SOFTWARE REQUIREMENTS SPECIFICATION

for

Skin Disease Identification System

Version 1.0

Prepared by : Muhammad Burhan Ahmed(210287)

Submitted to : Miss Marium Sabir
Lecturer

April 5, 2025

Contents

1 Introduction

1.1 Overview

Skin diseases are among the most prevalent health issues, affecting individuals across all age groups worldwide. Early and accurate diagnosis is crucial to prevent complications and to enable timely treatment. However, access to dermatological expertise may be limited in emergency situations or remote regions. To address this gap, this project proposes a reliable and responsive web-based platform that leverages deep learning and image classification techniques to detect and classify skin diseases from dermatoscopic images.

The system employs a Convolutional Neural Network (CNN)-based model known as ResNet, trained on the HAM10000 dataset. The platform is designed to be user-friendly and responsive, accessible via both desktop and mobile devices, enabling patients to upload images of skin lesions and receive predictive results within seconds.

1.2 Purpose

The purpose of this Software Requirements Specification (SRS) document is to outline the functional and non-functional requirements of the Skin Disease Identification System. It aims to ensure that all stakeholders, including team members, evaluators and users have a clear understanding of the system's design, functionality and expected behavior. This document serves as a foundation for the system's development, testing and future upgrades. (If we use ML approach then uses decision trees, However DL Approach involves the use of Resnet-50)

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- **Project team members**: to understand design goals and implementation details.
- **Academic supervisors/evaluators**: to assess the project's scope, relevance and execution.
- **Future developers or maintainers**: for understanding the architectural decisions.

While the end-users (patients) interact with the system, they are not the primary audience of this document.

Suggested Reading Path:

- Sections 1–2: For general understanding.
- Sections 3: For detailed functional and system design.
- Sections 4: For testing and deployment strategies.

1.4 Project Scope

The Skin Disease Detection System is a web-based platform designed to assist individuals in identifying common skin conditions through image classification. Figure 1.1 shows the activity diagram of the system where users can register or log in, upload an image of their skin lesion, and optionally crop the image to focus on a specific region of interest. This cropped image is then analyzed by a deep learning model based on the ResNet architecture, trained on the HAM10000 dataset. The model classifies the image into one of several predefined skin disease categories.

Once the analysis is complete, a detailed diagnostic report is generated. This report becomes accessible to the user upon successful payment through the integrated payment gateway. The platform is responsive and optimized for both desktop and mobile usage. While the system provides initial insights based on automated classification, it is not intended to replace consultation with medical professionals. Instead, it aims to encourage timely medical attention by offering a quick and accessible preliminary assessment.

The scope includes:

- A responsive web interface for uploading and viewing results.
- Backend integration of the Deep Learning (DL) model.
- Payment integration for revenue generation.
- User-friendly design and guidance for non-technical users.
- Unit and system testing to ensure reliability.

