MAKERERE  **UNIVERSITY**

GLAUCOMA DETECTION SYSTEM

By
BSE 23-30
WEB APPLICATION AND ARTIFICIAL INTELLIGENCE
DEPARTMENT OF NETWORKS
SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY

A Project Report Submitted to the School of Computing and Informatics Technology for the Study Leading to a Project in Partial Fulfillment of the Requirements for the Award of the Degree of Bachelor of Science in Software Engineering of Makerere University.

Supervisor
DR. MUWONGE BERNARD SSAJJABI
DEPARTMENT OF NETWORKS
School of Computing and Informatics Technology, Makerere University
bmuwonge@cit.ac.ug, +256776312246,

July, 2023

Declaration

We, group BSE 23-30 hereby declare that the work presented is original and has never been submitted for an award to any university or institution of higher learning. We can confirm that where we have done consultations either from published material or the works of others, it has been attributed in this report.

#	Names	Registration Number	Student Number
1	KIMURI VIANNEY W	19/U/12088/PS	1900712088
2	KAVUMA MARK	19/U/12292/PS	1900712292
3	BAHATIISA LUKIA	19/U/8204/EVE	1900708204
4	KYOMYA MUHAMMED	19/U/28697	1900728697

Approval

This project report titled GLAUCOMA DETECTION SYSTEM has been submitted for examination with my approval as the supervisor of group BSE23-30.

Dr. Bernard Muwonge

Department of Networks

School of Computing and Informatics Technology;

College of Computing and Information Sciences,

Makerere University

Signature: ..... Date: 19/07/23

Supervisor

Dedication

This work is dedicated towards the technological advancement of the medical field, especially the research and work practiced in the detections and diagnosis of eye diseases and the general field of ophthalmology.

Additionally, we dedicate this report to our lovely parents and guardians who have funded our education from childhood up to this point in life.

Acknowledgements

First and foremost, we express our gratitude to the Almighty God for His guidance and support throughout our University journey and most importantly during the writing of this report.

Additionally, we acknowledge this report to our lovely parents and guardians who have funded our education from childhood up to this point in life.

We would like to acknowledge our project supervisor, **Dr. Bernard Muwonge** and our academic supervisor, **Dr. Mary Nsabagwa** for the guidance they provided to us during the execution of this project, which would have otherwise been very troublesome to accomplish in some areas.

We are also grateful and thankful to the participants that were involved in this study for their willingness to provide the necessary information that was needed to aid this project. May the Lord reward you richly.

Abstract

Glaucoma is an eye disease which could lead to irreversible blindness if not treated early. Like many other eye diseases, glaucoma in its early stages doesn't usually present visible symptoms and also, its early stage symptoms can show some similarity to some other eye diseases.

Traditionally, the diagnosis of such glaucoma requires very specialized personnel (ophthalmologists) and very expensive tests, as well as expensive equipment to aid in the medical diagnosis, which is not available in every community. Also, people in most communities do not have access to these specialized personnel to carry out these diagnoses for them.

The Glaucoma Detection System (GDS) offers several advantages over traditional glaucoma diagnosis methods. It eliminates the need for specialized equipment and expensive tests, making it more accessible to a wider population. By leveraging deep learning, the tool provides reliable and consistent results, reducing the risk of missed or delayed diagnoses. Moreover, it overcomes the shortage of ophthalmologists in underserved areas by enabling early identification of glaucoma and facilitating timely interventions.

The Glaucoma Detection System, automates the process of diagnosing glaucoma in retinal fundus eye images through the use of a powerful deep learning algorithm called VGG16. Deep learning techniques have proven to be more accurate than traditional machine learning methods on a variety of tasks and thus justifying our need to apply them. This improved efficiency will consequently lead to a reduction in errors in diagnosis.

Table of Contents

Declaration ..... **ii**
Approval ..... **iii**
Dedication ..... **iv**
Acknowledgements ..... **v**
Abstract ..... **vi**
Table of Contents ..... **vii**
Relevant Links ..... **viii**
SOFTWARE DESIGN DOCUMENT **1**
GLAUCOMA DETECTION SYSTEM REPORT **32**
USER MANUAL **65**

Relevant Links

Project Repository: https://gitlab.cranecloud.io/2023-SE4/bse23-30
Project Wiki Page: https://gitlab.cranecloud.io/2023-SE4/bse23-30/-/wikis/Glaucoma-Detection-System-Documentation

Project Blog: https://sites.google.com/view/bse23-30-final-year-project/

SOFTWARE DESIGN DOCUMENT

BSE 23-30

GLAUCOMA DISEASE DETECTION SYSTEM

Software Design Document

NAME	REGISTRATION NUMBER	STUDENT NUMBER
KAVUMA MARK	19/U/12292/PS	1900712292
KIMURI VIANNEY W	19/U/12088/PS	1900712088
BAHATIISA LUKIA	19/U/8204/EVE	1900708204
KYOMYA MUHAMMED	19/U/28697	1900728697

Date: (03/16/2023)

TABLE OF CONTENTS

LIST OF FIGURES 2

1. INTRODUCTION 3

1.1 Purpose 3

1.2 Scope 3

1.3 Overview 4

1.4 Definitions and Acronyms 5

2. SYSTEM OVERVIEW 6

3. SYSTEM ARCHITECTURE 8

3.1 Architectural Design 8

3.2 Decomposition Description 9

3.3 Design Rationale 11

4. DATA DESIGN 13

4.1 Data Description 13

4.2 Data Dictionary 15

5. COMPONENT DESIGN 17

6. HUMAN INTERFACE DESIGN 19

6.1 Overview of User Interface 19

6.2 Screen Images 19

6.3 Screen Objects and Actions 24

7. REQUIREMENTS MATRIX 26

LIST OF FIGURES

Figure 3.1: Modular structure for Glaucoma Detection System 7
Figure 3.2 Context-level diagram for Glaucoma Disease Detection System..... 8
Figure 3.3 Top-level Data Flow Diagram for Glaucoma Disease Detection System 9
Figure 3.4 Structural decomposition diagram for Glaucoma Detection System 10
Figure 5 Web application Login Screen 18
Figure 6 Detection Screen 19
Figure 7 Results Screen 20
Figure 8 Eye Care Tips Screen 21

1. INTRODUCTION

1.1 Purpose

The purpose of the software design document is to describe the architecture, modules, system design interfaces and intended functionality for the Glaucoma Detection System. This document aims to guide developers implementing the system by providing them with the overall architecture, system functional requirements and the interfaces of the system. The intended audience also includes the supervisor who will use it to evaluate whether the system to be developed meets the requirements and purpose that it is meant to satisfy and project managers who will test and give useful feedback to the developers.

1.2 Scope

The Glaucoma Disease Detection system will provide aid to early diagnosis of glaucoma disease by the health Practitioners. The system will detect the presence of glaucoma from the eye image uploaded, provide results, and recommend eye care tips.

1.2.1 *Goals*

To develop a system that can be used to aid the diagnosis of glaucoma eye disease.

The project will support the following functionalities;

· Allow an authenticated user to upload eye images.

· Analyze the uploaded images for presence of glaucoma.

· Allow an authenticated user, who performed an analysis task to view the soft copy format of the results.

· Allow an authenticated user to download a copy of the results from the system.

· Store user account information in the database

· Store a copy of the analysis report in the database for users to view.

1.2.2 *Objectives*

The Following are the objectives of this project;

1. To design an Automated System to Detect Glaucoma Eye Disease that would be used by ophthalmologists to aid in the diagnosis of glaucoma

2. To provide a cheaper solution that can be easily accessed by communities through their health centers to be able to aid the diagnosis of glaucoma.

1.2.3. Benefits

The Glaucoma Detection System is therefore of the following benefit to its intended audience

1. Health Practitioners will be able to diagnose glaucoma from the retina eye images at an early stage.

2. System authentication from the user interface is to provide a security mechanism for only the intended user to be accepted to use the system services

1.3 Overview

This document is organized in 8 sections as described below;

Table 1.1. Sections and their different purposes

Section	**Purpose**
Section 1	Provides an introductory description of the System. It contains the purpose, the scope, reference material, definitions and acronyms of this document.
Section 2	This section contains the system overview. This provides detailed information about Glaucoma Disease Detection system functionality, to portray a good sense of the capacity of our system to the user, describe what the system will do and what it will interact with.
Section 3	This section contains the system architecture. This has the system

	architectural design, its decomposition description and reasons that support the choice of the architectural design to be followed in the implementation of the system.
Sections 4	This section contains the description of the data that will be used by the different system components to deliver its functionality to the user.
Section 5	This section contains the components' pseudocode logic. This will guide the implementation of the system components.
Section 6	The sixth section contains the system user interface layouts and the detailed description of the interface object actions like screen objects and images.
Section 7	Contains the requirements matrix that is used to demonstrate the relationship between requirements and other system artifacts and to prove that all requirements have been traced to ensure a seamless guide to update of the system artifacts.
Section 8	This section contains the appendix. This provides supplementary material for better understanding of the information in this document.

1.4 Definitions and Acronyms

Table 1.2. Acronyms and Definitions

Acronym	**Definition**
GDS	Glaucoma Detection System
OCT	Optical Coherence Tomography

2. SYSTEM OVERVIEW

The glaucoma detection system is a web based application that will be designed to detect and diagnose glaucoma, an eye disease that can cause irreversible blindness if it is not detected and treated earlier. Our system is intended to be used by the ophthalmologists to assist them in the diagnosis of the disease faster since the machines take longer time and they are expensive to be affordable by most of the eye clinics.

The system uses a deep learning algorithm to analyze the fundus images taken by the peek retina and provide a quick diagnosis for glaucoma.

The functionalities of the system are as follows:

1. Image processing: The images are taken with the aid of the peek retina camera and they are uploaded into the system by the user for processing.

2. Feature Extraction and classification: The system extracts relevant features from the fundus images that are indicative of glaucoma and uses machine learning algorithms to classify the images as glaucoma or non-glaucoma.

3. Analysis Report: The system automatically displays a results report after feature extraction and classification for the user to view their results.

4. Recommend Eye care tips: The system furthermore recommends eye care tips to the user automatically in case they do not have glaucoma.

5. Recommend nearby eye specialist: The system automatically recommends a nearby eye specialist in case their results show that they have glaucoma so that they can access treatment from them.

System Context

The context of the system is in the field of ophthalmology and optometry. The system is designed to assist the ophthalmologists in the quick diagnosis of glaucoma. The system can also be used in a variety of settings including hospitals, clinics and schools and also private use.

The system is based on a modular architecture that allows flexibility and scalability. The system is designed to easily integrate with existing clinical work flows and electronic medical record systems. The system is also user friendly and easy to use, with a simple navigable user interface.

