**[Title of the Project]**

|  |
| --- |
|  |

**By:**

**Mudabbir Ahmed**

**35830**

**Saad Munaf**

**38748**

**Saad Bin Rizwan**

**37968**

**Supervised by:**

**[Tjamal Shahzad]**

**Faculty of Computing**

**Riphah International University, Islamabad**

**Spring/Fall 2025**

**A Dissertation Submitted To**

**Faculty of Computing,**

**Riphah International University, Islamabad**

**As a Partial Fulfillment of the Requirement for the Award of the Degree of**

**Bachelors of Science in Computer Science**

**Faculty of Computing**

**Riphah International University, Islamabad**

Date: [date of final presentation]

**Final Approval**

This is to certify that we have read the report submitted by ***Mudabbir Ahmed (35830), Saad Munaf (3878) and Saad Bin Rizwan (37968)*** for the partial fulfillment of the requirements for the degree of the Bachelors of Science in Computer Science (BSCS). It is our judgment that this report is of sufficient standard to warrant its acceptance by Riphah International University, Islamabad for the degree of Bachelors of Science in Computer Science (BSCS).

**Committee:**

|  |  |
| --- | --- |
| **1** | [Tajamal Shahzad]  (Supervisor) |
|  |  |
| **2** | [Dr.Musharraf Ahmed]  (Head of Department/chairman) |

**Declaration**

We hereby declare that this document “**[Diabeto-Vision]**” neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this project with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of our teachers, especially our supervisor **[Tajamal Shahzad]**. If any part of the system is proved to be copied out from any source or found to be reproduction of any project from anywhere else, we shall stand by the consequences.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**[Mudabbir Ahmed]**

**[35830]**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**[Saad Munaf]**

**[38748]**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**[Saad Bin Rizwan]**

**[37968]**

**Dedication**

Our project is dedicated to our work to our parents, seniors, friends and our supervisor “Tajamal Shahzad” who has been our continual source of inspiration and whose support has helped this project succeed. This project would not have possible without their trust and support.

**Acknowledgement**

First of all we are obliged to Allah Almighty the Merciful, the Beneficent and the source of all Knowledge, for granting us the courage and knowledge to complete this Project.

We owe a heartfelt thank you to our project supervisor, “Prof. Tajamul Shahzad”. His guidance has been a beacon of light throughout our project journey. His patience and knowledge were key in overcoming the challenges we faced. We are truly thankful for his dedication and the time he invested in us. We also extend our deepest gratitude to our parents and family. Their unwavering belief in us and the values of hard work and integrity they have nurtured within us have been our guiding stars. It is with their blessings and constant encouragement that we have been able to achieve this milestone

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**[Mudabbir Ahmed]**

**[35830]**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**[Saad Munaf]**

**[38748]**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**[Saad Bin Rizwan]**

**[37968]**

**Abstract**

Diabetic Retinopathy is a serious complication of diabetes that affects the eyes, potentially leading to vision loss if left untreated. Early detection and accurate grading are crucial for timely treatment and better outcomes.

**Diabeto Vision** is a web application developed to offer assistance in scanning and determining the degree of severity of diabetic retinopathy using the latest machine learning algorithms. The system retrieves the fundus images of the eye and analyses them to see if the patient suffers from the problem and if yes, the level of severity is established. This easy to use platform seeks to give both patients and health care providers a fast and accurate method of detecting and coordinating the treatment of Diabetic Retinopathy.

Table of Contents

[Table of Contents i](#_Toc113957278)

[List of Tables iii](#_Toc113957279)

[List of Figures iv](#_Toc113957280)

[Abstract 1](#_Toc113957281)

[Chapter 1: Introduction 2](#_Toc113957282)

[1.1 Goals and Objectives 2](#_Toc113957283)

[1.2 Scope of the Project 2](#_Toc113957284)

[Chapter 2: Literature Review 3](#_Toc113957285)

[2.1 Introduction 3](#_Toc113957286)

[2.2 Background and Problem Elaboration 3](#_Toc113957287)

[2.3 Detailed Literature Review 3](#_Toc113957288)

[2.3.1 Definitions 3](#_Toc113957289)

[2.3.2 Related Research Work 1 3](#_Toc113957290)

[2.3.3 Related Research Work 2 3](#_Toc113957291)

[2.4 Literature Review Summary Table 3](#_Toc113957292)

[2.5 Research Gap 3](#_Toc113957293)

[2.6 Problem Statement 3](#_Toc113957294)

[Chapter 3: Requirements and Design 4](#_Toc113957295)

[3.1 Requirements 4](#_Toc113957296)

[3.1.1 Functional Requirements 4](#_Toc113957297)

[3.1.2 Non-Functional Requirements 4](#_Toc113957298)

[3.1.3 Hardware and Software Requirements 4](#_Toc113957299)

[3.2 Proposed Methodology 4](#_Toc113957300)

[3.3 System Architecture 4](#_Toc113957301)

[3.4 Use Cases 4](#_Toc113957302)

[3.4.1 Sample Use Case Name Here 4](#_Toc113957303)

[3.5 Database Design *(Optional)* 6](#_Toc113957304)

[3.6 Class diagram *(Optional)* 6](#_Toc113957305)

[3.7 Sequence diagram *(Optional)* 6](#_Toc113957306)

[3.8 Any Other Artifact… 6](#_Toc113957307)

[3.9 Graphical User Interfaces(GUI) (*Optional)* 6](#_Toc113957308)

[Chapter 4: Implementation and Test Cases 7](#_Toc113957309)

[4.1 Implementation 7](#_Toc113957310)

[4.1.1 Implementation of First Component/Algorithm 7](#_Toc113957311)

[4.2 Test case Design and description 7](#_Toc113957312)

[4.2.1 Sample Test case No.1 7](#_Toc113957313)

[4.2.2 Sample Test case No.2 7](#_Toc113957314)

[4.3 Test Metrics 8](#_Toc113957315)

[4.3.1 Sample Test case Matric.No.1 8](#_Toc113957316)

[4.3.2 Sample Test case Metric.No.2 8](#_Toc113957317)

[4.3.3 Sample Test case Metric.No.3 8](#_Toc113957318)

[Chapter 5: Experimental Results and Analysis 9](#_Toc113957319)

[Chapter 6: Conclusion and Future Directions 10](#_Toc113957320)

[References 11](#_Toc113957321)

[Appendix 12](#_Toc113957322)

[Appendix A: Guidelines 12](#_Toc113957323)

[Appendix B: Heading of Sample Appendix B 12](#_Toc113957324)

[Formatting Guidelines 13](#_Toc113957325)

[Chapter 1: Heading 1 13](#_Toc113957326)

[1.1 Heading 2 13](#_Toc113957327)

[1.1.1 Heading 3 13](#_Toc113957328)

[Tables and Figures 14](#_Toc113957329)

[Equations 15](#_Toc113957330)

[Header/Footer 15](#_Toc113957331)

[Other Formatting Guidelines 15](#_Toc113957332)

[References 15](#_Toc113957333)

List of Tables

[Table 1: This is Sample table caption 1](#_Toc55559466)

[Table 2: This is Sample table caption 1](#_Toc55559467)

List of Figures

Figure 1: List of Styles 1

Figure 2: IEEE Reference style 1

# Abstract

Diabetic Retinopathy is a serious complication of diabetes that affects the eyes, potentially leading to vision loss if left untreated. Early detection and accurate grading are crucial for timely treatment and better outcomes.