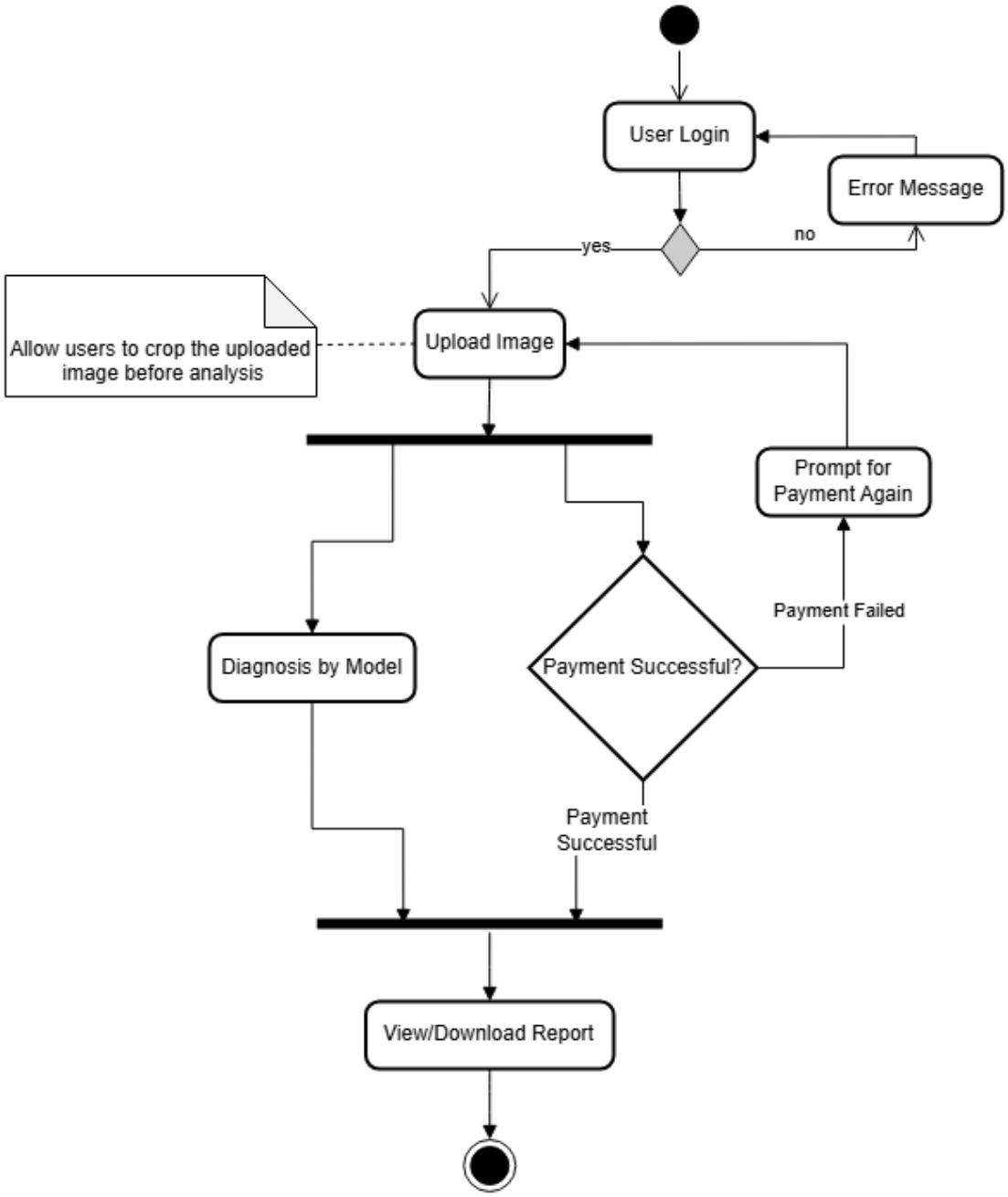

Figure 1.1: Activity Diagram for the Skin Disease Detection System

1.5 Glossary

Term	Definition
HAM10000	A large collection of multi-source dermatoscopic images of common pigmented skin lesions.
ResNet	Residual Neural Network – a type of deep learning model known for its accuracy and efficiency.
CNN	Convolutional Neural Network – a type of neural network commonly used for image recognition.
Deep Learning	A subfield of machine learning involving neural networks with many layers.
Model Inference	The process of using a trained model to make predictions on new data.
SRS	Software Requirements Specification – a document that describes the software's intended behavior.
UI	User Interface – the part of the system with which users interact.

Table 1.1: Glossary of Terms

1.6 References

- Tschandl, P., Rosendahl, C., & Kittler, H. (2018). The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, **5**, 180161.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*
- IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications.

2 Overall Description

2.1 Product Perspective

The Skin Disease Detection System is an AI-powered web application designed to assist users in identifying potential skin diseases from uploaded images. Unlike traditional diagnosis, which requires in-person consultation, this system allows users to receive preliminary classification results within seconds. The main objective is to enhance accessibility to skin disease detection using deep learning-based classification models. Figure 2.1 illustrates the context diagram for the Skin Disease Detection System.