Generally, the glaucoma diagnosis system is a powerful system to assist the ophthalmologists in the early diagnosis and detection of glaucoma hence potentially improving patient outcomes and reducing the risk of blindness.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design

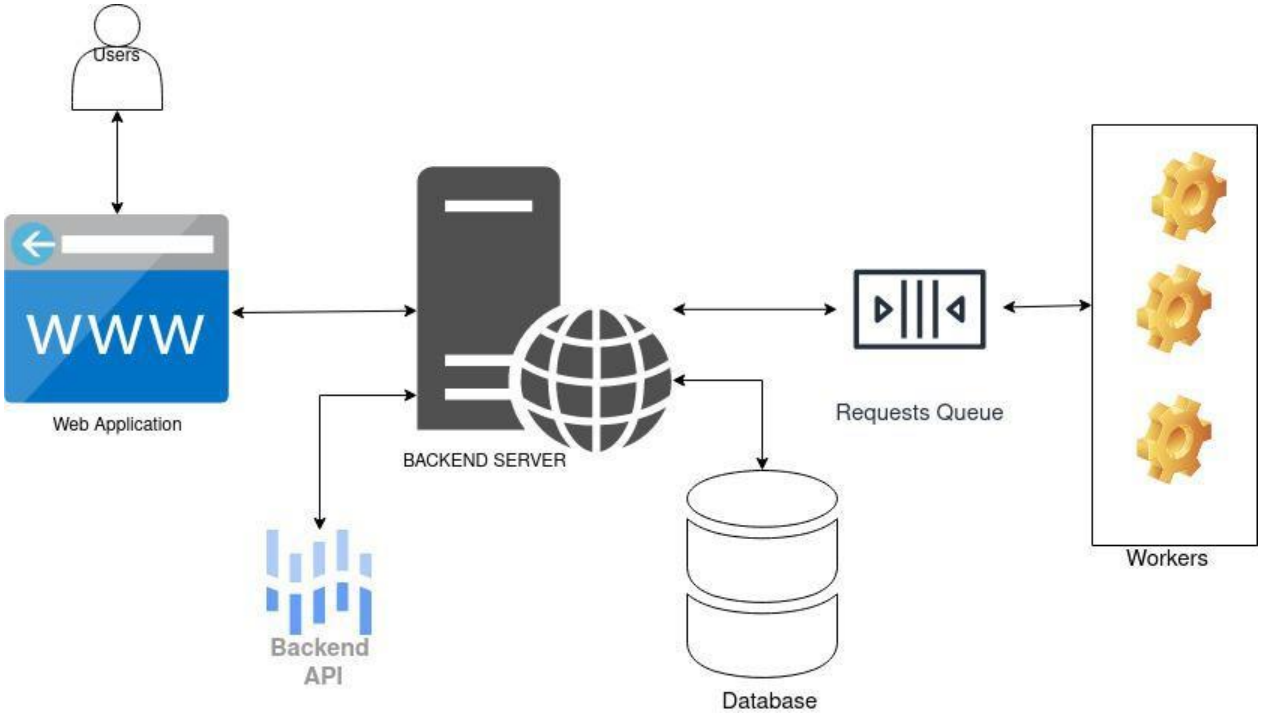


Figure 3.1: Modular structure for Glaucoma Disease Detection System

The system employs a client - server architecture that consists of a web application and a backend server. The web application is what will be used by the users to upload the retinal fundus images for processing, and also view the results of the analysis. The backend server is where the processing of information, as well as most of the system logic will occur.
A high-level overview of the system's operations is as follows;

- The user will submit an image, and the backend server will make a connection to an external data store where the images will be stored.
- The server stores a reference to the uploaded image for reference in the database.
- The server then queues the analysis job for the uploaded image on the request queue. The requests queue is used to free up the server to be able to accept more incoming requests. This encourages asynchronous processing.
- The server then sends a response, with an identifier (ID) of the queued job to the user, informing them that the image has been received for processing.
- There are workers running on the backend, that take the enqueued job and perform the analysis, then return the results and store them in the database. The worker only takes a job

if it is available (free).

- The web application keeps checking with the server, to check if the job with a given identifier (ID) is complete. The server also provides a response specifying if the request is being processed or is complete. If it is complete, the analysis report is sent along with the response.

3.2 Decomposition Description

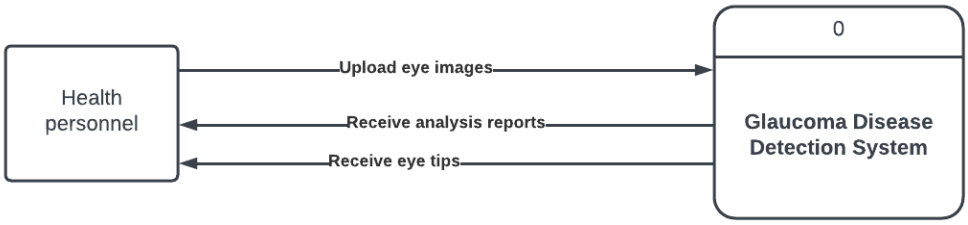


Figure 3.2:Context-level diagram for Glaucoma Disease Detection System

The system provides the following functions to its users:

- It will allow upload of eye images for processing/analysis
- It will analyze the uploaded images for the presence of glaucoma.
- It will provide a report after analysis to the users.
- It will store the uploaded images.
- It will store user details in the database.

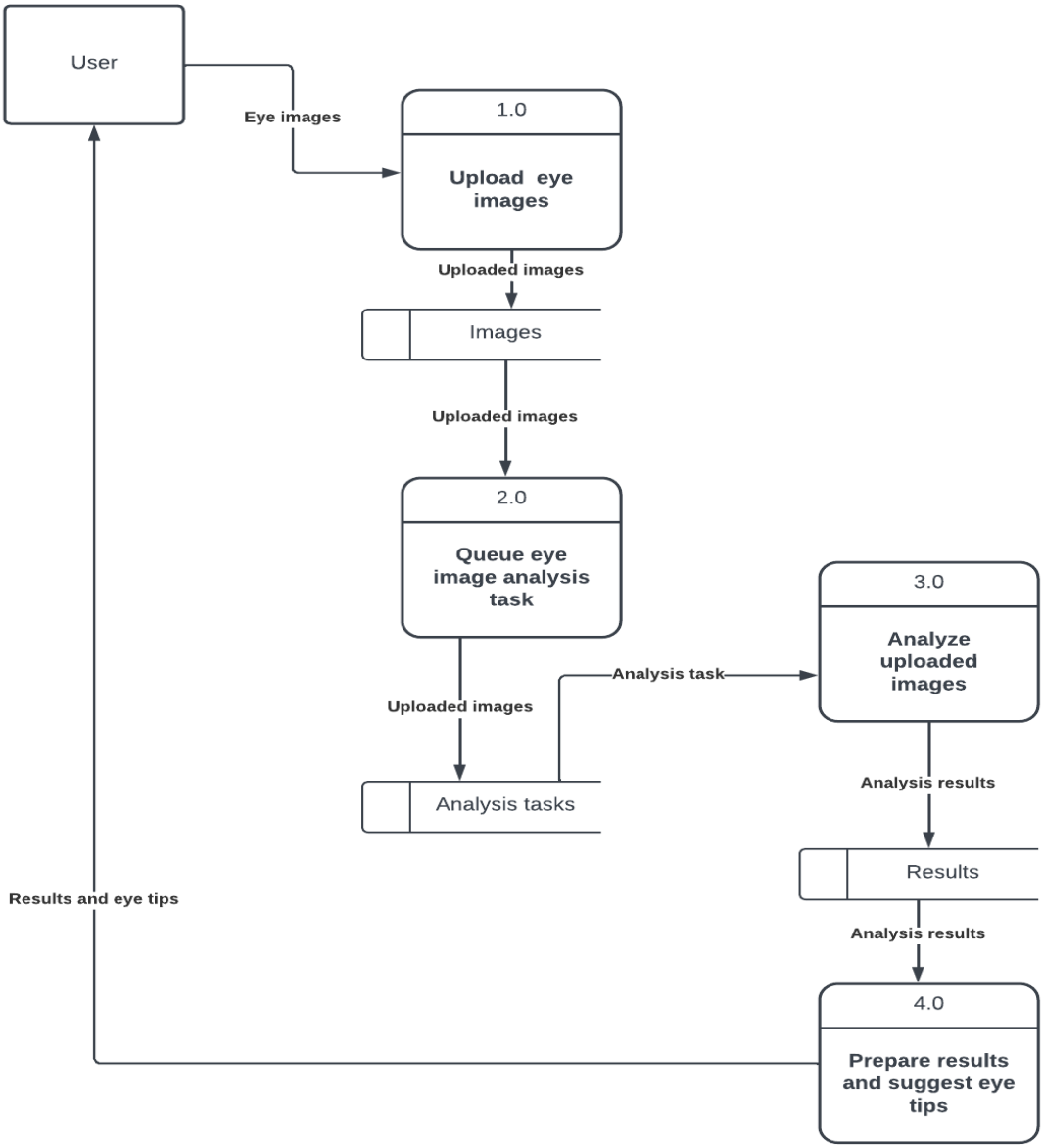


Figure 3.3: Top-level Data Flow Diagram for Glaucoma Disease Detection System

Structural Decomposition diagram

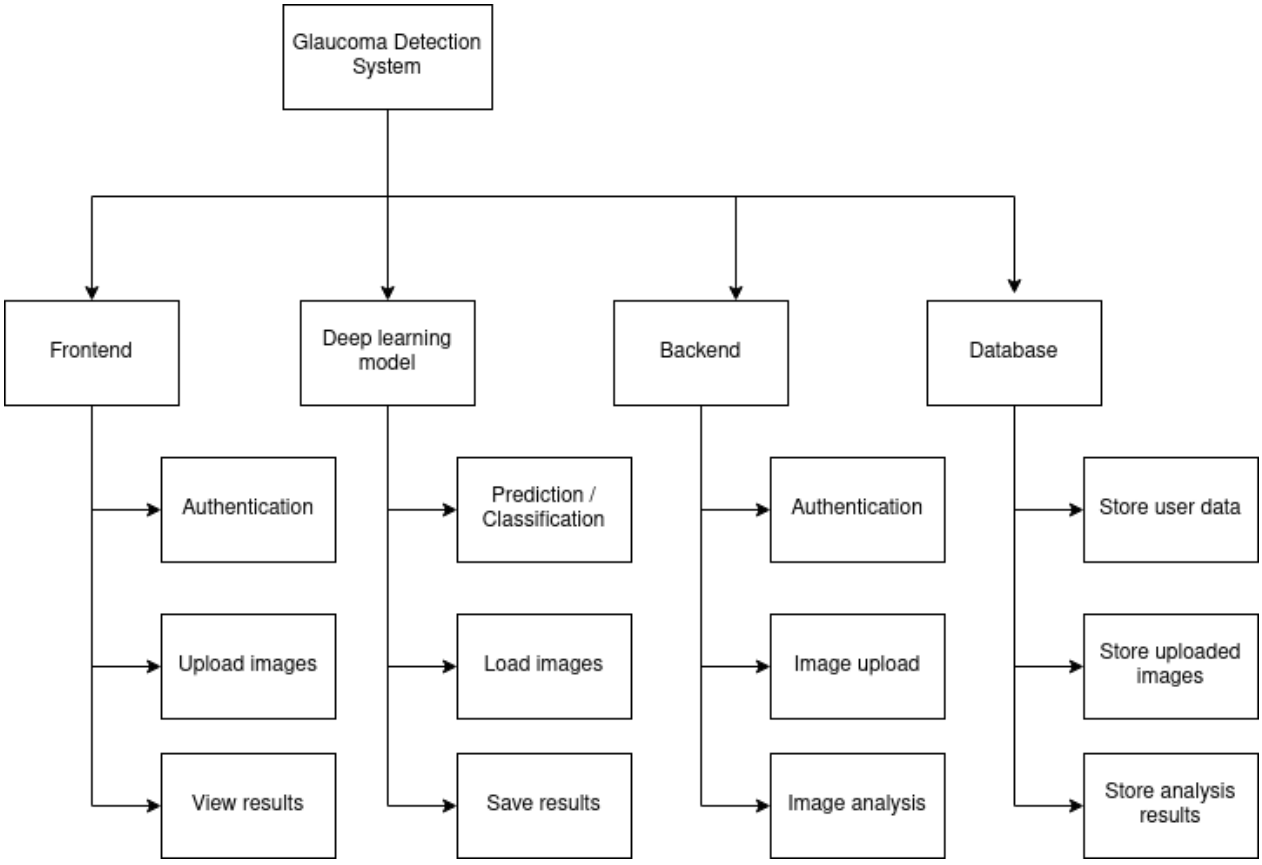


Figure 3.4: Structural decomposition diagram for Glaucoma Detection System

3.3 Design Rationale

The system will use a client-server architecture because of the following reasons:

1. ***Separation of Concerns:***
   The client-server architecture separates the client (presentation layer) from the business logic (server) allowing for easier development, maintenance and scalability of the system

2. ***Centralization***
   The model can be deployed on the server side reducing the need for resources on the client side and simplifying model management. Since the data is also stored in one central place, backup processes can easily be performed in case of any data loss risks.