**Diabeto Vision** is a web application developed to offer assistance in scanning and determining the degree of severity of diabetic retinopathy using the latest machine learning algorithms. The system retrieves the fundus images of the eye and analyses them to see if the patient suffers from the problem and if yes, the level of severity is established. This easy to use platform seeks to give both patients and health care providers a fast and accurate method of detecting and coordinating the treatment of Diabetic Retinopathy.

# Chapter 1: Introduction

According to the WHO, the number of visually impaired people worldwide is estimated to be 2.2 billion, of whom at least 1 billion have vision impairment could have been prevented or are yet to be addressed. The world faces considerable challenges in terms of eye care, including inequalities in the coverage and quality of prevention, treatment, and forestall of rehabilitation services. Early detection and diagnosis of ocular pathologies would enable forestall visual impairment. The traditional diagnosis systems are slow, time-consuming, and expensive require a certain level of expertise to use, whereas the proposed system will provide an easy-to- use, reliable, fast, and cheap alternative for the users. It will be a web-based project which will integrate image-processing techniques. Medical professionals can also benefit from the system, as it will enable them to verify the results from conventional systems. The users are required to input fundus and retinal photographs of their eyes, and the system will preprocess them, extract features, and make a diagnosis based on the available datasets.

## 1.1 Goals and Objectives

#### Goals

* Design a user friendly React based Frontend.
* Implement secure logging and registration system.
* Allow user to upload fundus image for analysis.
* Integrate trained ML Model for detection and severity, grading.
* Allow Doctor to chat with the patient

#### Objectives

* + - Develop a web-based platform for detecting and grading Diabetic Retinopathy
    - Utilize Machine Learning model to analyze fundus image for accurate diagnosis

# 1.2 Scope of the Project

**Healthcare Support**: Helps doctors and patients detect diabetic retinopathy early and understand its severity

**Web-Based Access:** Users can access the platform from anywhere to upload fundus images and get results.

**Machine Learning Powered:** Uses trained ML models to provide accurate and reliable diagnoses.

**User-Friendly Design:** Simple and intuitive interface, making it easy for anyone to use.

**Data Security:** Ensures that all user data and images are kept private and secure.

**Scalable and Future-Ready:** Designed to handle more users and datasets as it grows.

**Educational Purpose:** Useful for medical research, training, and learning about diabetic retinopathy.

## 

## Chapter 2: Literature Review:

## 2.1 Introduction

Diabetic Retinopathy (DR) is one of the leading causes of vision impairment globally, with an increasing prevalence due to the rise in diabetes cases. The detection of DR in its early stages is crucial to prevent vision loss. This report explores various CNN architectures and methodologies to automate DR detection using retinal images, enhancing accuracy, efficiency, and scalability.

## 2.2 Background and Problem Elaboration

Traditional DR diagnosis involves manual examination of retinal fundus images by ophthalmologists. However, this process is time-consuming, prone to human error, and dependent on specialized skills. The advent of Convolutional Neural Networks (CNNs) has revolutionized image recognition tasks, offering a promising solution for automating DR detection. CNNs can identify complex patterns in retinal images, enabling early detection and classification of DR stages with higher precision and speed.

## 2.3 Detailed Literature Review

Contributions of various studies aimed at improving DR detection through advanced deep learning methods. It focuses on different CNN architectures, preprocessing techniques, datasets, and their performance metrics

### 2.3.1 Related Research Work 1:

### Related Research Work 2:

## Literature Review Summary Table

The columns in the table depend upon your problem and should be specific to your project.

Table 1: History of Computing Devices

The summary of various computing devices invented in the past from 1833-1901 is presented here.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Name, reference | Inventor | Year | Input | Output | Description |
| 1. | Analytical Engine, [1] | Charles Babbage | 1833 | Punch cards | Printer, curve plotter, bell | First general purpose computer that had an arithmetical logic unit and could compute using conditional branching and loops. Also incorporated integrated memory. |

## 

## 2.3 Research Gap

Despite the advancements, several gaps persist in DR detection research:

• Difficulty in handling highly imbalanced datasets.

• Limited real-time applications due to high computational costs.

• Challenges in generalizing models across diverse datasets. 98.50%

• Lack of robust systems for early detection in resource-constrained settings.

## 2.4 Problem Statement

Manual diagnosis of diabetic retinopathy is inefficient, error-prone, and lacks scalability. Automated CNN-based systems address these challenges but face issues such as imbalanced datasets, computational limitations, and generalizability. This project aims to develop an efficient, accurate, and scalable CNN-based solution for DR detection, bridging these gaps and improving accessibility in diverse healthcare environments

# Chapter 3: Requirements and Design

# 3.1 Requirements

## 3.1.1 Functional Requirements

**3.1.1.1 FR-01: Admin**

|  |  |
| --- | --- |
| **ID** | **Requirement** |
| FR-1.1 | Admin can login account. |
| FR-1.2 | Admin can be able to add patients. |
| FR-1.3 | Admin can be able to add doctors. |
| FR-1.4 | Admin can view patients. |
| FR-1.5 | Admin can view results. |
| FR-1.6 | Admin can delete patients. |
| FR-1.7 | Admin can delete doctors |
| FR-1.8 | Admin can view patient result. |
| FR-1.9 | Admin can be able to not approve doctor. |
| FR-1.10 | Admin can be able to logout |

**3.1.1.2 FR-02: System**

|  |  |
| --- | --- |
| **ID** | **Requirement** |
| FR-2.1 | System will analyze the image. |
| FR-2.2 | System can be able to provide the results of the disease |
| FR-2.3 | System can be able to show the accuracy of the disease. |

**3.1.1.3 FR-03: Patient**

|  |  |
| --- | --- |
| **ID** | **Requirement** |
| FR-3.1 | Patient can be able to sign up the account. |
| FR-3.2 | Patient can be able to login account |
| FR-3.3 | Patient can able to search for available doctor. |
| FR-3.4 | Patient can detect Diabetic Retinopathy by uploading his/her retina picture. |
| FR-3.5 | Patient can be able to view result. |
| FR-3.6 | Patient can do chat with doctor. |
| FR-3.7 |  |

**3.1.1.4 FR-04: Doctor**

|  |  |
| --- | --- |
| **ID** | **Requirement** |
| FR – 1.1 | Doctor can sign up for his/her account. |
| FR – 1.2 | Doctor can be able to login his/her account. |
| FR – 1.3 | Doctor can see report of the patient. |
| FR – 1.4 | Doctor can do chat with patient. |

### 3.1.2 Non-Functional Requirements

The deep learning model shall utilize datasets like Aptos and IDRiD to achieve high accuracy in DR detection.