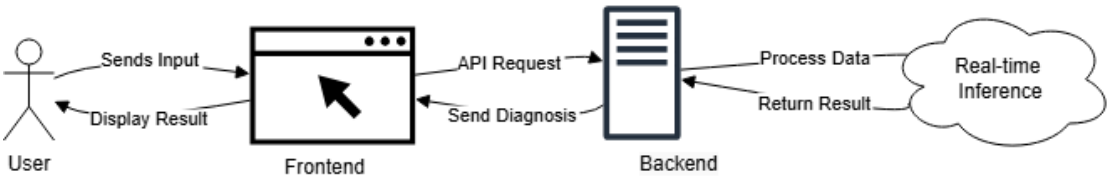

Figure 2.1: Skin Disease Detection System

2.2 Dataset Description

The table 2.1 summarizes the general specifications of the HAM10000 dataset, which is used for training and validating the proposed skin disease detection system.

Table 2.1: General Specifications of the HAM10000 Dataset

Attribute	**Description**
Dataset Name	HAM10000 (Human Against Machine)
Total Images	10,015 dermatoscopic images
Image Size	600 × 450 pixels (RGB)
Image Format	JPEG
Source	Images collected from populations in Austria and Australia
Class Labels	• Melanocytic nevi (nv) • Melanoma (mel) • Benign keratosis-like lesions (bkl) • Basal cell carcinoma (bcc) • Actinic keratoses (akiec) • Vascular lesions (vasc) • Dermatofibroma (df)
Dataset Size	Approximately 2.5 GB

2.3 User Classes and Characteristics

The system supports the following types of users:

- **General Users (Patients)** - Users who upload skin images for analysis.
- **System Administrators** - Manage system maintenance and model updates.

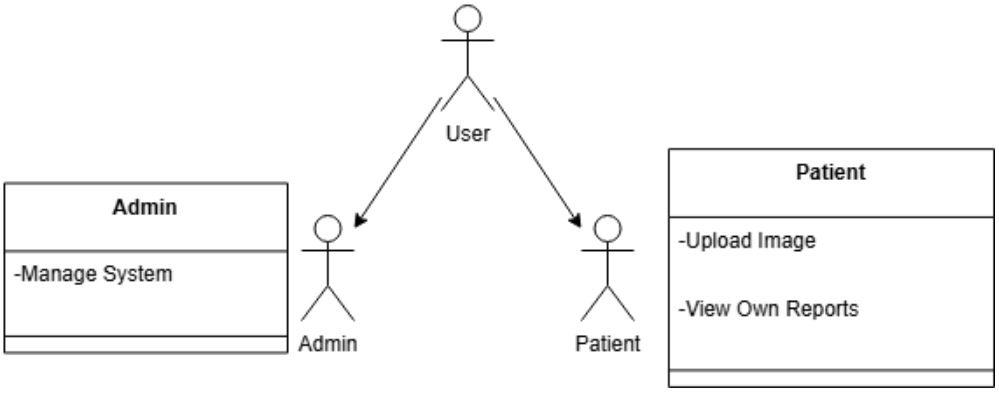

Figure 2.2: Types of Users

2.4 Product Functions

The core functionalities of the Skin Disease Detection System include:

- **Image Upload:** Users can upload images of affected skin areas for analysis.
- **Image Processing:** The system processes uploaded images using a deep learning model based on ResNet (CNN-based architecture).
- **Payment System:** Secure payment gateway integration for premium features or consultations.
- **Classification Results:** The system provides classification results along with confidence scores.
- **Medical Insights:** Displays basic medical information regarding the predicted skin disease.