3. ***Compute intensive tasks***

Since the system is using a deep learning algorithm, which requires heavy compute power such as GPU and CPU resources, it would be better to have a dedicated backend server to handle the analysis tasks of the system.

4. ***Scalability***

   The client-server architecture can easily be scaled by adding more servers to handle increased traffic, making it an ideal choice for a system that may experience a high volume of requests.This can be done without affecting users' experiences on their respective clients.

5. ***Security***

   The use of client-server architecture can allow for better security measures to be implemented at the server-side, such as authentication and authorization, to prevent unauthorized access to the system.

4. DATA DESIGN

4.1 Data Description

The user's and the doctor's(ophthalmologist) information is stored inform of data to the database that is made to hold the name, hashed password, physical address, contact information, email address and username thaton which is authenticated to be granted access to the system once registered. The diagnosis results are stored inform of a stack list at the cloud server and a copy to the user's local machine internal storage. The health tips and the clinic location are stored as Strings in the database machine internal storage. The health tips and the clinic location are stored as Strings in the database

Entities:
Patient.
System.
Attributes:
Patient:
Patient ID
First Name
Last Name
Email Address
Phone Number

Patient Address
Clinic:
Clinic ID
Clinic Name
Ophthalmologist:
Ophthalmologist ID.
Ophthalmologist First Name.
Ophthalmologist Last Name.
Email Address.
Phone Number.
Health Tips.
Data Description:

The system will store information about patients and the registered system users. Each patient will have a unique Patient ID, and their information will include their first name, last name, email address, phone number, and physical address.

The data for this system can be stored in a relational database with the table: Patient. The Patient table would have columns for each of the patient attributes listed above.

4.2 Data Dictionary

1:Table: Data Dictionary For the Glaucoma Detection System

Entity	Attributes	Data Type	Field	Description	Value
Patient	Contact information	String	255	The phone number/s for the user/glaucoma victim	Not Null
Patient	Diagnosis Date	Timestamp	255	The date and time when the user/glaucoma-victim last used the system for checking the status of glaucoma	Not Null
Patient	Glaucoma Status	String	255	The status of the user/glaucoma-victim after diagnosis by the deep learning model	Not Null
Patient	Image	Binary Large Object(BLOB)	255	The image of the user/glaucoma victim that were uploaded by the users	Not Null
Patient	Last Login	Date	255	The date the user/glaucoma-victim last accessed the system	Not Null
Patient	Name	String	255	The name of the personnel that's to use the system for diagnosis	Not Null
Patient	User address	String	255	The physical location of the user/glaucoma victim	Not Null

Patient	User email	String	255	The phone e-mail for the user/glaucoma victim	Not Null
Patient	User Id	Integer	255	A unique number assigned to the user/glaucoma victim by the system at period of account sign up	Primary Key Not Null
Patient	User name	String	255	A special name that the user/glaucoma victim of the system will use at the login page for authentication	Not Null
Patient	User password	String	255	A sequence of characters that the use to be authenticated by the system	Not Null

5. COMPONENT DESIGN

Deep learning algorithm component

```
start:
    model ← loadModel(preTrainedModel)
    imagesForAnalysis ← getImagesFromUser()
    results ← None     #initialize results variable

    results ← model.predict(imagesForAnalysis)   # serve images to model to analyze

    buffer ← results.save() #store images in temporary buffer
    results.show() # display analysis results

    for result in buffer:
        database.add(result)   #stores results in buffer to database


end
```

Upload Images Component

```
start:
    upload_details ← request.data   #get image from request
    validate_upload_details ← authenticate() #check that upload was made
    if is_valid:
        save_upload_details ← ImageUpload.save()
        return ← ImageUpload()

end
```

View Uploaded Image component

```
start:
    fetch_uploaded_image ← ImageUploads.filter(user = request.user)
    paginate_query ← uploaded_image.paginate()
    return ← uploaded_image
end
```

Delete Uploaded Image component

$start$:

$fetch\_uploaded\_image \leftarrow ImageUploads.get(pk = request.data.pk)$

$delete\_uploaded\_image \leftarrow uploaded\_image.delete()$

$return \leftarrow None$

end

User Registration component

```
start:
        registration_details ← request.data
        validate_registration_details ← validate()
        register_user ← User()
        return ← access_token
end
```

Authentication Login component

```
start:
        get_login_details ← request.data
        authenticate_login_details ← authenticate()
        if authenticated:
                return ← access_token
        return None
end
```

Generate report component

```
start:
        fetch_uploaded_images ← ImageUploads.get(pk = request.data.pk)
        generate_report ← uploaded_image.generate_report()
        return ← image_report
end
```

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

The user will interact with the system in-order to perform three main functions i.e. upload the eye images, view the Results, and view the Eye care tips generated by the system as explained below: -

6.1.1. Upload the Eye Images

To upload the eye image, the user will need to log in to the web interface using their username and password. Once logged in, the user will need to upload an image by dropping it to the “drop here” feature or upload from a file. The user then clicks on the “Detect” feature which is a menu item on the side bar. The user can then upload the eye image for detection of presence of glaucoma disease.

6.1.2. View the Results

To view the results, the user will need to log in to the web interface using their username and password. Once logged in, the user will click on the "Results" tab in the navigation menu. This will take them to a page where they can view detailed information about the results generated by the system. On the "Results" page, the user will be presented with the table containing Id, description, level of infection and the date of the results generated by the system. .

6.1.3. View Eye Care Tips

To view the results, the user will need to log in to the web interface using their username and password. Once logged in, the user will click on the "Eye Care Tips" tab in the navigation menu. This will take them to a page where they can view a list of all the eye care tips that have been generated by the system.

6.2 Screen Images

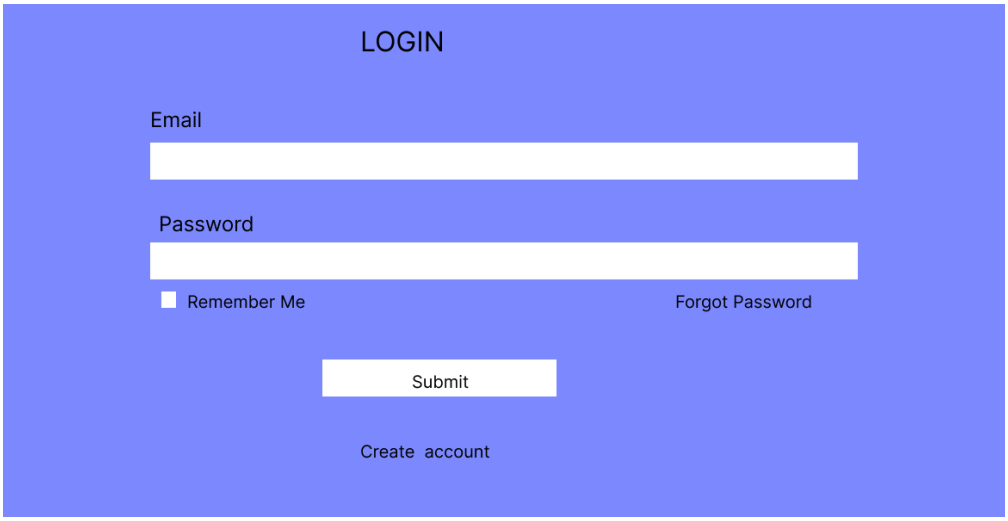


Figure 6.1. Web application Login Screen

Figure 6.1. Will be the screen displayed for the user to log into the application where users will be prompted to enter their login credentials. Once the user has successfully entered their login credentials, they will be granted access to the application's features and functions.

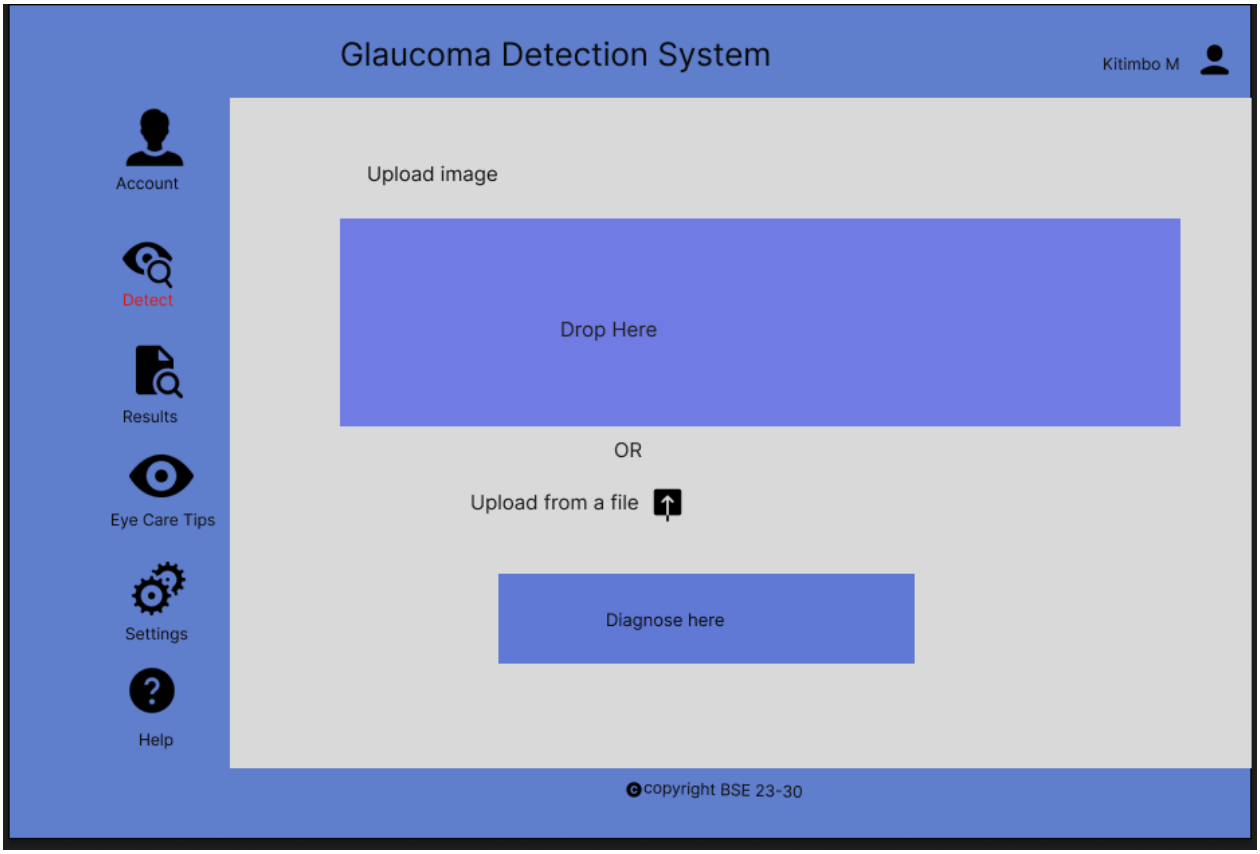


Figure 6.2 Detect Screen

Figure 6.2 will be the first screen that is displayed to the user after they have successfully logged in. To upload the eye, the user will click on the Detect menu item and drop the eye image or upload it from a file image for it to be detected for the presence of glaucoma.