• The system shall maintain a response time under 5 seconds for image analysis.

• MongoDB shall be used to ensure efficient and secure storage of patient and image data.

• The web interface shall be intuitive and user-friendly, enabling easy upload and result retrieval.

### 3.2.1 Hardware and Software Requirements

Training our model required high end systems for smooth processing.

**3.2.1.1 Hardware requirements:**

* GPU: Our group uses the Colab Pro (GPU) that is given by the colab for fast training of the model.
* We have to take picture from the fundoscopy camera and then upload this picture to our system and get results

**3.2.1.2 Software Requirements:**

The complete system will be built by using the following tech stack:

• Programming Language: Python (for model training and backend APIs).

• Frameworks: TensorFlow/Keras for deep learning model development.

• Libraries: OpenCV for image processing; Scikit-learn for evaluation metrics.

• Web Technologies: React, CSS for frontend development, Flask/Python for backend integration.

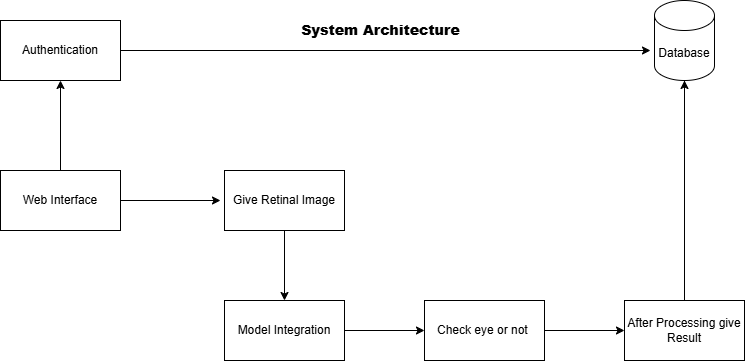
• Database: MongoDB for storing patient data and results.

• Development Environment: Google Colab and Jupyter Notebooks for model training and experimentation.

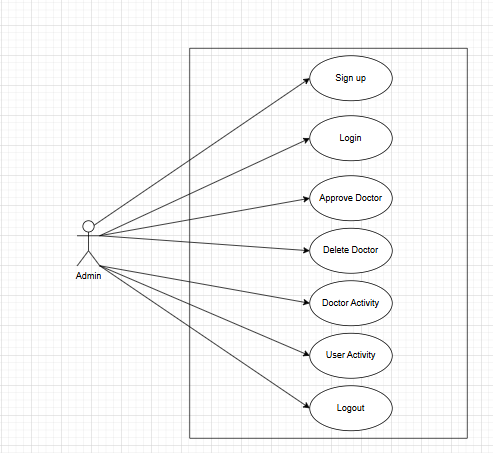
## 3.3 Proposed Methodology:

The Diabetic Retinopathy Detection System is a machine learning-based solution that uses a trained deep learning model to identify the presence and stage of diabetic retinopathy from retinal images. The system is accessible via a web application, making it user-friendly and scalable.

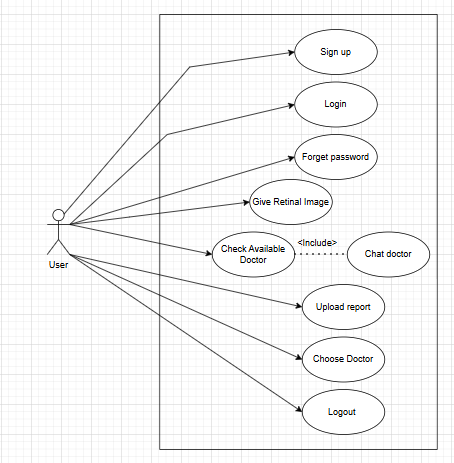
## 3.4 System Architecture:



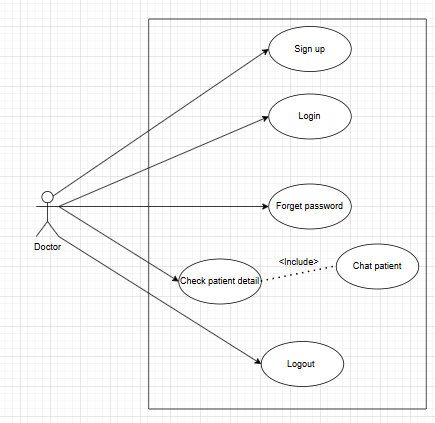
## 3.5 Use Cases

**3.5.1 Admin Use case:**

**3.5.2 Patient Use Case:**



### Doctor Use Case:



### 3.6.1 Sample Use Case Name Here

**3.6.1.1 Admin login:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Admin Login | | |
| Actors | | Admin | | |
| Summary | | Shows the step and interaction involved when an administrator logs into the system. | | |
| Pre-Conditions | | The system is running. | | |
| Post-Conditions | | Admin gets access to the system | | |
| Special Requirements | | None | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | The admin clicks on the "Forget Password" link. | | 2 | The system prompts for the registered email. |
| 3 | The admin enters a valid registered email. | | 4 | The system sends a password reset link or a temporary password to the email. |
| **Alternative Flow** | | | | |
| 3 | The admin enters an unregistered email. | | 4-A | The system displays: **"Email not found."** |

**3.6.1.2 Approve Doctor:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Approve Doctor | | |
| Actors | | Admin | | |
| Summary | | Admin approves newly registered doctors before they can access the system. | | |
| Pre-Conditions | | Doctors must have signed up and are pending approval. | | |
| Post-Conditions | | The selected doctor’s status is updated to “Approved.”. | | |
| Special Requirements | | None | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | The admin navigates to the list of pending doctors. | | 2 | The system displays doctors awaiting approval. |
| 3 | The admin clicks “Approve” on a doctor. | | 4 | The system updates the doctor’s status to approve. |
| **Alternative Flow** | | | | |
| 3-A | The admin attempts to approve a doctor who doesn’t exist. | | 4-A | The system displays: **"Doctor not found or already approved."** |

**3.6.1.3 Delete Doctor:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Delete Doctor | | |
| Actors | | Admin | | |
| Summary | | Admin deletes a doctor’s account from the system. | | |
| Pre-Conditions | | The doctor must exist in the system. | | |
| Post-Conditions | | The selected doctor’s data is permanently removed or deactivated. | | |
| Special Requirements | | None | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | The admin navigates to the doctor management section. | | 2 | The system displays a list of registered doctors. |
| 3 | The admin selects a doctor and clicks “Delete.” | | 4 | The system removes the doctor’s data. |
| **Alternative Flow** | | | | |
| 3-A | The admin selects a non-existent doctor. | | 4-A | The system displays: **"Doctor record not found."** |

**3.6.1.4 Doctor Activity:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Doctor Activity | | |
| Actors | | Admin | | |
| Summary | | Admin views a log of doctor-related activities like appointments, edits, or logins. | | |
| Pre-Conditions | | Doctors must have interacted with the system | | |
| Post-Conditions | | Activity log is shown. | | |
| Special Requirements | | None | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | The admin navigates to the “Doctor Activity” page. | | 2 | The system fetches and displays activity logs. |
| **Alternative Flow** | | | | |
| 3-A | No activity is recorded. | | 4-A | The system shows: **"No activity available."** |

