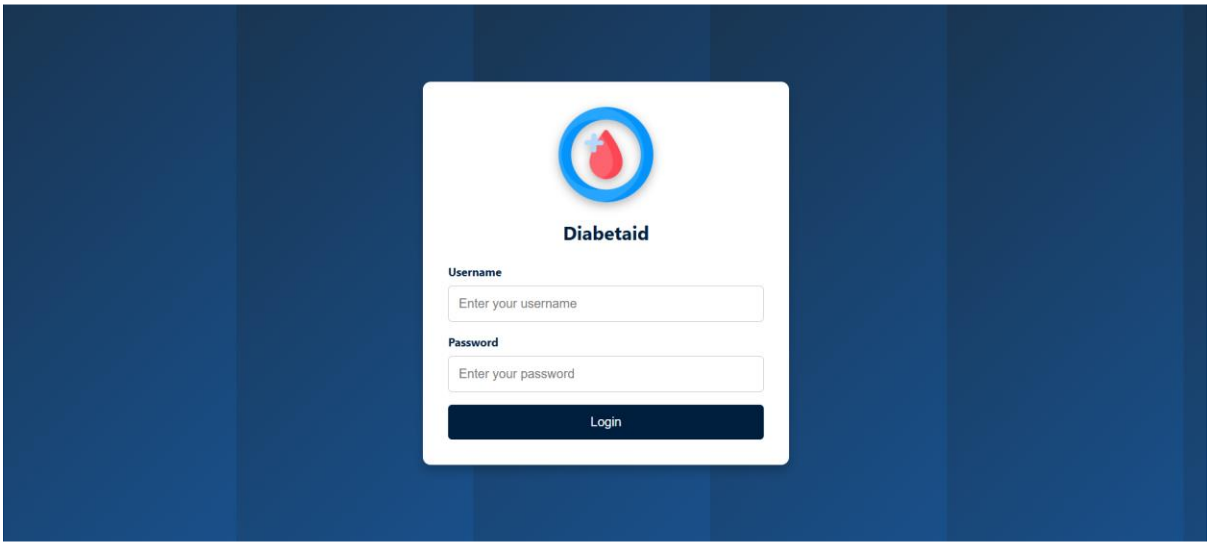

Figure 7: Login Screen.

Main Dashboard Screen:

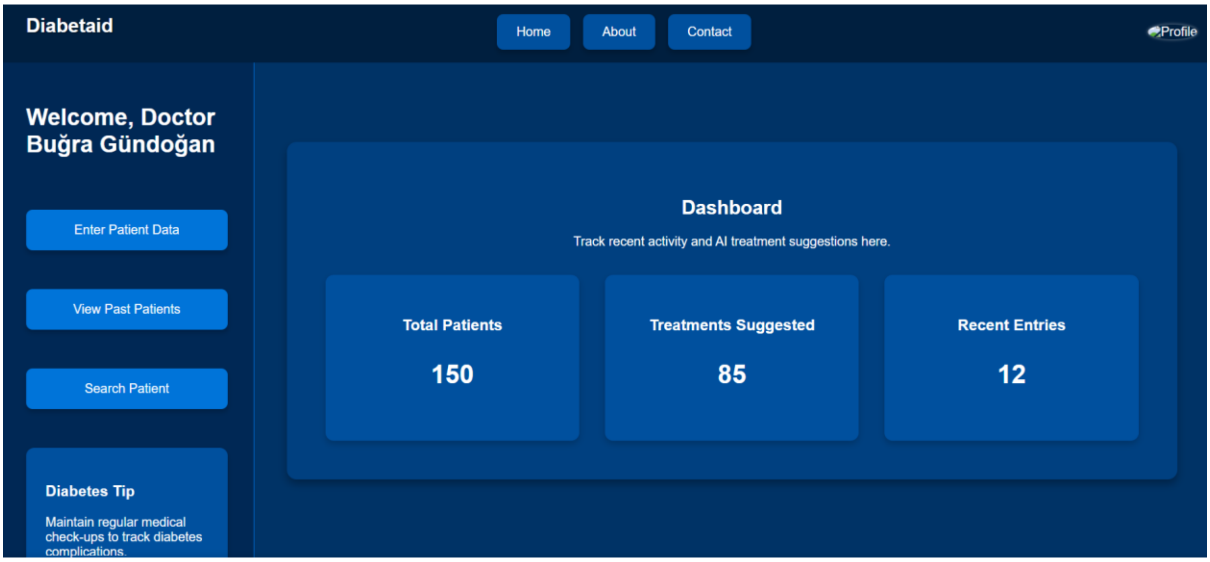

Figure 8: Main Dashboard Screen.