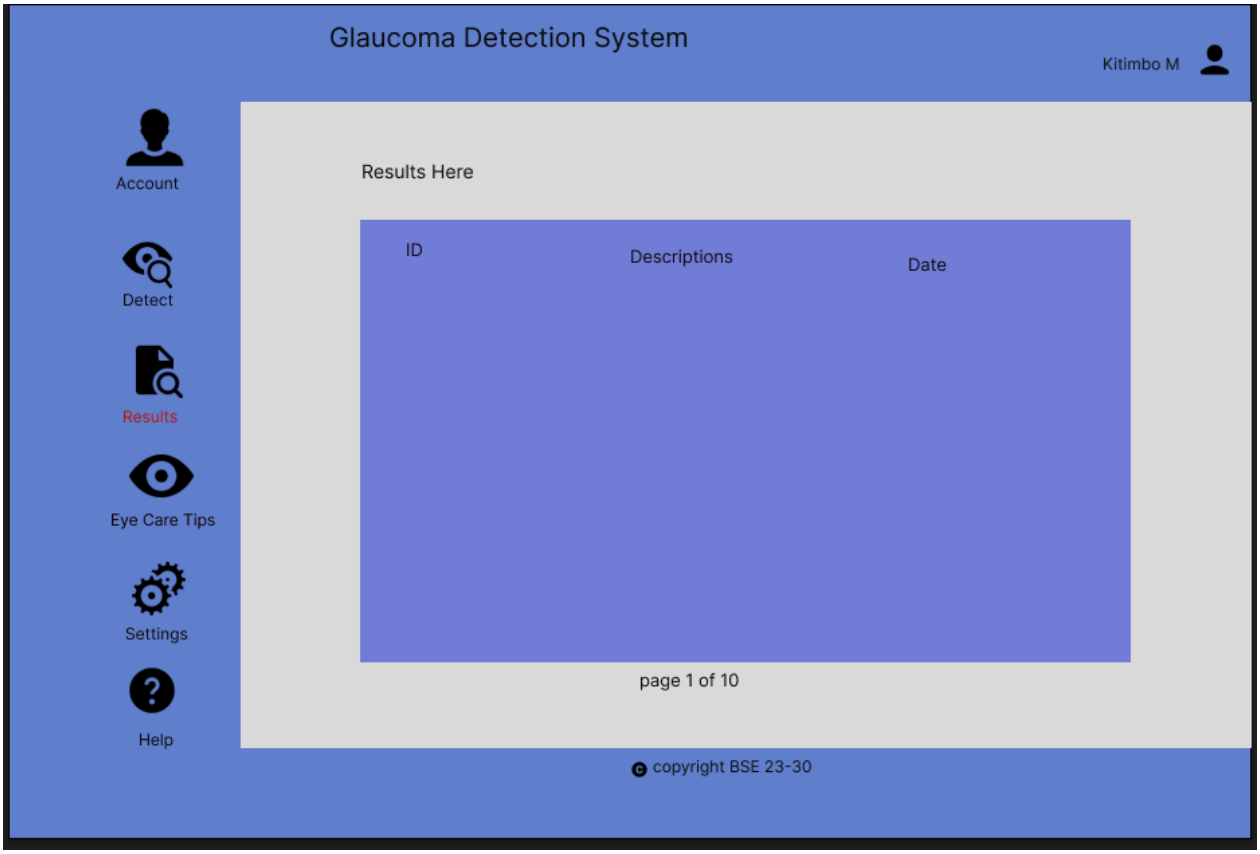


Figure 6.3. Results Screen

Figure 6.3 will be displayed when the user clicks on the Results menu item. The user will view detailed information about the results generated by the system regarding the eye image uploaded.

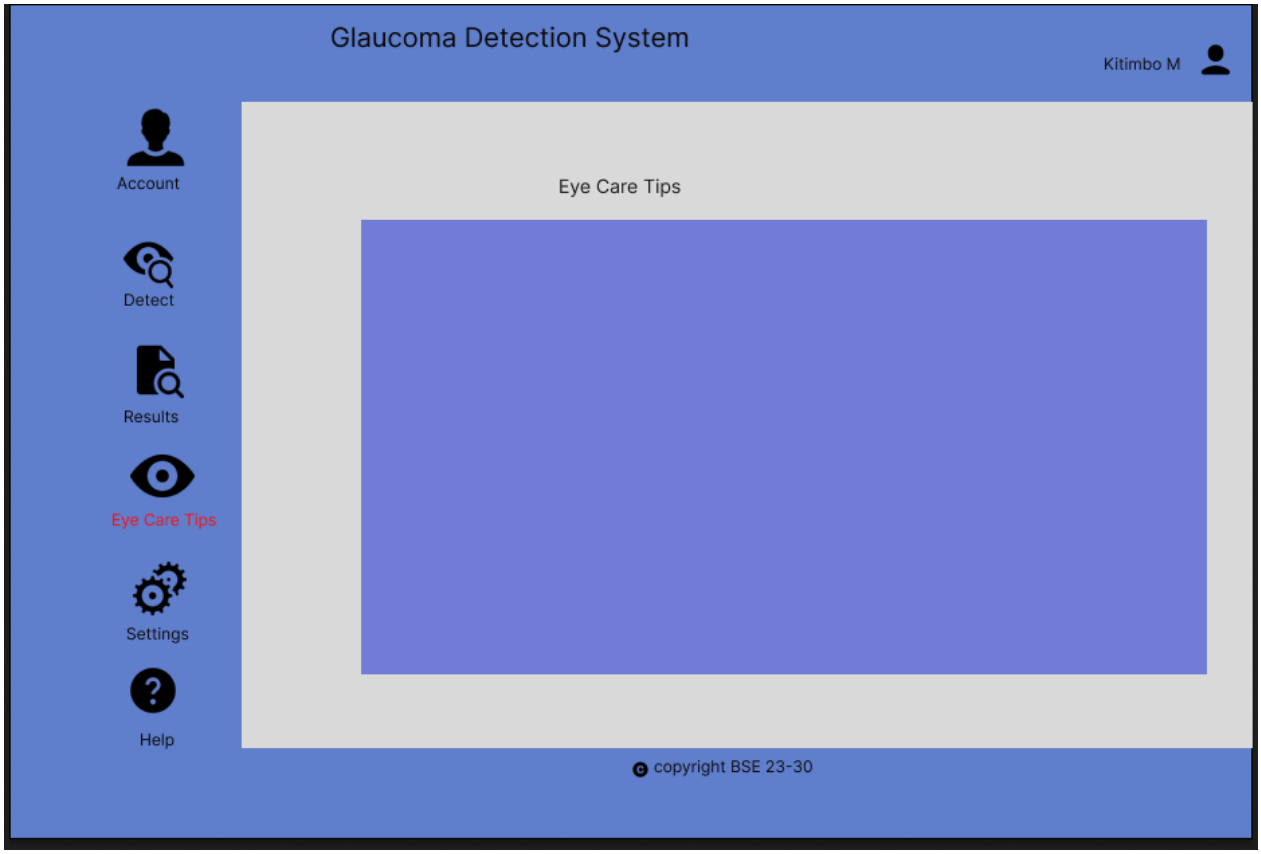


Figure 6.4 Eye Care Tips Screen

Figure 6.4 will be displayed when the user clicks on the Eye Care Tips menu item. The user views a list of eye care tips generated by the system

6.3 Screen Objects and Actions

Table 6.1. Screen objects and actions

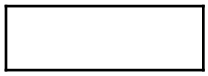
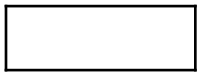
Identifier	Object	Name	Action
Figure 6.1		Username field	This field will allow Users to enter their username to identify themselves to the application.
Figure 6.1		Password field	This field will allow Users to enter their secret password to authenticate themselves and gain access to their account.
Figure 6.1		Submit button	On clicking the submit button, the user's login credentials are submitted for authentication and the user is granted access to their account.

Figure 6.2		Detect	On clicking the Detect menu item, the user will drop the eye image or upload it from a file image for it to be detected for the presence of glaucoma.
Figure 6.3		Result	On clicking the Result menu item, the user will view detailed information about the results generated by the system
Figure 6.4		Eye Care Tips	On clicking the Eye care Tips menu item, the user views a list of eye care tips generated by the system.

7. REQUIREMENTS MATRIX

Requirement ID	Functional Requirements	Priority	Test Cases
GDSF001	User Registration	High	The system shall allow the users to register their personal details into the system in order to be able to get access to the system
GDSF002	User Login	High	The system shall allow the registered user to log into the system by entering their login credentials.
GDSF003	Upload Eye Images	High	User uploads an image in the system for analysis
GDSF004	Storage of Eye Images	High	Images are stored in the system.

GDSF005	Detection of Glaucoma from the images uploaded by Deep Learning	High	Images are analyzed and results are produced by the algorithm and displayed
GDSF006	Analysis Report	High	Analysis results are displayed by the system
GDSF007	User Account details	Medium	User account information will be stored in the system.
GDSF008	Storage of Analysis reports.	High	A copy of analysis reports are stored in the database for the users to preview.

GLAUCOMA DETECTION SYSTEM REPORT

System implementation, testing and validation report for Glaucoma Detection System

GLAUCOMA DETECTION SYSTEM

By

BSE23-30

WEB SYSTEM AND ARTIFICIAL INTELLIGENCE

DEPARTMENT OF NETWORKS

SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY

A Project Report Submitted to the School of Computing and Informatics Technology

for the Study Leading to a Project in Partial Fulfillment of the

Requirements for the Award of the Degree of Bachelor of Science in Software Engineering of Makerere University.

Supervisor
Dr Muwonge Bernard
Department of Networks
School of Computing and Informatics Technology, Makerere University
bmuwonge@cit.ac.ug, +256776312246, Fax: +256-41-540620

June, 2023

Declaration

We, group BSE 23-30 hereby declare that the work presented is original and has never been submitted for an award to any university or institution of higher learning. We can confirm that where we have done consultations either from published material or the works of others, it has been attributed in this report.

No	Names	Registration Number	Student Number
1	KAVUMA MARK	19/U/12292/PS	1900712292
2	KIMURI VIANNEY W	19/U/12088/PS	1900712088
3	BAHATIISA LUKIA	19/U/8204/EVE	1900708204
4	KYOMYA MUHAMMED	19/U/28697	1900728697

Approval

This project report titled GLAUCOMA DETECTION SYSTEM has been submitted for examination with my approval as the supervisor of group BSE23-30.

Dr. Benard Muwonge

Department of Networks

School of Computing and Informatics Technology;

College of Computing and Information Sciences,

Makerere University

Signature: ..... Date: .....

Supervisor

Dedication

This work is dedicated towards the technological advancement of the medical field, especially the research and work practiced in the detections and diagnosis of eye diseases and the general field of ophthalmology.

Additionally, we dedicate this report to our lovely parents and guardians who have funded our education from childhood up to this point in life.

Acknowledgements

First and foremost, we express our gratitude to the Almighty God for His guidance and support throughout our University journey and most importantly during the writing of this report.

Additionally, we acknowledge this report to our lovely parents and guardians who have funded our education from childhood up to this point in life.

We would like to acknowledge our project supervisor, **Dr. Bernard Muwonge** and our academic supervisor, **Dr. Mary Nsabagwa** for the guidance they provided to us during the execution of this project, which would have otherwise been very troublesome to accomplish in some areas.

We are also grateful and thankful to the participants that were involved in this study for their willingness to provide the necessary information that was needed to aid this project. May the Lord reward you richly.

Abstract

Glaucoma is an eye disease which could lead to irreversible blindness if not treated early. Like many other eye diseases, glaucoma in its early stages doesn't usually present visible symptoms and also, its early stage symptoms can show some similarity to some other eye diseases.

Traditionally, the diagnosis of such glaucoma requires very specialized personnel (ophthalmologists) and very expensive tests, as well as expensive equipment to aid in the medical diagnosis, which is not available in every community. Also, people in most communities do not have access to these specialized personnel to carry out these diagnoses for them.

The Glaucoma Detection System (GDS) offers several advantages over traditional glaucoma diagnosis methods. It eliminates the need for specialized equipment and expensive tests, making it more accessible to a wider population. By leveraging deep learning, the tool provides reliable and consistent results, reducing the risk of missed or delayed diagnoses. Moreover, it overcomes the shortage of ophthalmologists in underserved areas by enabling early identification of glaucoma and facilitating timely interventions.