**3.6.1.5 Logout:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Logout | | |
| Actors | | Admin | | |
| Summary | | Admin ends their current session. | | |
| Pre-Conditions | | Admin must be logged in. | | |
| Post-Conditions | | Session is terminated and the user is redirected to the login page. | | |
| Special Requirements | | None | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | The admin clicks the “Logout” button. | | 2 | The system ends the session and redirects to the login screen. |
| **Alternative Flow** | | | | |
| 3-A | None | | 4-A | None |

**3.6.2 Doctor:**

**3.6.2.1 Doctor sign up:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Sign up | | |
| Actors | | Doctor | | |
| Summary | | A doctor creates an account by providing registration details. | | |
| Pre-Conditions | | The doctor must not already be registered. | | |
| Post-Conditions | | A registration request is submitted and awaits admin approval. | | |
| Special Requirements | | None | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | Doctor opens the sign-up page. | | 2 | System displays a form for registration. |
| 3 | Doctor fills in all required information. | | 4 | System stores the data and notifies admin for approval. |
| **Alternative Flow** | | | | |
| 3-A | Doctor submits incomplete or invalid data. | | 4-A | System shows: **"Please fill all required fields correctly."** |

**3.6.2.2 Login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Login | | |
| Actors | | Doctor | | |
| Summary | | A doctor logs into the system using valid credentials. | | |
| Pre-Conditions | | Doctor must be approved by the admin and not already logged in. | | |
| Post-Conditions | | A session is created and the doctor is redirected to the dashboard. | | |
| Special Requirements | | None | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | Doctor opens the Login page | | 2 | System displays Login form |
| 3 | Doctor fills valid credentials. | | 4 | System authenticates and logs in the doctor. |
| **Alternative Flow** | | | | |
| 3-A | Doctor enters invalid credentials. | | 4-A | System displays: **"Incorrect email or password."** |

**3.6.2.3 Forget Password:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Forget Password | | |
| Actors | | Doctor | | |
| Summary | | Allows the doctor to reset their password if forgotten. | | |
| Pre-Conditions | | Doctor must be registered in the system. | | |
| Post-Conditions | | A password reset email/link is sent. | | |
| Special Requirements | | Email must match existing records. | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | Doctor clicks "Forget Password" | | 2 | System asks for the registered email. |
| 3 | Doctor provides email. | | 4 | System sends reset link to the email. |
| **Alternative Flow** | | | | |
| 3-A | Email is not found. | | 4-A | System displays: **"Email not registered."** |

**3.6.2.4 Check Patient detail:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Check Patient Detail. | | |
| Actors | | Doctor | | |
| Summary | | Doctor views patient's profile and medical history. | | |
| Pre-Conditions | | Doctor must be logged in and have access to the patient list. | | |
| Post-Conditions | | A password reset email/link is sent. | | |
| Special Requirements | | Data must be securely handled | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | Doctor selects a patient from the list. | | 2 | System retrieves and displays patient's details. |
| **Alternative Flow** | | | | |
| 3-A | Patient record is missing or inaccessible. | | 4-A | System shows: **"Patient details not available."** |

**3.6.2.5 Chat Patient (Include in Check Patient Detail)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Chat Patient | | |
| Actors | | Doctor | | |
| Summary | | Enables real time chat with Doctor and Patient. | | |
| Pre-Conditions | | Doctor must be viewing a patient’s details. | | |
| Post-Conditions | | A secure chat session is established. | | |
| Special Requirements | | Live chat module or integration required. | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | Doctor clicks “Chat” inside patient detail. | | 2 | System opens a secure chat window with the patient.. |
| **Alternative Flow** | | | | |
| 3-A | Patient is offline. | | 4-A | System shows: **"Patient not available for chat."** |

**3.6.2.6 Logout:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Logout | | |
| Actors | | Doctor | | |
| Summary | | Ends the doctor’s session in the system. | | |
| Pre-Conditions | | Doctor must be logged in. | | |
| Post-Conditions | | Session ends and doctor is redirected to the login page. | | |
| Special Requirements | | None. | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | Doctor clicks on “Logout.” | | 2 | System logs out the doctor and redirects to login. |
| **Alternative Flow** | | | | |
| 3-A | None | | 4-A | None |

**3.6.3 User/Patient:**

**3.6.3.1 User/Patient sign up:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Actors | | Users / Patient | | |
| Summary | | A user creates an account by providing registration details. | | |
| Pre-Conditions | | User must not already be registered. | | |
| Post-Conditions | | Account is created and saved in the system. | | |
| Special Requirements | | Valid information must be entered. | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | User opens the sign-up page. | | 2 | System displays the sign-up form.. |
| **Alternative Flow** | | | | |
| 3-A | User submits incomplete or invalid info. | | 4-A | System shows: **"Please provide valid information."** |

**3.6.3.2 Login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Login | | |
| Actors | | Users | | |
| Summary | | A user logs into the system using valid credentials. | | |
| Pre-Conditions | | User must be registered | | |
| Post-Conditions | | User session is established and redirected to dashboard. | | |
| Special Requirements | | None | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | User opens login page. | | 2 | System displays login form. |
| 3 | User enters credentials. | | 4 | System authenticates and logs in the user. |
| **Alternative Flow** | | | | |
| 3-A | Credentials are invalid. | | 4-A | System displays: **"Incorrect email or password."** |

**3.6.3.3 Forget Password:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Forget Password | | |
| Actors | | Users | | |
| Summary | | Allows users to reset password if forgotten. | | |
| Pre-Conditions | | Email must be registered. | | |
| Post-Conditions | | A reset link is sent to the user’s email. | | |
| Special Requirements | | Email Verification. | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | User clicks "Forget Password". | | 2 | System asks for registered email. |
| 3 | User submits email. | | 4 | System sends a password reset link. |
| **Alternative Flow** | | | | |
| 3-A | Email doesn’t exist | | 4-A | System shows: **"Email not registered."** |

**3.6.3.4 Give Retinal Image:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Give Retinal Image | | |
| Actors | | User | | |
| Summary | | Users upload a retinal image to be analyzed. | | |
| Pre-Conditions | | User must be logged in. | | |
| Post-Conditions | | Retinal image is uploaded for analysis | | |
| Special Requirements | | Image must be in supported format. | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | User clicks “Give Retinal Image”. | | 2 | System prompts to upload image. |
| 3 | User uploads retinal image. | | 4 | System stores the image and begins analysis. |
| **Alternative Flow** | | | | |
| 3-A | Image format is unsupported | | 4-A | System shows: **"Invalid image format."** |