Enter New Patient Data Screen:

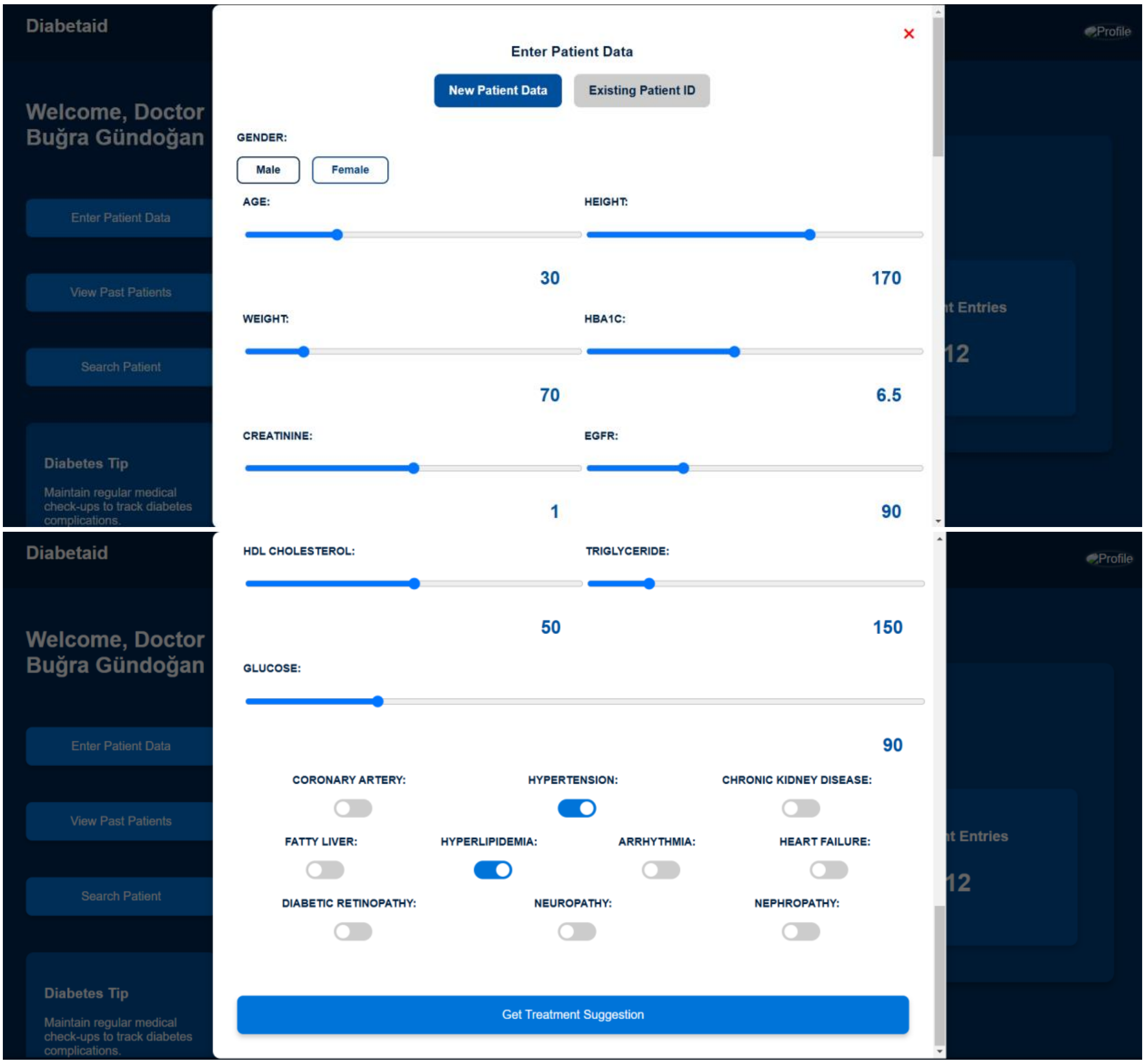

Figure 9, 10: Enter New Patient Data Screen.

Treatment Suggestion and Explanations Screen:

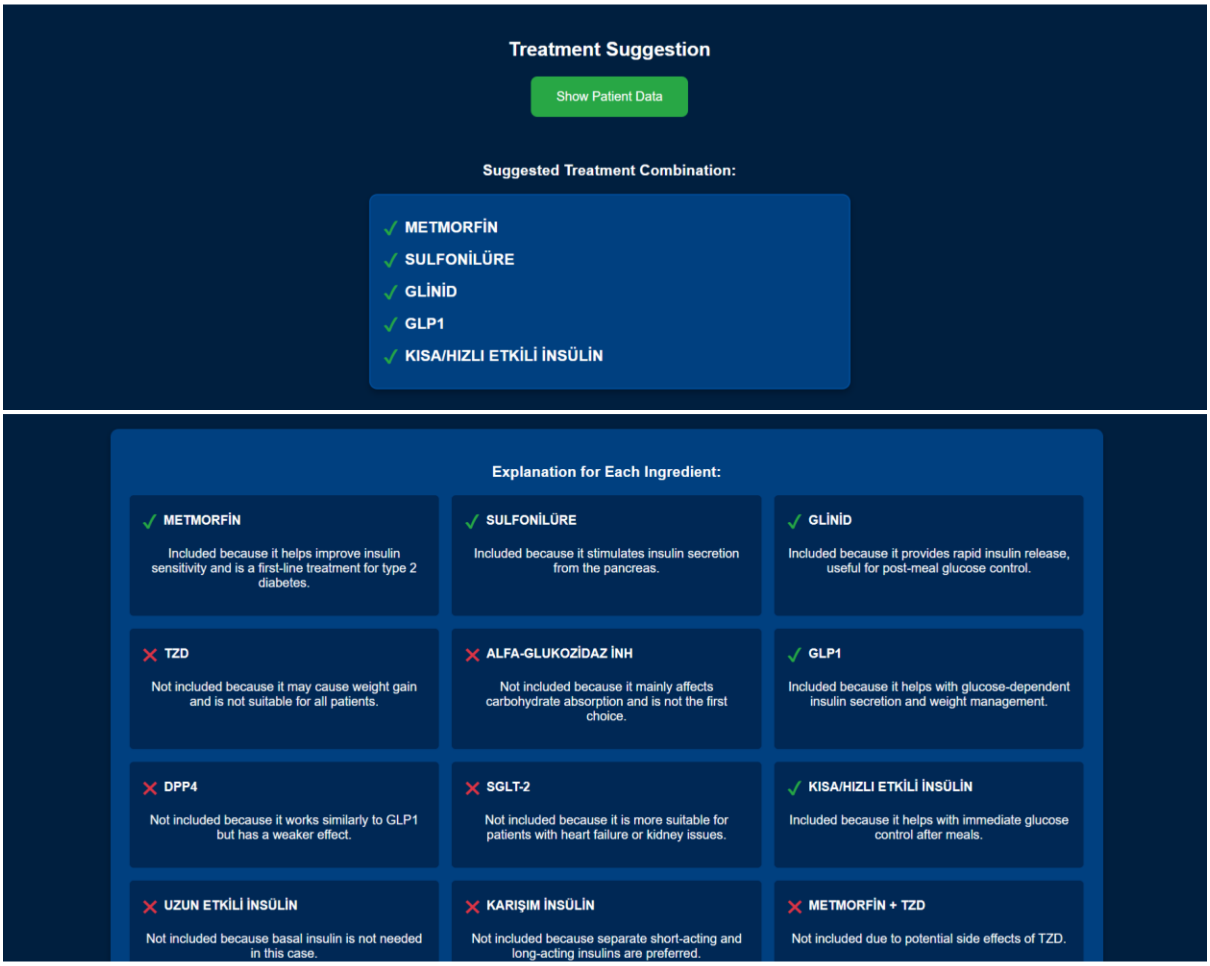

Figure 11, 12: Treatment Suggestion and Explanations Screen.

Search a Patient From Database Screens:

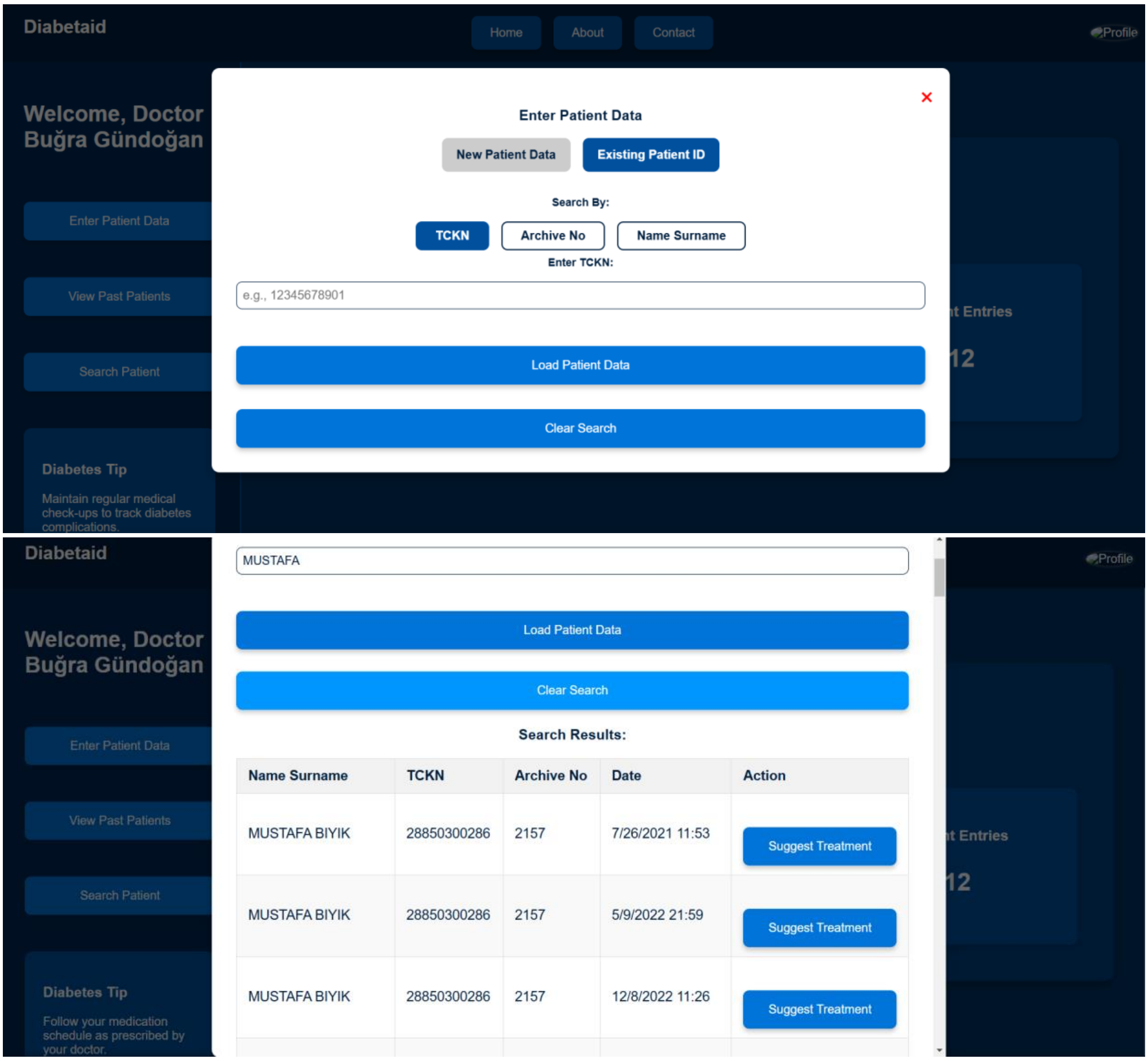

Figure 13, 14: Search a Patient from Database Screens.

4. Other Analysis Elements

4.1 Consideration of Various Factors in Engineering Design

4.1.1 Constraints

Implementation Constraints

- The project will be implemented as a web solution for the doctors in personal computers running on web at the Ankara Bilkent City Hospital and then adapted for use in other hospitals.

- GitHub and Git will be used for version control and collaboration during development.
- Jira will be used for project management and tracking development tasks to ensure smooth progress.
- Python will be the main programming language due to its strong libraries for machine learning and data processing.
- TensorFlow, Scikit-learn and LIME will be used for training, testing and explaining machine learning models.
- Patient data will be retrieved via the hospital's VPN and temporarily stored in structured formats like CSV or Excel for processing.
- The interface will be built using React and Flask to ensure ease of use for hospital staff.

Economic Constraints

- This will be costlier if the system were to be scaled up to other hospitals or institutions, considering infrastructural adaptation, permission that will have to be taken, probably retraining the model on new data.
- The system uses open-source libraries and tools, which eliminates licensing costs.
- Data storage and VPN access are provided by the hospital at no additional cost, reducing the overall expenses.
- There are no fees associated with using the hospital's existing IT infrastructure, as it is already maintained by the hospital's team.

Ethical Constraints

- All the information related to patients is anonymized. No personal data, including names or IDs, is either kept or processed in any form that would lead to an individual being identified. The project is in conformance with KVKK law to protect the security of information.
- The project has approval from the hospital's ethics committee to use patient data. If the system is expanded to other hospitals, new permissions will be required.

- Patient data is only used for training the system and providing treatment recommendations. Data is deleted immediately after processing to avoid unnecessary storage.
- The system will clearly explain how it generates its recommendations so that doctors can understand the logic and trust the outputs.
- The system will be a tool to assist doctors, not replace their expertise. Doctors are always responsible for the final treatment decisions.

4.1.2 Standards

In the development of DiabetAid, several international and local standards are considered to ensure safety, reliability, and compliance, particularly given the sensitive nature of medical data in Turkey. Below are the key standards considered:

KVKK (Personal Data Protection Law)

- KVKK is the Turkish version of the EU GDPR for the protection of personal data. All the patient data utilized in DiabetAid is anonymized and undergoes secure processing to fully comply with KVKK recommendations. This includes taking prior permission from the hospital's ethics committee for the use of patient data with the view of protecting their privacy and maintaining confidentiality [2].

ISO/IEC 27001 - Information Security Management System

- The ISO/IEC 27001 standard ensures that all patient information handled by DiabetAid is well kept and processed with security in mind. The security management of health data shall always have this standard to minimize the risks associated with data breach and unauthorized access [3].

ISO 9241 - Usability and Human-Centered Design

- It should be user-friendly for any and all medical practitioners irrespective of their technical background. ISO 9241 presents a standard of usability that gives rules for designing user interfaces with the intention of making DiabetAid as intuitive and effective as possible for care providers [4].

ISO 14971 - Risk Management for Medical Devices

- Since DiabetAid is a treatment recommender, there will need to be some regulation of risk and its mitigation regarding its use. ISO 14971 has been applied for the identification, assessment, and control of risks associated with patient safety so that the system can advance healthcare with safety for the patients [5].

ISO 62304 - Software Life Cycle Processes for Medical Devices

- Development of software for DiabetAid is based on the accepted standard ISO 62304 for guidance of design, development, and maintenance of software for medical devices. This standard provides assurance that our approach is well-organized and managed with substantial emphasis on software quality, safety, and reliability [6].

IEEE 11073 - Health Informatics

- DiabetAid uses the IEEE 11073 standard for the integration of the application with medical devices and health informatics systems. Adherence to this standard will ensure the smooth functionality of the application with the available hospital systems hence ensuring successful capture and sharing of information [7].