The Glaucoma Detection System, automates the process of diagnosing glaucoma in retinal fundus eye images through the use of a powerful deep learning algorithm called VGG16. Deep learning techniques have proven to be more accurate than traditional machine learning methods on a variety of tasks and thus justifying our need to apply them. This improved efficiency will consequently lead to a reduction in errors in diagnosis.

TABLE OF CONTENTS

Declaration.....**ii**

Approval. ..... **iii**

Dedication. .....**iv**

Acknowledgements. .....**v**

Abstract......**vi**

TABLE OF CONTENTS **39**

List of Figures.....**41**

List of Tables **42**

Abbreviations/Acronyms.....**43**

Chapter 1: Introduction **44**

1.1 Background and scope of the project 44

1.2 Overview of the document.....44

Chapter 2. System Specifications.....**46**

2.1 Version of requirements and Version Control 46

2.2 Input.....46

2.3 Output 46

2.4 Functionality.....47

2.5 Limitations and safety 47

2.6 Default settings 48

2.7 Special requirements 48

2.8 Errors and alarms.....48

Chapter 3: Design output **50**

3.1 Implementation (coding and compilation) 50

3.4 Documentation. 51

A user manual has been provided to guide users in using the application on the web application.....51

Chapter 4: Inspection and testing **53**

4.1 Introduction 53

4.2 Test plan and performance 55

4.2.1 Test objectives 55

4.2.2 Scope and Relevance of tests....56
4.2.4 Types of tests....57
4.2.5 Sequence of tests....57
4.2.6 Configuration and calculation tests....58
4.3 Precautions....58
4.3.1 Anomalous conditions....58
4.3.2 Precautionary steps taken....58
Chapter 5: Installation and system acceptance test....59
5.1 Input files....59
5.2 Supplementary files....59
5.3 Installation qualification....59
Chapter 6: Performance, servicing, maintenance, and phase out....61
6.1 Service and maintenance....61
6.2 Performance and Maintenance....61
Chapter 7: Conclusion and Recommendations....63
7.1 Conclusion....63
7.2 Recommendations....63
Appendix A: User Manual....66

List of Figures

Figure 1 Classification report showing model performance on test dataset 50
Figure 2 login page ..... **Error! Bookmark not defined.**
Figure 3 Register form for Glaucoma Detection System 66
Figure 4: Image upload form on Analyze screen. 67
Figure 5: Results page for GDS system 68
Figure 6: Eye tips screen.....68
Figure 7: User Profile page for Glaucoma Detection System 69
Figure 8: FAQ screen for Glaucoma Detection System.....69

List of Tables

Table 1 abbreviations 43
Table 2. Sections and their different purposes 44
Table 3 design details 52
Table 4 Inspection plan and performance 53
Table 5 Inspection plan and performance 54
Table 6 description of testing 55
Table 7 sequence of tests 57
Table 8 Checklist of the Installation and system acceptance test 59
Table 9 Installation Procedure Check 59
Table 10 Performance and maintenance details 62
Table 11 final approval for use ..... **Error! Bookmark not defined.**

Abbreviations/Acronyms

Table 1 abbreviations

Acronyms/ Abbreviations	Definition
GDS	Glaucoma Detection System
ORM	Object Relational Mapper
GPU	Graphics Processing Unit
NPM	Node Package manager
API	Application Programming Interface

Chapter 1: Introduction

1.1 Background and scope of the project

Glaucoma is an eye disease that poses a significant threat to vision, often leading to irreversible blindness if not detected and treated early. However, in its early stages, glaucoma typically does not present visible symptoms, and its initial symptoms may resemble those of other eye diseases. Furthermore, diagnosing glaucoma requires the expertise of specialized personnel (ophthalmologists) and expensive tests, which are not readily available in every community. These challenges highlight the need for a more accessible and cost-effective solution to aid in the diagnosis of glaucoma.

In response to these challenges, the Glaucoma Detection System (GDS) has been developed to assist ophthalmologists in the diagnosis of glaucoma, providing a cheaper and easily accessible solution for communities through their health centers. The GDS leverages advanced technologies and algorithms to analyze relevant data and support the diagnostic process.

The scope of the project includes the design, development, and implementation of the system. The GDS is user-friendly and specifically tailored to the needs of ophthalmologists, health centers, eye specialists, and schools. It provides a non-invasive and affordable alternative to the traditional diagnostic methods currently available.

It is important to note that the system complements the expertise and judgment of ophthalmologists rather than replace them. It serves as a supportive tool to enhance their diagnostic capabilities and improve the accessibility of glaucoma screening.

By implementing the GDS, the diagnosis and treatment of glaucoma is enhanced, ultimately benefiting individuals and communities by reducing the risk of irreversible blindness. The system provides an easily accessible solution for communities through their health centers, improves the early detection of glaucoma, leading to timely treatment and prevention of irreversible vision loss.

1.2 Overview of the document

This document describes the implementation, testing and validation findings for the Glaucoma Detection System. It is divided into the following sections:

Table 2. Sections and their different purposes

Section	**Purpose**

Section 1. Introduction	Provides a background and scope of the Glaucoma Detection System i.e. a scope on what the Glaucoma Detection System is to address.
Section 2. System Specifications	This section contains the Design Output. This provides a version of requirements and version control, the inputs, outputs and functionality of the Glaucoma Detection System, limitations and safety in the use of the system, its default settings, special requirements and the errors and alarms that could arise in the use of the system.
Section 3. Design Output	This section contains the Design Output. This provides the implementation i.e. the development tools used to implement the system and the documentation from the design.
Section 4. Inspection and Testing	This section contains the ex-tent of the testing in compli-ance with the requirements, elements that are about to be tested, test objectives, scope and relevance of tests, level of tests, types of tests, sequence of tests, calculation tests confirm that known inputs lead to specified outputs and precautions to be taken.
Section 5. Installation and System Acceptance Test	This section contains the List of relevant files on the installation media and what they are used for, Readme files, License agreements and installation qualification i.e. Steps to ensure and document that each component is installed correctly.
Section 6. Performance, Servicing, Maintenance and Phased out	The sixth section contains the Documentation of service and sup-port concerning maintenance, fu-ture updates, problem solutions, requested modifications. It also provides the requirements for service, maintenance, performance
Section 7. Conclusion and Recommendations	Contains the conclusion of the whole report. It also provides information about details on how the system can be used, where to go for help

Chapter 2. System Specifications

This section describes and specifies the system completely and is the basis for the validation process. In this chapter, we describe the whole of the GDS system in detail including its functionality, settings, requirements, expected input, versioning and output.

2.1 Version of requirements and Version Control

Software requirements version name: GDRS_01. This version of the software requirements specification was derived from the initial version of the requirements specification. This version included minor improvements to the performance and safety requirements of the initial version

Software requirements version: V1.0

GDS version control system: GitHub. We used the GitHub platform to track and maintain any changes to our project code.

2.2 Input

Input 1: UUID

A UUID is a unique string. This uniquely identifies each image upload in the database to make querying of results easy.

Input 2: Description

This allows the user to provide some descriptive text for the image they are uploading into the system.

Input 2: Image

This is a url of the image that has been uploaded to our cloud storage service, in this case Cloudinary. These uploaded images are what will be used by the model to perform analysis.

Input 3: User Details.

The user enters register and login details that are used to authenticate the users. The details include name, email and a password.

2.3 Output

Output 1: Analyze image.

The system displays an auto generated ID, a short description of the image to be uploaded and the fundus image itself that is uploaded and previewed.

Output 2: Results

This contains a description of the results produced by the deep learning algorithm after performing analysis on the images. These results can be downloaded in pdf format by the user.

Output 3: Eye care tips

This contains a description of eye care tips that the user can follow to take care of their eyes and maintain good eye health.

Output 4: Account Details.
The users can edit their account information.

2.4 Functionality

1. User Authentication: This makes sure that only the correct and valid users can access the system and perform any operations on it. Without this, no user can have access to any other main system functions such as image upload and access to analysis results.

2. Image upload: This is responsible for uploading eye images to the cloud storage service and the respective link to the image in the database.

3. Data storage: All user data is stored in a PostgreSQL database. With the help of an ORM, queries, as well as read and write can be performed on the database. In this case the ORM used is the DjangoORM.

4. Image classification model: This is the deep learning algorithm responsible for performing automated analysis on the uploaded images. This model uses retinal fundus images of the eye to classify whether the image is of a normal eye or if the image is of an eye with glaucoma.

5. Storage of eye images which can be used to improve the deep learning algorithm through training. When the number of images stored is large enough to create a reasonable dataset, it can be used to manually further train the model to improve its accuracy and precision when dealing with tasks.

2.5 Limitations and safety

All precautions taken to prevent overflow and malfunction due to incorrect input or use. When the system is in use, such must be done to prevent incorrect input or use of the system.
The Glaucoma Detection System has the following limitations in its use;

- There needs to be an active internet connection in order to use this system. One of the reasons for this is that images need to be uploaded to a cloud storage service, among other reasons.
- The system requires GPU hardware in order to provide the required hardware acceleration in order to efficiently use the VGG16 deep learning algorithm for inference on the uploaded eye images.
- The system has not been vetted by any official medical body to check and ensure that the results produced are per the medical standard, acceptable and therefore, it is not advisable to use the system in a real world setting.
- The deep learning algorithm used in the Glaucoma Detection System is limited to classification tasks, in that it can only identify if the images are normal or have glaucoma, but cannot detect the severity or progression of the disease based on the images.

- The system allows only authorized users to use the system.

When using the system, the following precautions need to be taken into account. They include;

- The deep learning model trained and used in this system is still far from being used for any clinical purposes as more in-depth training, evaluation and validation need to be carried out with different sets of data and with the collaborative guidance of practicing expert ophthalmologists. Therefore the system as is in the current version can only be used for experimentation/laboratory purposes and the predictions generated by the model should not be taken as 'gospel truth'.
- The results produced by the deep learning algorithm in the system are still subject to review and validation by the ophthalmologists. This is to ensure the safety of the patients and also there might be cases where glaucoma positive images are classified as normal images and vice versa, therefore, further validation from ophthalmologists can prevent erroneous diagnoses.
- The deep learning model used in the system only handles classification tasks, as it can only identify two classes that is normal eye images and glaucoma eyes.

2.6 Default settings

The system has no users by default. All the users of this system are created using a single registration process offered by the system. Therefore, safety of the data used and stored by this system is ensured.

2.7 Special requirements

- All user credentials such as passwords are encrypted to prevent unauthorized access by anyone other than the user.
- The results produced by the deep learning algorithm in the system are still subject to review and validation by the ophthalmologists. This is to ensure the safety of the patients and also there might be cases where glaucoma positive images are classified as normal images and vice versa, therefore, further validation from ophthalmologists can prevent erroneous diagnoses.
- The system ensures protection and backup of code by sharing the code via version control systems like GitHub.

2.8 Errors and alarms

1. **Invalid registration details:** The user may input invalid data when they are registering into the system. This will lead to errors which are displayed to the user in a clear way such that the user understands the problem.
2. **Invalid login details:** When a user tries to login into the system using the wrong credentials, they receive an error and are requested to use the correct details to access the system.

3. **Invalid image upload**: When a user is trying to upload an image, they may upload another file instead of the required image or try to omit a field for example description and this will raise an error. The error displayed to the user will guide them as to where the problem is.

Chapter 3: Design output

3.1 Implementation (coding and compilation)

The frontend of the system is built using TypeScript and the React Library. All dependencies for the frontend are managed by the **npm package manager**. Npm is a package manager for the Javascript programming language. We used **Visual Studio Code** as our code editor since its optimized for building and debugging modern web applications

The backend of this application is an API that serves the frontend system. It is built using Django and the Django REST framework. All the dependencies for the backend system are managed by the **pip package manager**.