**3.6.3.5 Check Available Doctor:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Check Available Doctor | | |
| Actors | | User | | |
| Summary | | Shows list of doctors available for consultation. | | |
| Pre-Conditions | | User must be logged in. | | |
| Post-Conditions | | System displays list of available doctors. | | |
| Special Requirements | | None | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | User selects “Check Available Doctor”. | | 2 | System displays real-time list of available doctors. |
| **Alternative Flow** | | | | |
| 3-A | No Doctors available. | | 4-A | System shows: **"No Doctors currently available."** |

**3.6.3.6 Chat Doctor:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Chat Doctor | | |
| Actors | | User | | |
| Summary | | User chats with a doctor in real-time | | |
| Pre-Conditions | | Doctor must be available and selected. | | |
| Post-Conditions | | Secure chat is established between user and doctor. | | |
| Special Requirements | | None | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | User selects an available doctor and clicks “Chat”. | | 2 | System opens chat window with the doctor |
| **Alternative Flow** | | | | |
| 3-A | No Doctors available. | | 4-A | System shows: **"No Doctors currently available."** |

**3.6.3.7 Upload report:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Upload Report | | |
| Actors | | User | | |
| Summary | | User uploads medical reports for doctor review. | | |
| Pre-Conditions | | User must be logged in. | | |
| Post-Conditions | | Report is uploaded and stored securely. | | |
| Special Requirements | | Allowed file types (PDF, images). | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | User selects “Upload Report”. | | 2 | System prompts to choose a file. |
| 3 | User uploads report. | | 4 | System confirms upload success. |
| **Alternative Flow** | | | | |
| 3-A | File is not accepted format. | | 4-A | System shows: **"Invalid report file."** |

**3.6.3.8 Choose doctor:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Choose | | |
| Actors | | User | | |
| Summary | | User selects a doctor for treatment or follow-up. | | |
| Pre-Conditions | | List of doctors must be available. | | |
| Post-Conditions | | Selected doctor is saved in user profile. | | |
| Special Requirements | | Doctor ID and Rating. | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | User views list of doctors. | | 2 | System displays all available options. |
| 3 | User selects preferred doctor. | | 4 | System links user to the selected doctor. |
| **Alternative Flow** | | | | |
| 3-A | Doctor is no longer available. | | 4-A | System shows: **"Doctor unavailable, please choose another."** |

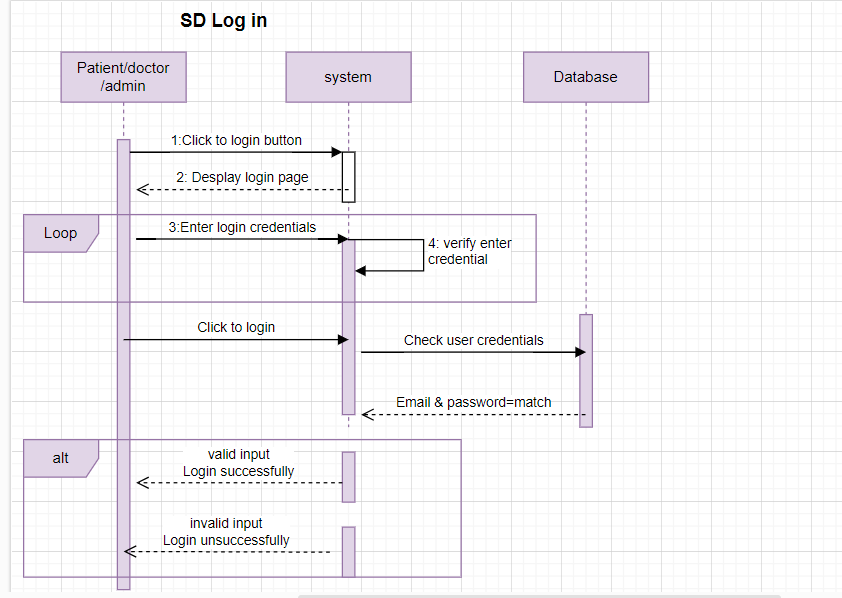
**3.6.3.9 Logout:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | | Logout | | |
| Actors | | User | | |
| Summary | | Ends the user’s session in the system. | | |
| Pre-Conditions | | User must be logged in. | | |
| Post-Conditions | | User is logged out and returned to login screen. | | |
| Special Requirements | | None. | | |
| Basic Flow | | | | |
| Actor Action | | | **System Response** | |
| 1 | User selects “Logout”. | | 2 | System prompts to choose a file. |
| **Alternative Flow** | | | | |
| 3-A | None | | 4-A | None |