Figure 2.3 shows the functions of the system

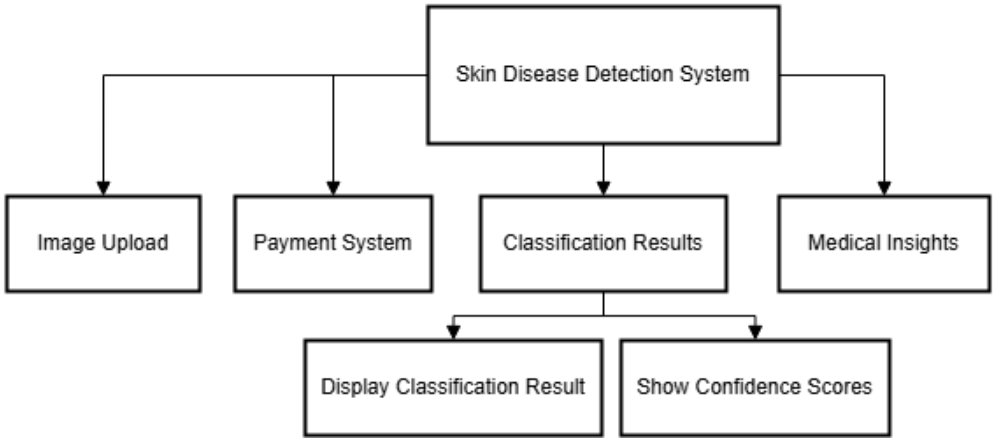

Figure 2.3: Functionalities of the System

2.5 Operating Environment

The system is designed to function in a cloud-based and web-accessible environment with the following components:

- **Frontend:** A responsive web application compatible with major browsers, including Google Chrome and Mozilla Firefox.
- **Backend:** JavaScript-based framework (Express.js) to handle API requests.
- **Model Deployment:** The deep learning model will be hosted on cloud servers such as AWS or Vercel.
- **Image Storage:** Cloudinary will be used for secure and efficient image handling.

2.6 Design

The system is designed to follow the user interaction flow, ensuring a smooth experience from login to receiving the analysis results. The steps involved in the user interaction are as follows and Figure 2.5 is the class diagram for this:

1. **User Navigation**: After logging in, the user navigates to the upload page.
2. **Image Upload**: The user uploads an image of the skin lesion for analysis.
3. **Payment Verification**: The system prompts the user for payment. If the payment is completed, the user proceeds; otherwise, they are prompted to make the payment again.
4. **Result Display**: After payment confirmation, the system displays the analysis results, including confidence scores and additional medical information.