The deep learning algorithm used in this system is built using the Python Programming language and the PyTorch library for machine learning. However, some functions and classes from the scikit-learn machine learning library were also used in some cases for example in measuring the performance of the trained model and handling class imbalance. Additional libraries utilized in the development of this model include pandas for data preprocessing, matplotlib for data visualization, seaborn for some visualizations as well and imshow to display images during data analysis and testing.

We used an open source kaggle dataset of 650 retinal fundus images [1] which was sourced from the kaggle dataset repository to aid in the training of the deep learning model. To perform the model training, we preprocess the data by performing some data augmentations such as random cropping, random centering to allow the model not to memorize the images but instead learn, then we split the data into training, validation and test datasets and then use the VGG16 model with its pre-trained weights to learn to classify the images on training data, while evaluating performance on the validation data. This approach of model training is known as **transfer learning.** The model performance is evaluated after training on the test dataset using the following metrics; precision, recall and F1 score. The F1 score is the preferred metric in this case to help us determine if the model is performing well.

	precision	recall	f1-score	support
0	0.97	1.00	0.99	69
1	1.00	0.94	0.97	35
accuracy			0.98	104
macro avg	0.99	0.97	0.98	104
weighted avg	0.98	0.98	0.98	104

Figure 1 Classification report showing model performance on test dataset

The report shown in the image shows the different metrics on the test dataset for both classes, that is 0 indicating normal eyes and 1 indicating glaucoma eyes.

Since model training is computational intensive, a GPU is required to provide hardware acceleration for both training and inference phases. For this, we used free GPU cloud service providers, in this case Google Colaboratory for the respective phases.

Other tools used in the development of the system include;

- **Insomnia:** This is an API client which is used to test the backend API of the system.
- **PostgreSQL:** This was used to create the database that was used in the backend system of the application.
- **Visual Studio code:** This was used in the programming tasks for our system

Anomalies

- The accuracy of the deep learning model is less than 100% due to the small size of the image dataset that was used in training, among other factors.
- The deep learning model can sometimes detect glaucoma images as normal images. This is due to the class imbalance in the dataset used, where the number of normal images was larger than the number of glaucoma images and also the small size of the image dataset. However, we employed certain machine learning techniques to curb this issue and ensure the model can be able to correctly identify both classes.

3.4 Documentation.

A user manual has been provided to guide users in using the application on the web app

This report is intended to inform a specific section of readers, that is the developers and academic

supervisors on the system design process and implementation, as well as its outputs.

Table 3 design details

Topics	**Design output**	
Good programming practice *Efforts made to meet the recommendations for good programming practice...*	Source code is... ☑ Modulized ☐ Encapsulated ☐ Functionally divided ☐ Strictly compiled ☑ Fail-safe (handling errors)	Source code contains... ☐ Revision notes ☑ Comments ☑ Meaningfull names ☑ Readable source code ☐ Printable source code
Dynamic testing *Step-by-step testing made dynamically during the implementation...*	☑ All statements have been executed at least once ☑ All functions have been executed at least once ☐ All case segments have been executed at least once ☑ All loops have been executed to their boundaries ☐ Some parts were not subject to dynamic test Comments:	

Chapter 4: Inspection and testing

4.1 Introduction

Table 4 Inspection plan and performance

Topics	**3.3.1 Inspection plan and performance**	*Date / Initials*
Design output - **Data dictionary** - **Data flow diagrams(context and Level 0 Diagrams)** - **Screen mockups**	☐ Program coding structure and source code ☐ Evidence of good programming practice ☐ Design verification and documented reviews ☑ Change-control reviews and reports Comments: 1. The expectation will cover entirely the analysis of the data dictionary design to make sure it is consistent with the users system parameters to be captured and stored. 2. Data flow diagrams to scrutinize the control flow of data from one entity to the other. 3. Have an overview of the interfaces the system exhibits	Inspection Date:19-6-2023 Inspection members initials: KM KM KV LB
Documentation: - **Software Requirement Specification**	☑ System documentation, flow charts, etc. ☐ Test results ☐ User manuals, On-line help, Notes, etc. ☐ Contents of user manuals approved Comments: - To evaluate the document against the target user requirements	Inspection Date:25-6-2023 Inspection members initials: KM KM KV LB

Topics	**3.3.1 Inspection plan and performance**	*Date / Initials*
Software development environment - **Local development laptops** - **AI-model-hosting server**	☑ Data integrity ☑ File storage ☑ Access rights ☐ Code protection ☑ Installation kit, replication and distribution Comments: - To check for malware and Trojans on laptops - To ensure that the required softwares are installed - To check the capability and speed of the development laptops - To have an overview of the server services charges. - To access constraints involved in exchanging data with the server - To ensure that the level-0 data flow diagram is consistent with the context system diagram. .	Inspection Date:27-6-2023 Inspection members initials: KM KM KV LB
Result of inspection - **Functionality** - **Performance** - **Portability** - **Effectiveness**	☑ Inspection approved Comments: The system performs the diagnosis of the disease as intended	Inspection Date:19-6-2023 Inspection members initials: KM KM KV LB

Table 5 Inspection plan and performance

4.2 Test plan and performance

Based on the user requirements it is important to test the software and check if the different modules that have been implemented meet the user requirements. This helps to assure the quality of the software before it is released. The testing process requires substantial resources in software development. Thorough testing is necessary to be confident that the system works as it was intended to in its intended environment.

4.2.1 Test objectives

The main objective is to verify that the functionality of Glaucoma Disease Detection System version 1.0 works according to the specifications provided in its Requirements Specification Document. The tests were executed to verify that the test cases identify, fix and retest the defects in the severity classes. Other objectives include;

➔ To examine the different modules that have been implemented have the correct functionality based on the requirements.

➔ To improve the software's quality.

➔ To identify and locate the errors in the software.

➔ To define the tools to be used throughout the testing process.

➔ To communicate to the responsible parties, the items to be tested, the schedule, the test budget and defined environment needs for testing

Table 6 description of testing

What was tested	**How it was tested**	**Why it was tested**
System's ability to restrict access to only valid users	Multiple login attempts with both correct, incorrect credentials and with malicious scripts	To verify that only registered users can access the system features
User's ability to upload correct files ie images	Upload wrong file type	To ensure that the user can upload only the correct image file type
System's ability to send a correct response after	Series of image analysis requests were made for both	To verify that the correct response message and

handling image analysis	images that had glaucoma and those that didn't have	details are sent to the user after image analysis

4.2.2 Scope and Relevance of tests

The scope of the tests that were conducted is based on code coverage and module-based coverage. Code coverage is based on code, for example React code and module-based coverage is based on the modules that is the

- User Account Authentication module
- Image analysis module

The tests were designed to cater for verification and validation testing techniques. Verification testing covered the systems requirements during the requirements phase while validation testing was employed during the implementation phase.

4.2.3 Levels of tests

Module Tests: These tests were largely white-box oriented. Each module was isolated and tested independently. This saved a lot of time since issues found in a specific module were quickly detected and resolved.

Integration Tests: After the successful testing of the individual modules, we combined the modules and tested them as a whole. This was intended to expose any defects that may arise in the interactions between the integrated modules in the system. Complete use cases such as image analysis and provision of results to the users to view were performed during these tests.

System Acceptance Test: The purpose behind system acceptance testing is to confirm that the system is developed according to the specified user requirements and is ready for operational use. After testing the individual modules, we combined the modules and tested them as a group. This was intended to expose any defects that may arise in the interactions between the integrated modules in the system.

4.2.4 Types of tests

Input Tests: These were intended to test for the issues that may arise depending on user input such as the images for analysis and malicious characters that may result in security threats.

Functionality Tests: These were performed to ensure that the application proves positive against the functional requirement specifications. The functions were tested for correctness, reliability, testability and accuracy of the expected output/data.

Usability Tests: These were carried to ensure that the system is easy to use and the end user will be able to interact with the system easily. Testing focused on the look and feel, error messages, spelling mistakes and GUI guidelines.

Performance Tests: These were done to evaluate the performance of the system with consideration of metrics such as image analysis time, page load times and maximum load the system can handle.

4.2.5 Sequence of tests

Table 7 sequence of tests

Test Case	Description	Test Procedure	Test Data	Expected Result
TC01	Verify if user is authenticated to get access into system	Input email and password in the login form	Valid email and password	Corresponding dashboard displayed or error message displayed for failed authentication
TC02	Verify that the user can upload file and file is valid format.	Log in as a user and upload any file.	Files in different formats such as .jpg, .pdf, .png etc.	File uploaded or error message indicating invalid file format
TC03	Validate correct results	Place an image analysis request	An image for	Success message and

	are sent to the user after the image is analyzed	to detect the disease by submitting image	analysis	analysis details in case of image containing the disease or not

4.2.6 Configuration and calculation tests

The system is accessed through various devices connected to the internet and its functionality and behavior is observed via the web.

4.3 Precautions

4.3.1 Anomalous conditions

Anomalies are events that differ from the standard events defined in the application. Therefore, anomalous conditions in our application basically refer to the deviations of the system from the user expected behavior. From our system, the following anomalous condition may occur during operation of the application.

- The system is entirely online and thus some features may delay or lag to respond due to poor internet connection.

4.3.2 Precautionary steps taken

Make sure to have a stable internet connection to use the system to upload the fundus images to our AI algorithm.

Chapter 5: Installation and system acceptance test

5.1 Input files

For the Glaucoma Detection System, no input or installation files will be required since the system is a web application written in TypeScript which is included in all of the top modern browsers. Therefore, the user only requires a web browser to allow them to navigate to the application.

5.2 Supplementary files

There are no supplementary files provided for the application's installation process. A user manual will be provided to guide the users on how to use the system.

5.3 Installation qualification

Steps to ensure and document that each component is installed correctly

Table 8 Checklist of the Installation and system acceptance test

Topics	**Installation summary**
Installation method *Automatic or manual installation...*	☑ Automatic - installation kit located on the installation media ☐ Manual - Copy & Paste from the installation media Comments:
Installation media *Media containing the installation files...*	☐ Diskette(s) ☐ CD-ROM ☑ Source disk folder (PC or network) ☐ Download from the Internet Comments:
Installed files *List of (relevant) installed files, e.g. EXE- and DLL-files, spreadsheet Add-ins and Templates, On-line Help, etc.*	• Frontend Typescript files • Django REST API files • Pytorch model (PTH files)

Table 9 Installation Procedure Check

Topics	**Installation procedure**	*Date / Initials*
Authorization	Person responsible: Development team member	20/06/2023

Installation test	☑ Tested and approved in a test environment ☐ Tested and approved in actual environment ☑ Completely tested according to test plan ☐ Partly tested (known extent of update)	20/06/2023

Chapter 6: Performance, servicing, maintenance, and phase out

6.1 Service and maintenance

The service and maintenance of The Glaucoma Detection System shall be handled as follows;

- The programmers shall put in effort to ensure that any bugs identified within the system are resolved to ensure smooth operation.
- Any issues encountered while working with the system should be reported to the development team through the contacts listed on the system.

6.2 Performance and Maintenance

Since the system is powered by an API, most of the updates will be done on the backend and these changes, when made, shall be instantly accessible to the users of the Glaucoma Detection system through the web client / interface.

The deep learning model will constantly be improved with an increase in the number of images stored in the system. These images shall enable creation of a sizeable dataset to support the manual training of the model. This will provide improved performance to the users of the system.

In the case of system bugs, registered users will be able to report issues and tickets will be created by the development team accordingly to track the issues' resolutions. We plan on using an issue tracker for this and the issues will be solved based on their priority in the system, from critical issues to the minor ones.

The need for service and maintenance might arise due to some issues or factors including, detection of bugs, slow processing, scaling servers or databases. The other factor is software upgrades and ensuring compatibility to the new operating systems or different environments. The team will be able to provide the following support where need be.

- Collaborating with the technical in-house team to ensure familiarity with technology.
- Email support will be available for users and all support emails should receive response within 24 hours.
- Data migration from the old system. There will be a phased cleaning and migration of the data in the old system to the new system.