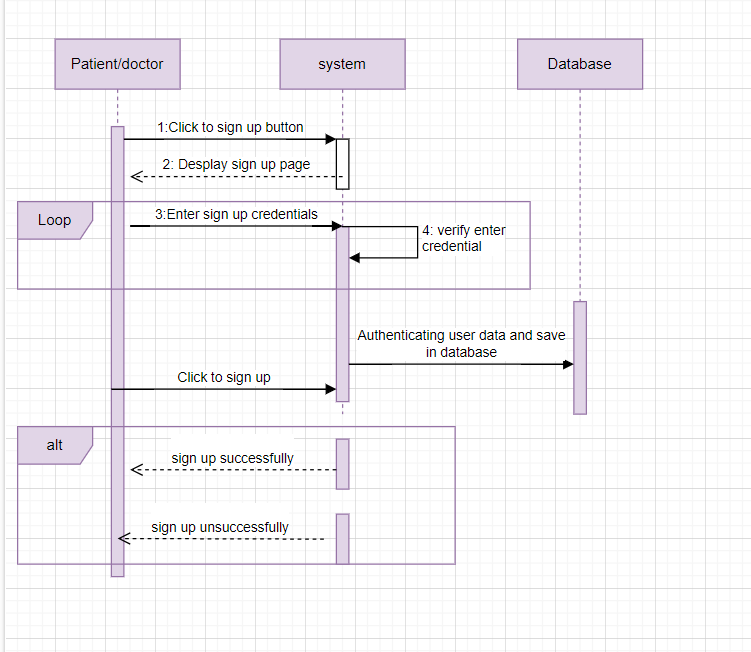
## Class Diagram (*Optional)*

## Sequence diagram *(Optional)*

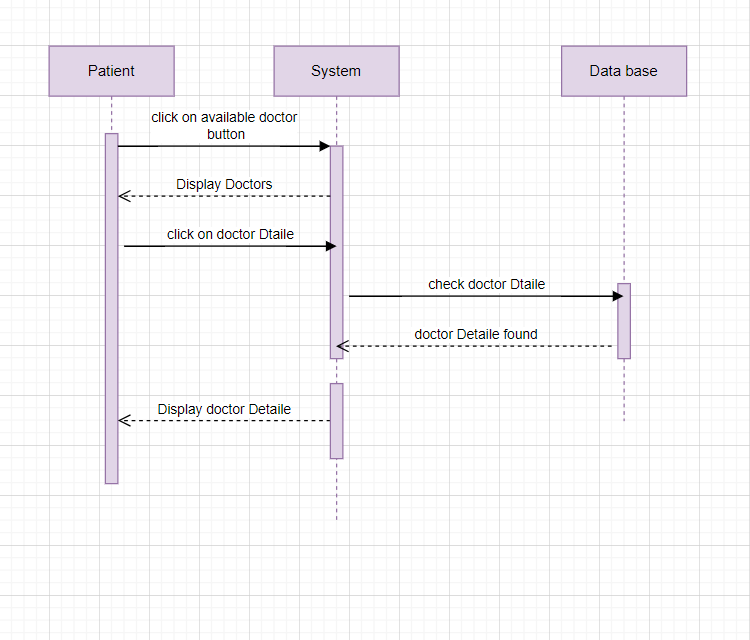
**Login**



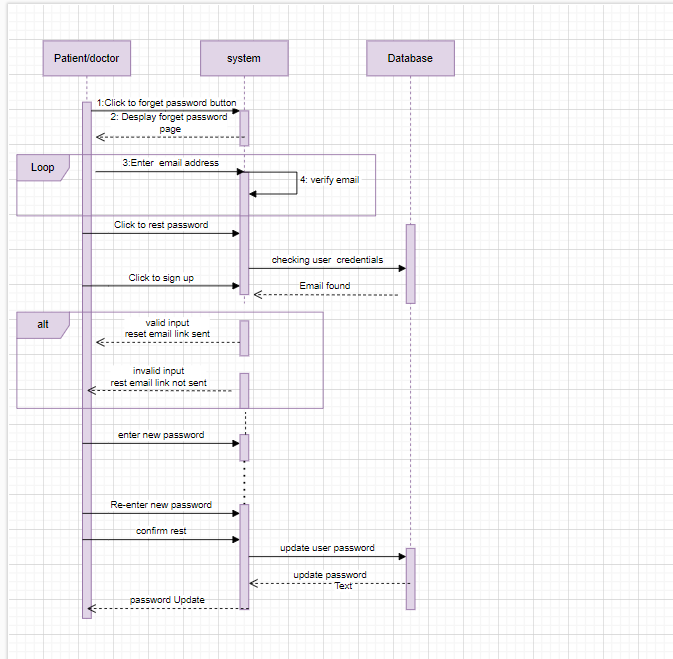
**Sign up:**

****

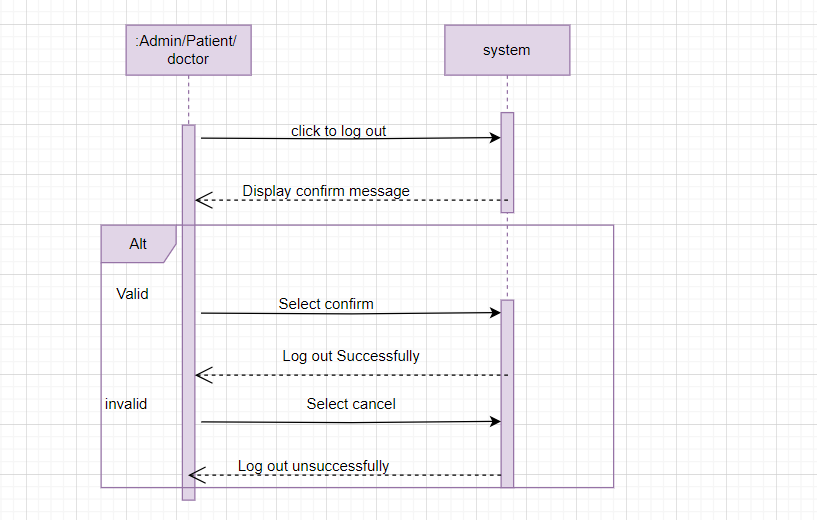
**Doctor Detail**:

****

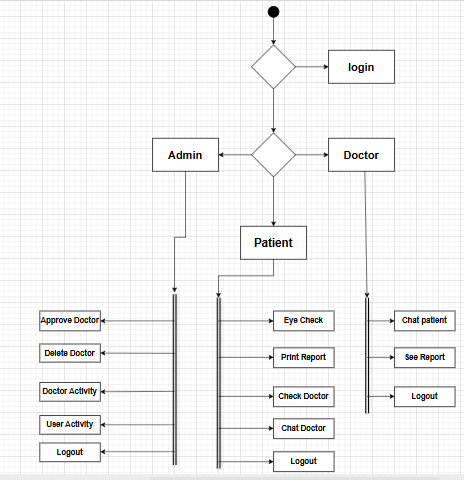
**Forget Password**:

****

**Logout:**

****

## Flow control:



## GUI Graphical User Interfaces (*Optional)*

This section should give the GUI dumps of each screen, with reference to the user. The navigation flow of each user is also required, and each GUI should mark the functionality/use case that it covers.

# Chapter 4: Implementation and Test Cases

**For each chapter provide a paragraph of introduction and in the end a paragraph of conclusions.**

## 4.1 Implementation

This section covers the implementation details of the project's core components, including algorithms used, development environment, tools, and libraries. Python is the primary language due to its versatility and extensive support for machine learning and web development. Key technologies include **Flask** for the web framework, **TensorFlow** for machine learning, and **MySQL** for database management.

### 4.1.1 Implementation of First Component/Algorithm

In our system, we implemented a two-stage classification pipeline using deep learning. In the first stage, a model is trained to classify images as either Valid (retina image) or Invalid (other image). Only the valid images are passed to the second stage, where a separate model classifies them as either Healthy or Diabetic. After experimenting with multiple CNN architectures, we finalized a hybrid model combining features from ResNet50V2 and DenseNet169, leveraging their strengths to improve accuracy. This hybrid model extracts robust features from both architectures, concatenates them, and passes them through dense layers to make the final prediction. This two-step approach ensures that only high-quality images are used for disease detection, improving the reliability and performance of our system.

## **4.2 Test case Design and description**

### 4.2.1 Sample Test case No.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **<Admin Login Module>** | | | | | |
| **<Reference>** | | | | | |
| Test Case ID: | | *TC-01* | Test Date: | | *2025-2-19* |
| Test case Version: | | *V1.0* | Use Case Reference(s): | | *Admin -->Login* |
| Revision History: | | *Initial Version* | | | |
| Objective | | *To verify that Admin can successfully login.* | | | |
| Product/Ver/Module: | | *Diabeto Vision v1.0 – Admin Panel.* | | | |
| Environment: | | *Mongo DB, Node.js, React.js* | | | |
| Assumptions: | | *Admin Account already exist in the database.*  *Admin has network access.* | | | |
| Pre-Requisite: | | *Admin is on the Login page of the system.* | | | |
| Step No. | Execution description | | | Procedure result | |
| 1 | *Enter valid username or password.* | | | *Field accepts input.* | |
| 2 | *Click on “Login” Button.* | | | *System processes login.* | |
| 3 | *Redirect to Admin Dashboard* | | | *Admin dashboard is loaded successfully.* | |
| Comments:   * Ensure the system handles login attempts gracefully. * Failed login should load to another try. | | | | | |
| *Passed* *Failed* *Not Executed* | | | | | |