HL7 (Health Level Seven)

- This will work with most EHR systems available today since it uses HL7 standards, which are common in hospitals and facilitate the interaction between different applications through the interface by providing smooth connections and sharing of data. This enables DiabetAid to communicate efficiently with other hospital information systems for better output in patient care [8].

ISO 13485 - Medical Device Quality Management Systems

- DiabetAid has been developing and continuously maintaining following the norms of ISO 13485 for quality management in the development of medical software solutions. This helps the product maintain high-quality standards throughout the life cycle of the product [9].

Open-Source Software Compliance

- DiabetAid uses different open-source tools like TensorFlow and Scikit-learn. The project follows all the licensing conditions and hence is with the policy of open-source software, which avoids all the possible legal issues concerning intellectual property rights accordingly [10].

SSL/TLS Encryption

- Communication between DiabetAid, hospital servers, and doctor devices is secured using **SSL/TLS** encryption protocols. This ensures data integrity and confidentiality during transmission [11].

Together, these standards ensure that DiabetAid is built to be a reliable, secure, and user-friendly healthcare tool. By following these guidelines, we aim to enhance diabetes treatment while keeping patient safety and data privacy at the highest level.

4.2 Risks and Alternatives

There are some possible risks that could halt the progress of the project. It is critical to consider the probabilities of each one of these risks and contemplate on them to come up with solutions and plans if any of them occurs.

Risk1: Model Could not Reach Expected Accuracy

Machine learning part is at the heart of our application and the accuracy of the model will decide the success and usability of the application. There is not a guarantee that the model will be successful to identify the patterns of the dataset and come up with more accurate treatment decisions than that of doctors. For this case the solution is finding ways to access more data from the hospital since a bigger dataset means higher accuracy for most machine learning models.

Risk 2: Integration Problems with the Hospital System

For the system to be usable by doctors, it has to be integrated with the hospital system. However, there is a risk that the integration will be problematic and the system may not work up to the desired and envisioned standards. This problem could be solved by moving the system to the cloud and making the doctor access the application through the cloud server.

Risk 3: Explanations are Not Satisfactory for the Doctors

The system is designed for the health industry and it makes decisions about patient health. This means that the system has to convince the doctors that the decisions are not coming without reason. Convincing explanations for each prediction is necessary for the application to help the doctors do their jobs. If these explanations are not convincing enough, the B plan is to reconsider the explanation mechanism and go for other options to explain the predictions of black-box machine learning models.

	Likelihood	Effect on the project	B Plan Summary
Risk 1	Low	Doctors will not use the product if it gives frequent wrong and unacceptable results.	Train the model with more data and get better accuracy.
Risk 2	Moderate	Doctors would have difficulties using the product without seamless integration.	Use cloud servers to facilitate easy integration and access for the doctors.
Risk 3	High	Doctors will not be willing to accept the recommendations unless they are given convincing and detailed explanations.	Go for different explanation methods to render more convincing explanations.

Table 1: Risks.

4.3 Project Plan

WP#	Work package title	Leader	Members involved
WP1	CS491 Demo and Presentation	Çağatay	All Members
WP2	Detailed Design Report	Sude	All Members
WP3	Design Project Final Report	Begüm	All Members
WP4	Final Demo and Presentation	Çağatay	All Members
WP5	Data Collection and Preprocessing	Yamaç	All Members
WP6	Model Training and Improvement	Osman	Çağatay, Osman, Yamaç
WP7	Discussions with Stakeholders	Sude	Sude, Çağatay

Table 2: List of Work Packages.

WP 1: CS491 Demo and Presentation			
Start date: Dec 5, 2024 **End date:** Dec 19, 2024			
Leader:	*Çağatay*	**Members involved:**	All Members
Objectives: Introducing the project and demonstrating work done			

Tasks:

Task 1.1 Coding and Designing the UI: *User interface for the app will be implemented. The design will focus on the requirements and preferences of the doctors who are the target users of the application.*

Task 1.2 Establishing Connection with the Database: *Integrating the application with a database with some mock patient data to demonstrate how the application works to the audience.*

Task 1.3 Preparation of the Presentation**:** *Slides will be prepared in a well-organized manner to introduce the project to the audience. Every*

member of the team will present some part of the project and work that was done in a concise way.			
Deliverables ***D1.1:*** *CS491 Demo and Presentation (Dec 20, 2024)*			
WP 2: Detailed Design Report			
Start date: *Dec 20, 2024* **End date:** *Feb 10, 2025*			
Leader:	*Sude*	**Members involved:**	*All Members*
Objectives: Designing and documenting the overall architecture of the project using diagrams and charts. Writing test cases and experimenting with different scenarios.			
Tasks: ***Task 2.1 Drawing the Diagrams:*** *Architecture diagrams must represent the implementation details comprehensively including all the subsystems utilized in the application.* ***Task 2.2 Preparation of the Report*****:** T*he other parts of the report including setting design goals will be completed and specific test cases will be designed to test the robustness of the application.*			
Deliverables ***D2.1:*** *Detailed Design Report (Feb 10, 2025)*			
WP 3: Design Project Final Report			
Start date: Feb 11, 2025 **End date:** Apr 24, 2025			