- Diagnosing of the system problems, inefficiencies, and weaknesses.
- Watch out for any bugs in the system and fix them as quickly as possible to avoid poor user experience.

Table 10 Performance and maintenance details

Topics	Performance and maintenance	Date / Initials
Problem / solution	Problem: Errors while uploading the images for analysis, when multiple images are added to the waiting list, it's impossible to alter it. Solution: The images will first be stored in a database, to give a chance for altering them in case of an error.	KM, BL
Functional maintenance	In case the system upgrades the stakeholders will be notified and the project documentation shall be updated when new user requirements arise.	KM, BL, KV, KM
Functional expansion and performance im-provement	Maintain a stable internet connection since the system is mainly an online system. Shifting to a more scalable network to maintain the availability and efficiency of the system.	KM, BL, KV, KM

Chapter 7: Conclusion and Recommendations

7.1 Conclusion

The Glaucoma Detection System is still in its infancy and is still a long way from being used for clinical purposes. The deep learning model used in the system still needs to be trained, evaluated and validated on new sets of data with the help of a skilled ophthalmologist in order for it to drastically improve and be used in a clinical setting. Therefore, the current version of this system can be used for experimentation and laboratory purposes

7.2 Recommendations

Despite the success of the glaucoma disease detection system, there should be further improvement. The system should undergo rigorous testing using diverse patient populations to ensure its performance across different demographics and stages of the disease.

There should be collaboration with healthcare providers for successful implementation of the glaucoma disease detection system. By meeting experts in the field, the system can be fine-tuned to meet specific clinical requirements and integrate seamlessly into existing healthcare workflows.

We recommend that for further versions of this system, in order to ensure patient's safety, the results provided should be subjected to further review by an ophthalmologist to reduce the risk of error and misdiagnosis, as there can be cases where glaucoma images can be classified as normal images

Final approval for use	
Identification:	
Responsible for validation:	
Remarks:	
Date:	Signature:

USER MANUAL

GLAUCOMA DETECTION SYSTEM
User Manual

Register screen.

All new users register into the system using their valid names, email and password which are then submitted to the database.

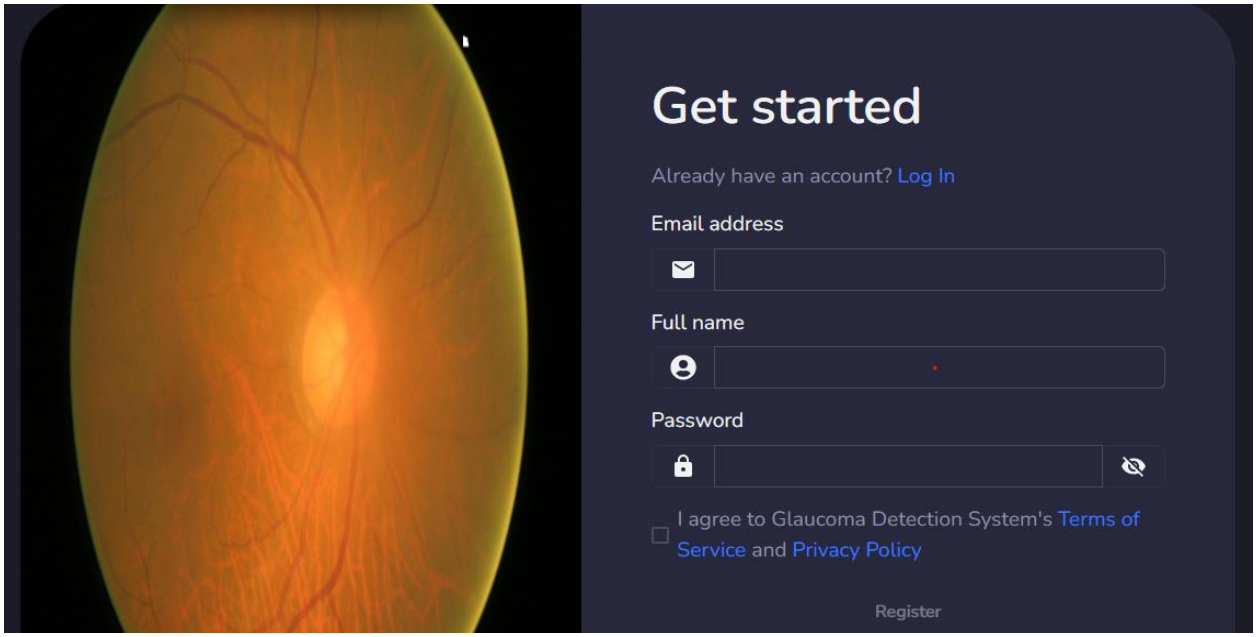
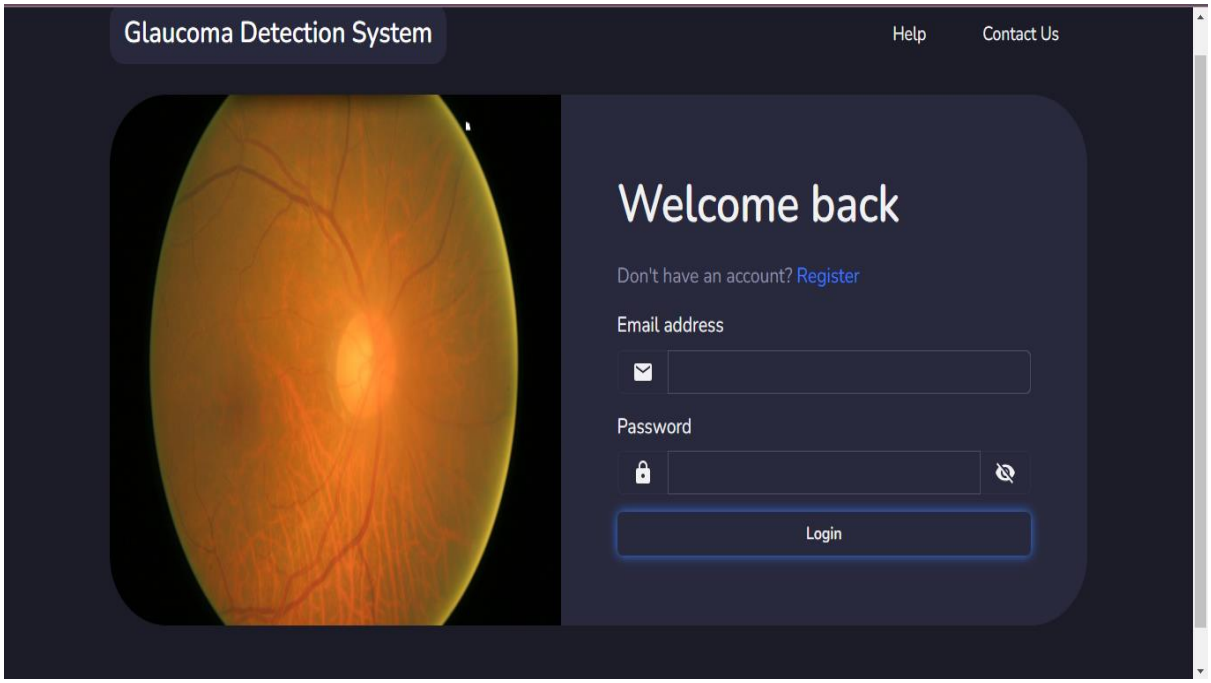


Figure 2 Register form for Glaucoma Detection System

Login Screen.

Once a user is registered on to the system, they sign in using their emails and passwords.



Analyze Screen.

Once a user is logged in, they are redirected to the detection page where they can upload the image that is to be analyzed, an ID is auto generated and the user gives a brief description about their condition. The image is then submitted for analysis.

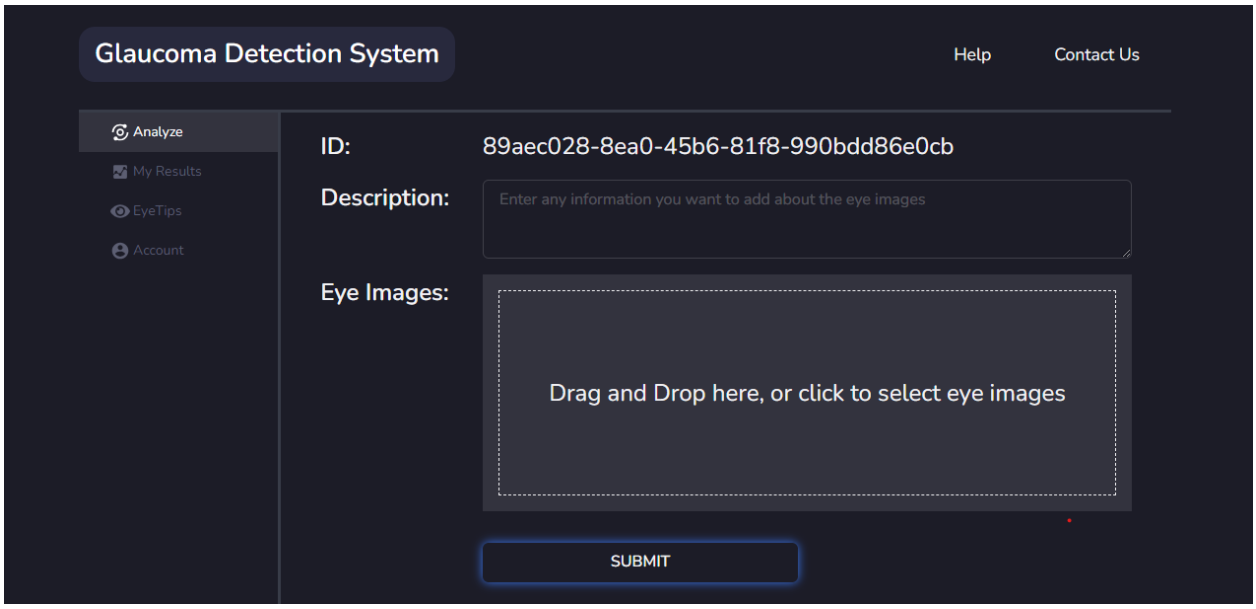


Figure 3: Image upload form on Analyze screen.

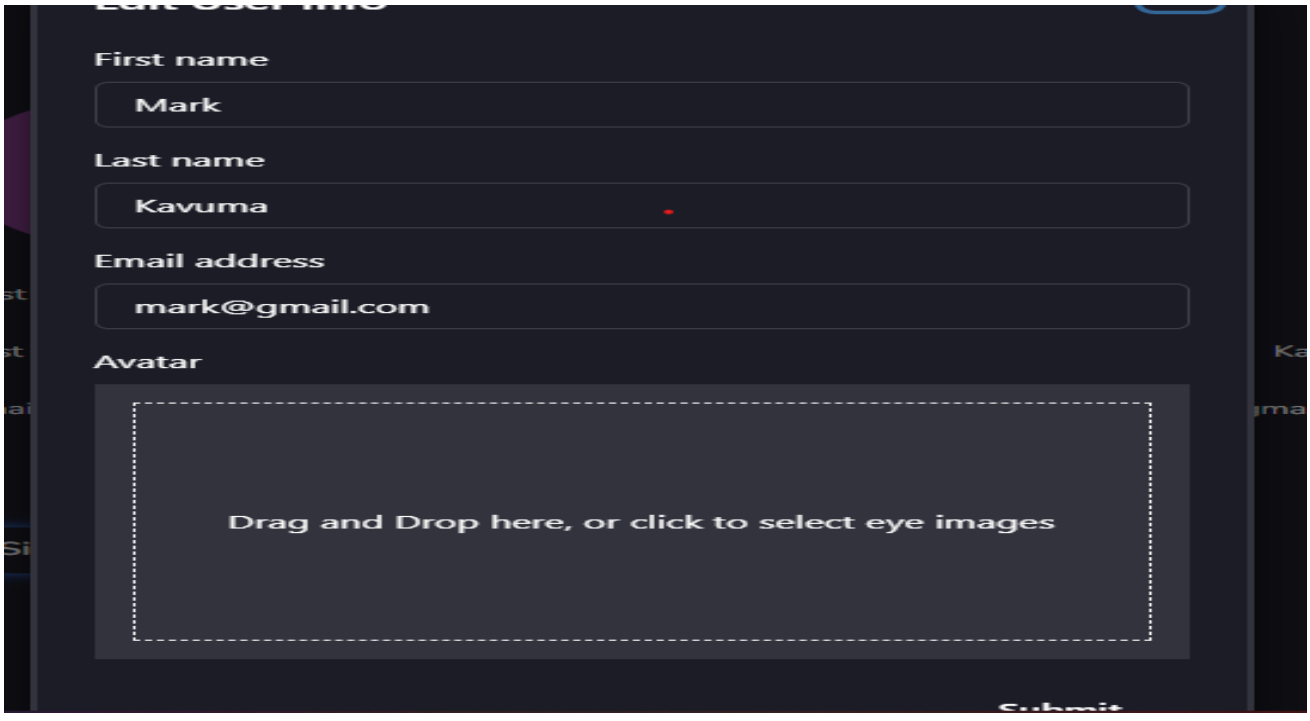


Figure 8: screen for updating user information

Results Screen.