### 4.2.2 Sample Test case No.2

.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **<Doctor Login Module>** | | | | | |
| **<Doctor Functionalities reference>** | | | | | |
| Test Case ID: | | *TC-02* | Test Date: | | *2025-2-19* |
| Test case Version: | | *V1.0* | Use Case Reference(s): | | *Doctor -->Login* |
| Revision History: | | *Initial Version* | | | |
| Objective | | *To validate that the doctor can successfully log in with valid credentials and access their dashboard to view patient information.* | | | |
| Product/Ver/Module: | | *Diabeto Vision v1.0 – Doctor Panel.* | | | |
| Environment: | | *Mongo DB, Node.js, React.js* | | | |
| Assumptions: | | *Doctor has already signup and has valid account.*  *Credentials are stored in system database.* | | | |
| Pre-Requisite: | | *Doctor is on the Login page of the system.*  *Valid login credentials are available.* | | | |
| Step No. | Execution description | | | Procedure result | |
| 1 | *Enter valid username or password.* | | | *Field accepts input.* | |
| 2 | *Click on “Login” Button.* | | | *Request sent to server.* | |
| 3 | *Redirect to doctor Dashboard* | | | *Dashboard with patient list is displayed.* | |
| Comments:   * Failed login scenarios will be tested separately. * Security of login assumed to be implemented. | | | | | |
| *Passed Failed Not Executed* | | | | | |

### 4.2.3 Sample Test case No.3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **<User upload report Module>** | | | | | |
| **<User Functionalities reference>** | | | | | |
| Test Case ID: | | *TC-03* | Test Date: | | *2025-2-19* |
| Test case Version: | | *V1.0* | Use Case Reference(s): | | *User -->Upload report* |
| Revision History: | | *Initial Version* | | | |
| Objective | | *To ensure that a user can successfully upload a medical report (e.g., test results or documents) to the system..* | | | |
| Product/Ver/Module: | | *Diabeto Vision v1.0 – Doctor Panel.* | | | |
| Environment: | | *Mongo DB, Node.js, React.js* | | | |
| Assumptions: | | *User is already logged into the system.*  *Report file is in accepted format (PDF, JPG, PNG).* | | | |
| Pre-Requisite: | | *User has a digital report file ready to upload.*  *Upload “Report” page is accessible.* | | | |
| Step No. | Execution description | | | Procedure result | |
| 1 | *Navigate to upload report page.* | | | *Page is displayed.* | |
| 2 | *Click “Choose File” and select report.* | | | *File path is shown in input.* | |
| 3 | *Click “Upload” button* | | | *System uploads and confirm successful upload.* | |
| Comments:   * No comments. | | | | | |
| *Passed Failed Not Executed* | | | | | |

## 4.3 Test Metrics

Summarize here the common ground of attributes of test case metrics.

### 4.3.1 Sample Test case Matric.No.1

|  |  |
| --- | --- |
| Metric: | Purpose |
| Number of Test Cases | 12 total test cases have been developed covering Admin, Patient, and Doctor modules shown in the diagram. |
| Number of Test Cases Passed | 10 test cases passed successfully after execution. |
| Number of Test Cases Failed | 2 test cases failed due to incorrect file handling in “Upload Report” and session timeout on “Doctor Chat Patient.” |
| Test Case Defect Density | (2 failed / 12 executed) × 100 = 16.67% |
| Test Case Effectiveness | (5 defects detected via test cases / 6 total known defects) × 100 = 83.33% |
| Traceability Matrix | All test cases are traceable to specific actions in the system diagram: login, approve/delete doctor, patient report upload, chat modules, and logout. Each feature is aligned with its related requirement ensuring 100% traceability. |

# Chapter 5: Experimental Results and Analysis

**Introduction:**

The Diabetic Retinopathy Detection System is developed to facilitate the early and accurate diagnosis of diabetic retinopathy using advanced artificial intelligence techniques. By leveraging hybrid deep learning models, including ResNet50V2 and DenseNet169, this project aims to assist healthcare professionals in making timely decisions, thus improving patient outcomes and reducing preventable vision loss.

1. **Project Achievements:**

The system successfully achieved accurate classification through a two-stage approach: first, distinguishing valid retinal images from invalid ones, and secondly, categorizing valid images into Healthy or Diabetic classes. The final hybrid model notably improved classification accuracy compared to individual CNN models, demonstrating the efficacy of combined architectures.

1. **Critical Analysis:**

* **Security:**

The system employs secure authentication processes for users, doctors, and administrators, ensuring data confidentiality and integrity. Sensitive patient data is encrypted both in transit and at rest, complying with industry-standard cyber security protocols.

* **Scalability:**

Designed for scalability, the system effectively manages increased user traffic and large datasets. Leveraging cloud-based architectures ensures easy resource allocation and minimal downtime during peak usage.

* **Accessibility:**

The user interface is intuitive and responsive, supporting diverse devices such as desktops, tablets, and smartphones. This ensures broad accessibility, allowing users and healthcare providers to access the system seamlessly from various platforms.

* **Regulatory Compliance:**

The system adheres to healthcare regulations, including data protection standards such as GDPR and HIPAA, ensuring legal compliance and patient privacy protection.

1. **Conclusion:**

This project successfully developed a robust and accurate Diabetic Retinopathy Detection System, addressing critical healthcare needs through advanced AI solutions. While ensuring security, scalability, accessibility, and compliance, the system demonstrates significant potential to enhance healthcare delivery and patient outcomes in diabetic care management.

# Chapter 6: Conclusion and Future Directions

**Conclusions**

**1. Summary of Work Done**

In this project, we designed and implemented an AI-based system to detect diabetic retinopathy from retinal images. Using a hybrid deep learning model that integrates ResNet50V2 and DenseNet169, we developed a two-stage classifier: the first model filters out invalid images, and the second classifies valid images as either Healthy or Diabetic. Additionally, a role-based user interface was developed for admins, doctors, and patients to ensure seamless interaction with the system.

**2. Key Findings and Results**

The system achieved high accuracy in distinguishing between healthy and diabetic eyes, and its performance improved significantly through model fusion. We found that image quality plays a vital role in prediction accuracy, justifying the need for an initial validation stage. Moreover, the interface proved user-friendly and responsive across devices, making it accessible to a wide audience.

**3. Scope and Objectives Evaluation**

The core objectives were successfully fulfilled. The system can now detect diabetic retinopathy in retinal images, manage user access by roles, and filter out low-quality inputs. However, due to time constraints, some advanced features such as real-time camera integration and multilingual support were not implemented. The initial scope was mostly covered, with a few enhancements reserved for future phases.

**4. Challenges Faced**

Some challenges included poor image quality in the dataset, managing imbalanced class distribution, and model over-fitting in early stages. Ensuring privacy and security compliance (GDPR, HIPAA) also required detailed consideration and testing. Integration between frontend and backend modules across different roles was complex and needed multiple iterations.

**5. Recommendations for Future Work**

- Expand the dataset with high-resolution and labeled images to further improve model robustness.

- Add real-time image capture via mobile or webcam integration.