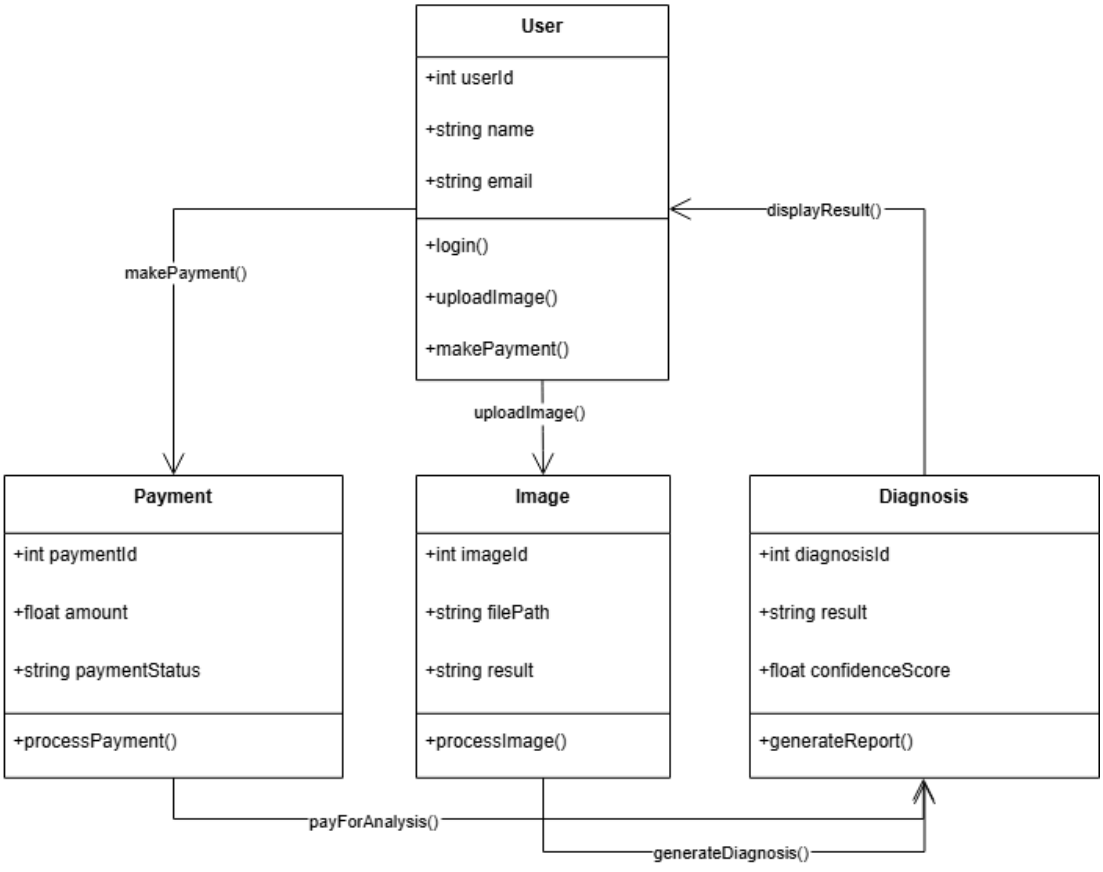

Figure 2.4: User Workflow

2.7 System Interfaces

The backend will provide a REST API for communication between the frontend and the AI model. Image processing requests will be sent to the model server, which returns classification results. Backend will generate a downloadable PDF report for the user. This report will include:

- User's Name
- Uploaded Image Preview (optional)
- Predicted Skin Disease
- Basic Medical Information
- Disclaimer that this is not a certified medical diagnosis

2.8 User Interfaces

The user interface will feature a minimalist design with a focus on simplicity. Key elements are shown in figure 2.5 and include:

- Image upload button with drag-and-drop functionality.
- Result display section with classification and confidence score.
- Mobile-friendly, responsive design.

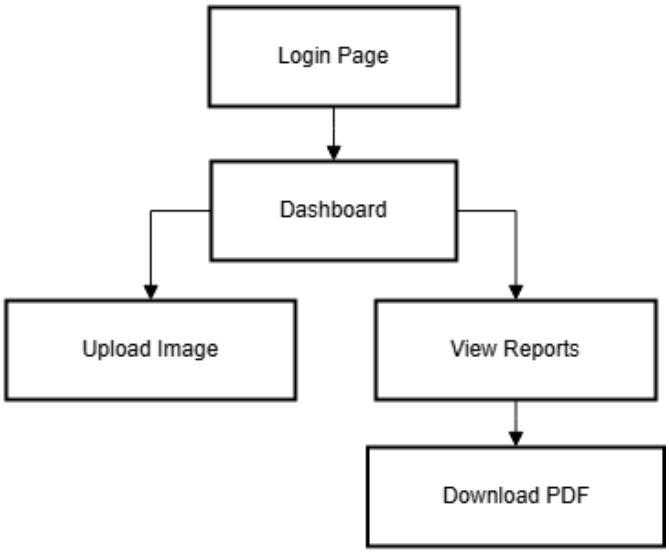

Figure 2.5: User Workflow

2.9 Software Interfaces

The following software tools and frameworks will be used to build and support the system:

- Developed using TypeScript, Next.js and styled with Tailwind CSS for a responsive and modern user interface.
- Either TensorFlow or PyTorch will be used for training and deploying the ResNet-based deep learning model for skin disease classification.
- The server-side logic and API endpoints will be implemented using Express.js or FastAPI, depending on performance and ease of integration with the ML model.
- MongoDB may be used for session and user data storage, depending on project needs.
- Cloudinary will be used to store and manage user-uploaded skin images securely and efficiently.
- Confluence will be used for maintaining project documentation and collaborative editing.

2.10 Enterprise Requirements

The project does not have specific enterprise requirements but aims to provide:

- High classification accuracy.
- Comprehensible web design for the user.

2.11 Constraints

- The real-time inference speed is dependent on the server's processing capacity and resources.
- The model's accuracy may vary across different skin tones and for rare medical conditions.
- The system is not intended to replace professional medical diagnoses and should be used as a supplementary tool.
- The website should be optimized for performance and user experience across different devices and platforms.