Leader:	*Begüm*	**Members involved:**	*All Members*
Objectives: Documenting work done over the course of the project and teamwork. Testing the application using the previously designed test cases.			
Tasks: ***Task 3.1 Testing of the Application*****:** Tests will be performed to ensure the application is working to high standards. ***Task 3.2 Preparation of the Report*****:** Report will be prepared in a professional manner with attention to detail. Contributions of each member will be mentioned in this report.			
Deliverables ***D3.1:*** *Design Project Final Report (Apr 24, 2025)*			
WP 4: Final Demo and Presentation			
Start date: Feb 11, 2025 **End date:** 3rd Week of May			
Leader:	*Çağatay*	**Members involved:**	*All Members*
Objectives: Presenting the work done and demonstrating *the final product to the jury and the audience*			
Tasks: ***Task 4.1 Finishing up the Coding Process*****:** *The application must be well-tested and working with all the important features implemented to a satisfactory level.* ***Task 4.2 Presentation and Demo Preparation*****:** *Every member of the team will present what they have done over the course of the year. The order of the features to be demonstrated will be decided and planned for a successful demonstration.*			
Deliverables ***D4.1:*** Final Presentation and Demo (3rd week of May)			

WP 5: Data Collection and Preprocessing

Start date: Dec 20, 2024 **End date:** Apr 24, 2025

Leader:	*Yamaç*	**Members involved:**	*All Members*

Objectives: Retrieving patient data including laboratory test results, chronic diseases, and treatments that are given to them. Processing the data into a form that is learnable by a machine learning model.

Tasks:

Task 5.1 Retrieving Raw Data from the System: *The patient information will be retrieved from the hospital system in two ways, from the doctors' notes and the lab test results.*

Task 5.2 Conversion of the Data: *Doctors' notes will be processed using a lexical analyzer and medication information is converted to binary data. Columns with missing values will be taken out to make data ready for the machine learning model. A final dataset will be obtained at the end of this process.*

Deliverables

D5.1: *Diabetaid Patient Dataset (Apr 24, 2025)*

WP 6: Model Training and Improvement

Start date: Dec 20, 2024 **End date:** 2nd week of May

Leader:	*Osman*	**Members involved:**	*Çağatay, Yamaç*

Objectives: After obtaining some ready data, trying out some machine learning algorithms on the data. Finding out an algorithm that best fits the data. Looking for a reputable and accurate way to explain treatment decisions of the algorithm.

<table>
<tr><td colspan="4">Tasks:

Task 6.1 Choosing the Best Machine Learning Model: Some machine learning models, mainly decision trees will be trained using the prepared patient dataset. The model that returns best possible treatment options will be chosen as the machine learning model.

Task 6.2 Improving the Model to Get Better Results: Best choice of hyperparameters will be found to further improve the accuracy of the model.

Task 6.3 Explanation of the Decisions: Search for methods of explaining model decisions and try to create a model that can accurately explain why that decision is made.

Task 6.4 Incorporating the Model into the Application: Integrate the model into the application to have a functioning final version for the demo.</td></tr>
<tr><td colspan="4">Deliverables

D6.1: Explainable Model for Diabetes Treatment Decision (2nd week of May)</td></tr>
<tr><td colspan="4">WP 7: Discussions with Stakeholders</td></tr>
<tr><td colspan="4">Start date: Dec 20, 2024 End date: 2nd week of May</td></tr>
<tr><td>Leader:</td><td>Sude</td><td>Members involved:</td><td>Çağatay</td></tr>
<tr><td colspan="4">Objectives: This project will be conducted in collaboration with doctors from Bilkent City Hospital. Correction of the treatment decisions and explanations, preparation of dataset, types of possible treatments, usability and acceptability of the application are topics that will be discussed with stakeholders to enhance the quality of the product.</td></tr>
<tr><td colspan="4">Tasks:

Task 7.1 Paying Frequent Visits to the Hospital: Hospital will be visited to have meetings with the doctors and questions will be asked to further move the project towards the needs and expectations of the target users.</td></tr>
<tr><td colspan="4">Deliverables

D7.1: Project Demo (3rd week of May)</td></tr>
</table>

Table 3: Project Plan.

Gantt Chart:

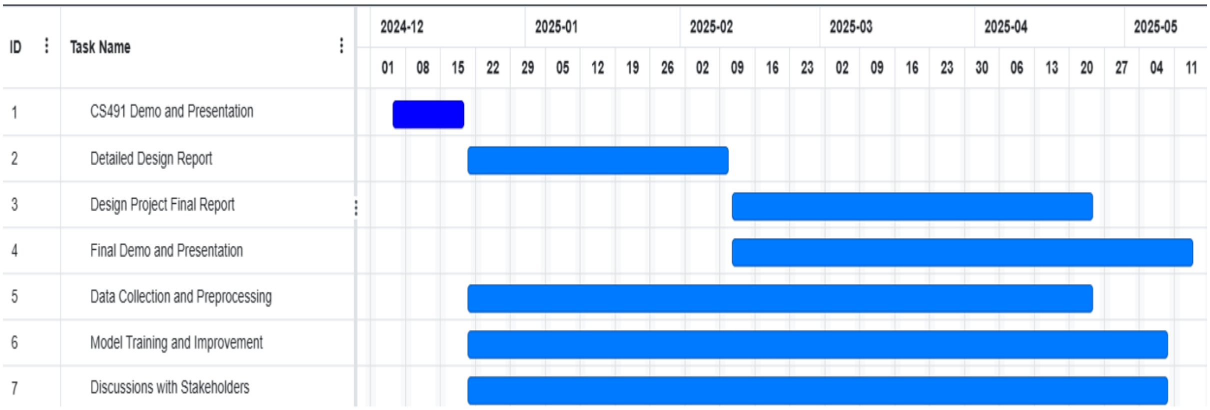

Figure 15: Gantt Chart.

4.4 Ensuring Proper Teamwork

Proper Work Allocation:

Frequent meetings (once or twice a week) are held to distribute workload to ensure every member of the team is assigned equal amounts of work. For some large tasks, two or three people are assigned for better distribution of workload. To ensure proper work allocation and workload distribution, Jira software helps massively by providing frameworks for agile project development.

Scrum with Jira:

Jira is a project management tool for software development teams to track their progress and supports scrum framework. Scrum is an agile project management strategy that is commonly used by professional software development teams [12]. For this project, one of the members of the team is assigned as the scrum master and each sprint is planned and executed under control of the scrum master. This makes the team as a whole more efficient and productive throughout the development process of the project.

Git and Github:

Github is a version control platform which allows members of the project to work on the same codebase at the same time without compatibility issues or conflicts [13]. Git is the version control system that makes it happen and changes in files in Github are tracked by Git. Github also makes reviewing code and making suggestions much easier. Using Git and Github makes it possible for the team members to collaborate on the project codebase at the same time.

4.5 Ethics and Professional Responsibilities

- The source code of DiabetAid will be maintained on GitHub privately and is only accessible to the members of the project team and the supervisors authorized. The source code will not be shared with any third party without explicit approval until the project is completed.
- Software frameworks and libraries used are TensorFlow, Scikit-learn, and LIME. Software licensing will be strictly complied with, and appropriate citation is done to honor the licensing terms of the open source or proprietary software being utilized.
- The use of data from patients has been approved by the ethics committee of Ankara Bilkent City Hospital. Access to data about patients will be strictly done with ethical consideration, and no data will be shared outside the project scope without explicit permission.
- Regular weekly team meetings will be held every Thursday to review the progress, discuss difficulties, and confirm that developmental goals are agreed upon. Participants will include all team members, our supervisor, and are to be conducted face to face or virtually as needed.
- All patient data accessed through the hospital's VPN will be anonymized and securely stored during processing. Data will be deleted immediately after use to prevent unauthorized access or misuse.
- Consultations with doctors and hospital staff will be carried out on a regular basis during the project to validate system recommendations for conformance with clinical needs. In this context, the feedback of all relevant stakeholders will be incorporated into the development process.
- The development will be based on the relevant standards like KVKK, ISO/IEC 27001, and ISO 14971, which indicate compliance with data protection, security, and risk management.
- Full documentation will be kept for the source code, system models, and progress on the project for maximum transparency and accountability from a project's life cycle.

4.6 Planning for New Knowledge and Learning Strategies

In the DiabetAid project, there are many areas where knowledge and skills need to be developed to ensure the success of the entire system. Though the team holds basic knowledge about machine learning and software development, application in a healthcare-oriented project requires further exploration and study. Below is an overview of how the team aims to develop new knowledge by implementing the proposed learning strategy.

Machine Learning for Health Care Applications:

While our team has experience in machine learning, we would have to dive deeper into those particular applications in the medical domain. We will explore relevant academic papers such as those focused on treatment optimization, explainability in AI (XAI), and models for clinical decision support systems (CDSS). Moreover, we will study how to technically implement supervised and explainable machine learning models, such as decision trees, neural networks, and LIME (Local Interpretable Model-Agnostic Explanations).

Training and Fine-tuning Machine Learning Models:

The team will train and fine-tune machine learning models using anonymized patient data to give accurate recommendations. This also encompasses the study of techniques to optimize model performance, including hyperparameter tuning, cross-validation, and data preprocessing.

Explainable AI:

Since explainability is one of the main parts of DiabetAid, the team will be working on how to implement XAI techniques. We will study frameworks such as LIME to generate clear, interpretable explanations of model outputs. This includes experimenting with these tools and adapting them to our specific use case.

Medical Knowledge Acquisition:

We will be working closely with doctors from Ankara Bilkent City Hospital to ensure that the system's suggestions are in line with real-world clinical practices. We will interview health professionals to understand their workflow, the specific challenges faced in managing diabetes, and expectations from a clinical decision support system. We will also study literature and medical guidelines about diabetes treatment.

Data Privacy and Ethical Practices:

The information related to patients is sensitive, and the team needs to be aware of KVKK (Turkey's Personal Data Protection Law), GDPR (General Data Protection Regulation), and other relevant data security standards. This would include studying secure data handling practices, anonymization techniques, and encryption protocols for full compliance.