- Introduce multilingual support for better accessibility.

- Include explainable AI techniques to visually highlight areas of concern in the eye.

- Improve session management and two-factor authentication for added security.

# References

List all important sources of information which have been consulted for this project

# Appendix

## Appendix A: Guidelines

This section should include all supporting information from the project that was not included in the body of the report.  You should include surveys, complex statistical calculations, certain detailed tables and other such information in an appendix.  The information presented in this section is important to support the work presented in the body of the report but would make it more difficult to read and understand if presented within the body of the report.

Cite the appendix items in the report narrative (write "see Appendix A") and organize appendices (e.g., Appendix A, Appendix B,

Any tables, figures, forms, or other materials that are not totally central to the analysis but that need to be included are placed in the Appendix.

## Appendix B: Heading of Sample Appendix B

Following is a sample code with “code” style format.

Void SampleFunction(){

Print “Hello World.”;

}

# Formatting Guidelines

This document also serves as style guide for final year project reports. In order to give a similar high-quality appearance to all final year software project reports this template uses a collection of predefined Microsoft Word formatting styles. **These styles should be used without modification or replacement.** Font in the document is ***“Time New Roman”.*** This template provides following styles:

* **Title** – the main title style
* **Title2** – the subtitle style
* **Body Text** – style for paragraphs
* **Caption** – the style for a figure or table caption
* **Table Description** – the style for description of table, it must be added after caption.
* **Figure Description** - the style for description of figure, it must be added after caption.
* **Code** – the style for program source code

**int x** = 10; // Writing important code

* **Table Header Row** – Style for the header row of table
* **Table Grid** – the style for the data rows in the tables
* **Reference** – The style for references
* **Bullets** – The style for the bullet lists
* **Numbered** **List**– Style for numbered lists

All Heading styles with different level numbers are listed below.

# Heading 1

## Heading 2

### Heading 3

#### Heading 4

##### Heading 5

###### Heading 6

Heading 7

Heading 8

Heading 9

## Tables and Figures

Tables and figures should be centered horizontally. The caption button should be used to insert caption for both the figures and tables. All figures and tables must be numbered properly. Always refer to tables and figures according to their numbers. A table or figure can be cited as follows: ‘see Table1’ or ‘as shown in Table1’. The caption of table should be centered above the table and figure caption should be centered below the figure. Place the tables/figures close to their reference. Use “Table Header Row” and ‘Table Grid’ style for table’s header and data rows respectively. It is compulsory to provide brief description of table/figure after its caption. Styles for table and figure descriptions are “Table Description” and “Figure Description” respectively.

Press Ctrl+Shift+S to see list of styles mentioned above. Figure 1 shows the Apply Style window displaying the list of styles. Select any text then press Ctrl+Shift+S, the Apply Style window will show you the current style applied on that text and if required, you can change the style by selecting any other style from the “Style Name” dropdown.

This is brief description of above figure.

Figure 1: List of Styles

Table 1: This is Sample table caption

This is brief description of following Table.

|  |  |  |  |
| --- | --- | --- | --- |
| Header row | Header row | Header row | Header row |
| Row1 col1 | Row1 col2 | Row1 col3 | Row1 col4 |
| Row2 col1 | Row2 col2 | Row2 col3 | Row2 col4 |

Table 2: This is Sample table caption

This is brief description of following Table.

|  |  |  |  |
| --- | --- | --- | --- |
| Header row | Header row | Header row | Header row |
| Row1 col1 | Row1 col2 | Row1 col3 | Row1 col4 |
| Row2 col1 | Row2 col2 | Row2 col3 | Row2 col4 |

## Equations

Use equation editor to write equations in this report. Use last button of the custom tool bar to invoke equation editor. Similar to tables and figures, equations should also be aligned centered horizontally. Number all equations and insert them in parenthesis. Below is a sample equation and its reference number. An equation can be referenced like this: ‘it is clear from (1)’.

 (1)

## Header/Footer

Notice the headers in this document, before Introduction (i.e. the main content of this document) page numbers are in roman numerals. The page numbers of the actual content start with Arabic numerals i.e. 1, 2, 3 and so on. All of the **odd numbered pages** contain title of your project while the **even numbered pages** contain the section heading (i.e. chapter’s name) in the headers.

## Other Formatting Guidelines

* Keep 2-4 GUIs in one page. Consume as much space as possible. Do not leave most of page blank unnecessarily.
* Do not break tables (or use cases) in multiple pages unless the table is too large to fit in one page.
* Re-arrange the content i.e., text, images, and tables properly to meet above two guidelines.

## References

Always refer to the source of information by inserting the reference number in square brackets like this [5]. The reference numbers can either be added at the end of the sentence or within the sentence without changing the punctuation of sentence. A reference can also be cited as follows: ‘as Ruskey [2] mentioned’. List each source only once on your reference page.



Figure 2: IEEE Reference style

This figure represents the styling information for adding references in IEEE format

**Following is a list of sample reference for various typed of sources in IEEE format.**

1. P.M. Morse and H. Feshback, *Methods* of *Theoretical Physics*. New York: McGraw Hill, 1953. **//Format for Book**
2. S.K. Kenue and J.F. Greenleaf, “Limited angle multifrequency diffiaction tomography,” *IEEE Trans. Sonics Ultrason*., vol. SU-29, no. 6, pp. 213-2 17, July 1982. **//Format for Journal Article**
3. B. Tsikos, “Segmentation of 3-D scenes using multi-modal interaction between machine vision and programmable mechanical scene manipulation,” Ph.D. dissertation, Univ. of Pennsylvania, BCE Dept., Philadelphia, 1987. [Add if applicable: University Microfilms, Inc., University of Michigan, Ann Arbor, Michigan.] **//Format for Dissertation or thesis**
4. R. Finkel, R. Taylor, R. Bolles, R. Paul, and J. Feldman, “An overview of AL, programming system for automation,” in *Proc. Fourth Int. Joint Conf Artif. Intell*., pp. 758-765, Sept. 3-7, 1975. **//Format for Proceedings paper**
5. “Technology threatens to shatter the world of college textbooks, *The Wall Street Journal*, vol 91, pp. Al, A8, June 1, 1993. **//Format for Newspaper article**
6. R. Cox and J. S. Turner, “Project Zeus: design of a broadband network and its application on a university campus,” Washington Univ., Dept. of Comp. Sci., Technical Report WUCS-91-45, July 30, 1991. **//Format for Technical Report**
7. M. Janzen, *Instant Access Accounting*. Computer software. Nexus Software, Inc IBM-PC, 1993. **//Format for** **Software**
8. Fuminao Okumura and Hajime Takagi, “Maglev Guideway On the Yamanashi Test Line,” *http://www.rtri.or.jp/rd/maglev2/okumura.html*, October 24, 1998. **//Format for** **World Wide Web** (give author and title if named)
9. “AT&T Supplies First CDMA Cellular System in Indonesia,” http://www.att.com/press/1095/951011.nsa.html, Feb 5, 1996. **//Format for World Wide Web**