Once the image has been analyzed, the results are displayed back to the user on the results screen and the user can view them in detailed form by clicking on the results record.

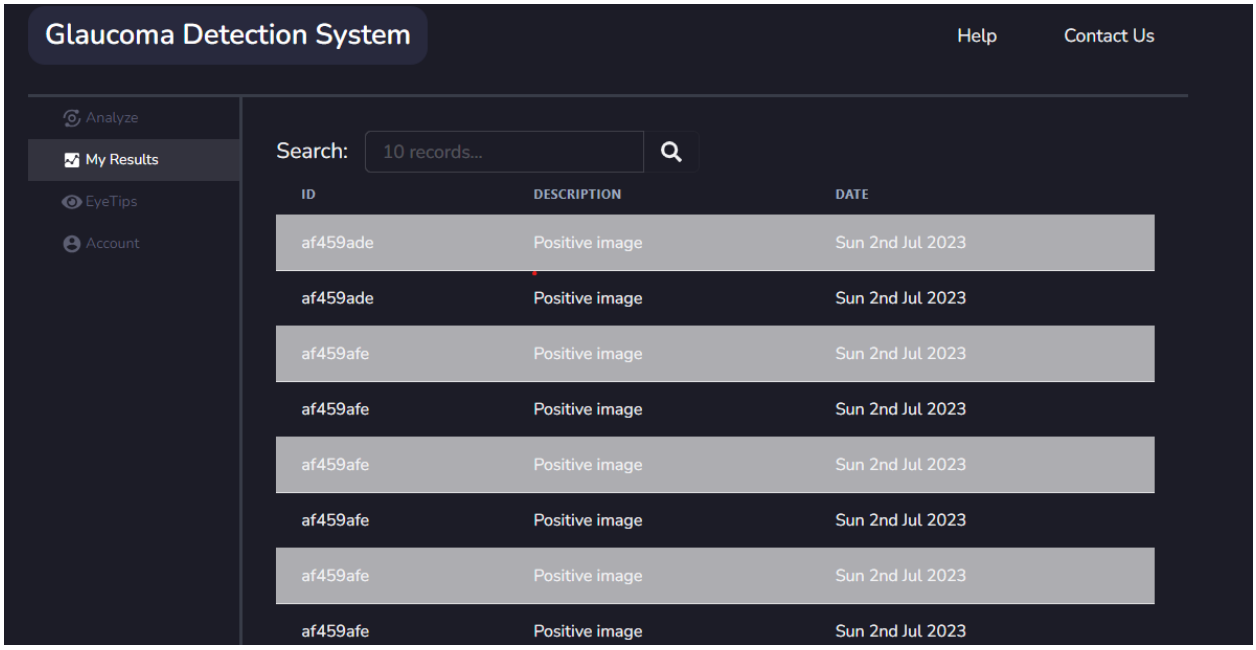


Figure 4: Results page for GDS system

Tips screen.

The tips are given to the user depending on the results obtained from the analysis of the image and displayed on this screen.

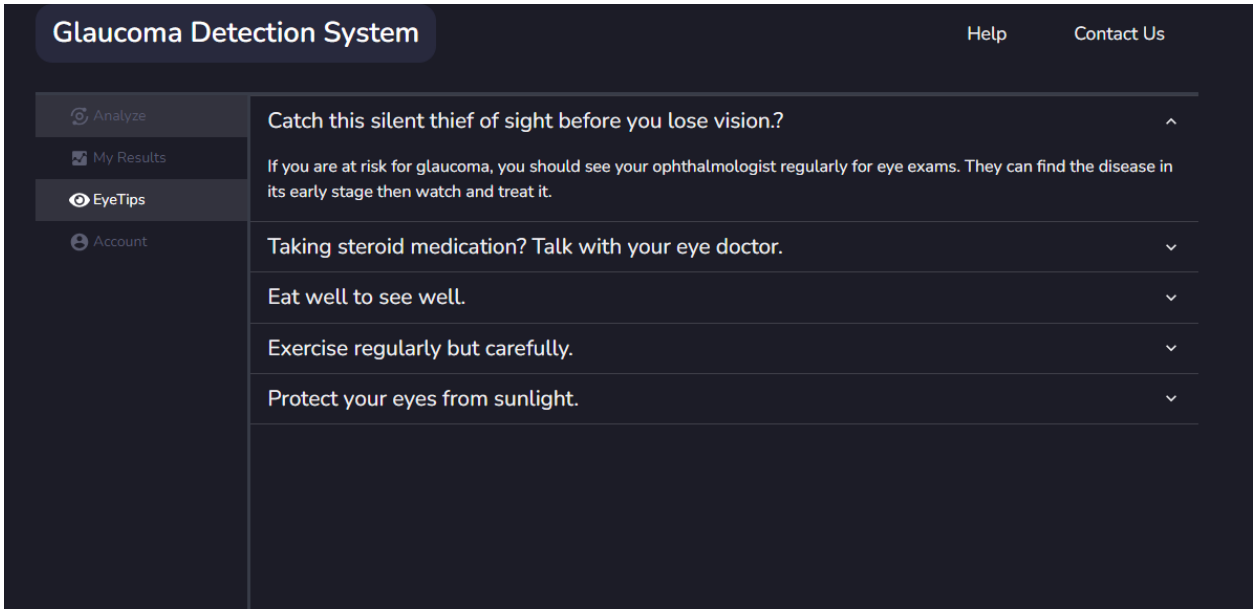


Figure 5: Eye tips screen.

Account screen.

This displays the user's information which can be edited using the edit button.

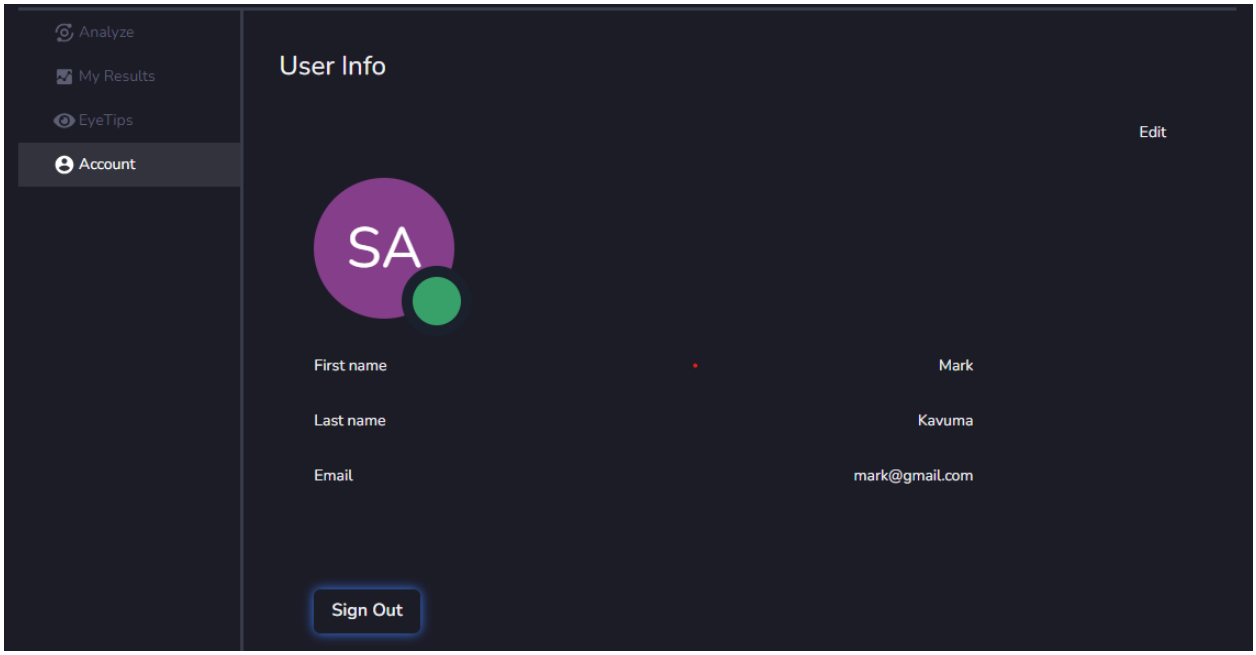


Figure 6: User Profile page for Glaucoma Detection System

Help screen.

This screen displays the frequently asked questions. The users can find answers to their questions and navigate the system effectively and efficiently.

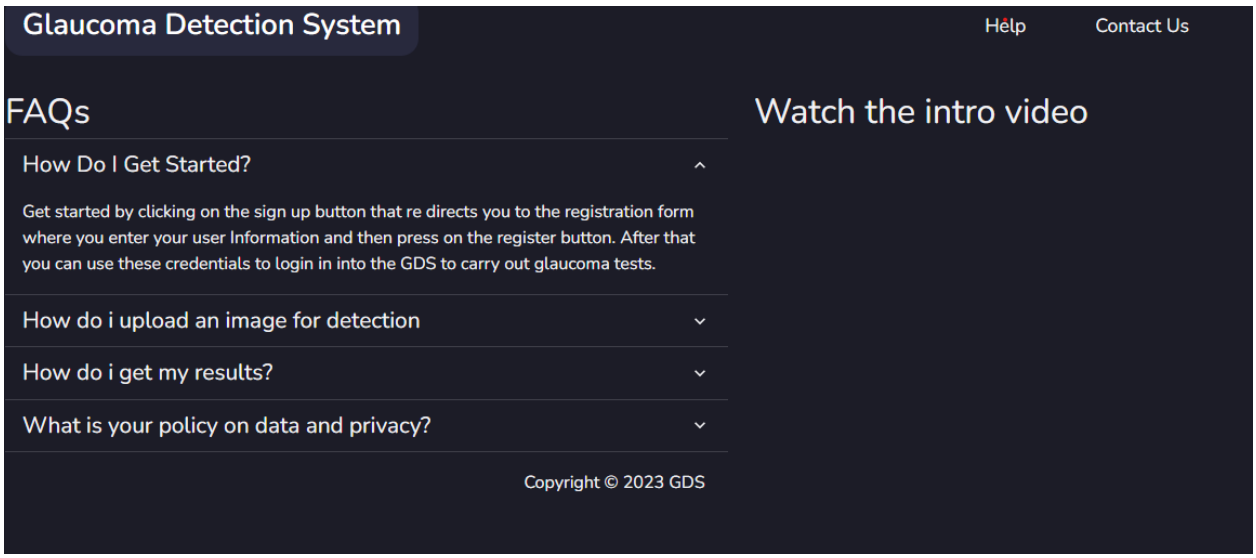


Figure 7: FAQ screen for Glaucoma Detection System