Integration with Hospital Systems:

The team has to be trained on how to integrate the system with existing hospital infrastructure, such as EHRs. For this, we will study HL7 standards for healthcare interoperability and research how these systems work in real-world environments.

Software Development and Deployment Tools:

We will be leveraging frameworks such as TensorFlow, Scikit-learn, Flask, and React in building the system. Specific time will be utilized to understand the particular functionalities that each tool can offer toward the development of scalable, efficient, and user-friendly applications.

User Interface Design and Usability Testing:

Since DiabetAid will be used by doctors of all different technical expertise, the user interface has to be intuitive and highly efficient. The team is going to learn the principles of usability design and the interface will be reviewed and tested in person with doctors to ensure that it will meet the needs, remain user-friendly, and align with their workflow for effective clinical use.

Healthcare Industry Standards and Regulations:

The team will review healthcare industry standards, including ISO/IEC 27001 on information security, ISO 14971 on risk management, and ISO 13485 on quality management of medical devices.

Collaboration with Experts and Stakeholders:

Regular meetings with the project supervisor, hospital doctors, and innovation experts will be carried out to acquire insights and to keep the system in line with medical and technical expectations. Their feedback will be incorporated at each stage of the development.

In this way, the team intends to develop structured knowledge leading to the creation of a robust, reliable, and ethical healthcare solution in DiabetAid. This planning will ensure that all technical, medical, and ethical aspects are covered, thus enabling the project to meet its objectives effectively.

5. Glossary

Accuracy: A metric used to measure how often the machine learning model's predictions are correct.

BMI (Body Mass Index): A numerical value derived from an individual's weight and height to assess whether they are underweight, normal weight, overweight, or obese.

Clinical Decision Support System (CDSS): A software system designed to assist healthcare professionals in making clinical decisions by analyzing patient data and providing evidence-based recommendations.

CSV (Comma-Separated Values): A file format that stores tabular data in plain text, with each row separated by a newline and each column separated by a comma.

EHR (Electronic Health Record): A digital version of a patient’s paper chart, containing medical history, diagnoses, medications, and treatment plans.

Explainable AI (XAI): A type of AI that provides clear explanations of how and why it arrives at specific outputs or decisions.

Hyperglycemia: A condition characterized by abnormally high blood sugar levels.

KVKK (Personal Data Protection Law): Turkish legislation that governs the protection and processing of personal data.

LIME (Local Interpretable Model-Agnostic Explanations): A tool used to explain predictions made by machine learning models.

Precision: A metric that measures the accuracy of positive predictions made by a machine learning model.

React: React is a JavaScript library for building user interfaces, focusing on creating reusable components.

Flask: Flask is a lightweight Python web framework used to build web applications quickly and easily.

Scikit-learn: A Python library for building and evaluating machine learning models.

TensorFlow: An open-source machine learning library used for building and training ML models.

VPN (Virtual Private Network): A secure connection used to access remote systems and protect data during transmission.

6. References

[1] “Diabetes,” World Health Organization, [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/diabetes.

[2] “6698 Sayılı Kişisel Verilerin Korunması Kanunu (KVKK),“ www.mevzuat.gov.tr/mevzuat?MevzuatNo=6698&MevzuatTur=1&MevzuatTertip=5. Accessed 21 Nov. 2024.

[3] “ISO/IEC 27001:2022,” ISO, https://www.iso.org/standard/27001 (accessed Nov. 21, 2024).

[4] “ISO 9241-210:2019,” ISO, https://www.iso.org/standard/77520.html (accessed Nov. 21, 2024).

[5] “ISO 14971:2019,” ISO, https://www.iso.org/standard/72704.html (accessed Nov. 21, 2024).

[6] “IEC 62304:2006,” ISO, https://www.iso.org/standard/38421.html (accessed Nov. 21, 2024).

[7] “IEEE SA - IEEE Standard for Health Informatics - device interoperability - part 10701: Point-of-care medical device communication - metric provisioning by participants in a service-oriented device connectivity (SDC) system,” IEEE Standards Association, https://standards.ieee.org/ieee/11073-10701/7538/ (accessed Nov. 21, 2024).

[8] Health Level Seven international - homepage | HL7 international, https://www.hl7.org/ (accessed Nov. 21, 2024).

[9] “ISO 13485:2016,” ISO, https://www.iso.org/standard/59752.html (accessed Nov. 21, 2024).

[10] J. Perez, “Unpacking open source compliance,” OpenLogic by Perforce, https://www.openlogic.com/blog/open-source-compliance-overview (accessed Nov. 21, 2024).

[11] “What is SSL/TLS encryption?,” F5, Inc., https://www.f5.com/glossary/ssl-tls-encryption (accessed Nov. 21, 2024).

[12] “What is JIRA Software? - GeeksforGeeks.” Accessed: Dec. 15, 2024. [Online]. Available: https://www.geeksforgeeks.org/what-is-jira-software/

[13] “About GitHub and Git - GitHub Docs.” Accessed: Dec. 15, 2024. [Online]. Available: https://docs.github.com/en/get-started/start-your-journey/about-github-and-git