3 System Features

3.1 Description and Priority

The proposed system is a skin disease classification platform intended for patients to upload images of visible skin conditions and receive an AI-generated diagnosis. The system leverages a Convolutional Neural Network (CNN) model, specifically ResNet, to analyze the uploaded images and identify potential diseases. Users are allowed to crop the image to focus on the affected area. Following the image upload and cropping process, users are required to make a payment before viewing the diagnostic results, which are also downloadable in PDF format. Below are the key features of the system along with their respective priorities:

Feature	**Description**	**Priority**
User Authentication	Enables user registration and login functionality	Medium
Image Upload	User can upload images of skin conditions for analysis	High
Image Cropping	User can crop the uploaded image for better clarity before analysis	Medium
Disease Prediction	CNN-based ResNet model classifies the image and predicts skin disease	High
Payment Integration	Users must make a payment before accessing the results	High
Result Display and PDF Generation	Displays the results and allows downloading them as a PDF	High
Optimized Workflow	Ensures quick and efficient processing throughout the system	Low

Table 3.1: System Features and Their Priority

3.2 Functional Requirements

- **FR1:** The system shall allow users to register and log in using secure credentials.
- **FR2:** The system shall provide an interface for uploading skin disease images(PNG, JPG).
- **FR3:** The system shall allow users to crop the uploaded image before analysis.

- **FR4:** The system shall preprocess the image and use a ResNet CNN model for classification.
- **FR5:** The system shall require users to make a payment before revealing the diagnostic results.
- **FR6:** The system shall display prediction results after successful payment.
- **FR7:** The system shall generate and provide a downloadable PDF report of the results.
- User History (Optional): The system may allow users to view their previous diagnosis reports, if consent is given.

3.3 Non-Functional Requirements

- **Performance:** The system should provide diagnosis results within 4-5 seconds of image upload.
- **Usability:** The system should have an responsive and user-friendly web interface accessible to users with minimal technical knowledge.
- **Scalability:** The system should be scalable to support multiple concurrent users without performance degradation.
- **Availability:** The system should be available 99% of the time, excluding scheduled maintenance.
- **Security:** User authentication and data handling should follow secure practices.
- **Portability:** The system should be accessible from both desktop and mobile devices through standard web browsers.

3.3.1 Use Case 1: Login

- **Actors:** User
- **Description:** The user logs in to access diagnosis features and report history.
- **Preconditions:** The user must have a registered account.
- **Main Flow:**
  1. User enters email/username and password.
  2. System validates credentials.
  3. If valid, user is redirected to dashboard else ask again.
- **Alternative Flow:**

– If credentials are invalid, the system displays an error message.

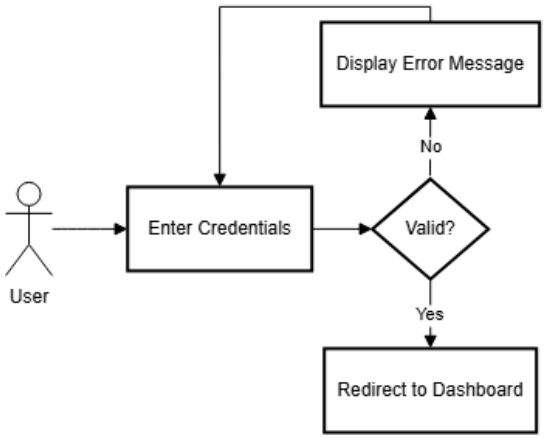

Figure 3.1: User case diagrams for Login

3.3.2 Use Case 2: Upload Image for Diagnosis

- **Actors:** User (Patient)
- **Description:** The user uploads an image of their skin condition for automated diagnosis.
- **Preconditions:** The user must be logged in to the system.
- **Main Flow:**
  1. User selects or captures an image.
  2. User uploads the image via the web interface.
  3. The system validates the image format and size.
  4. The image is passed to the model for analysis.
- **Alternative Flow:**
  – If the image is invalid (wrong format, size too large), the system prompts the user to try again.

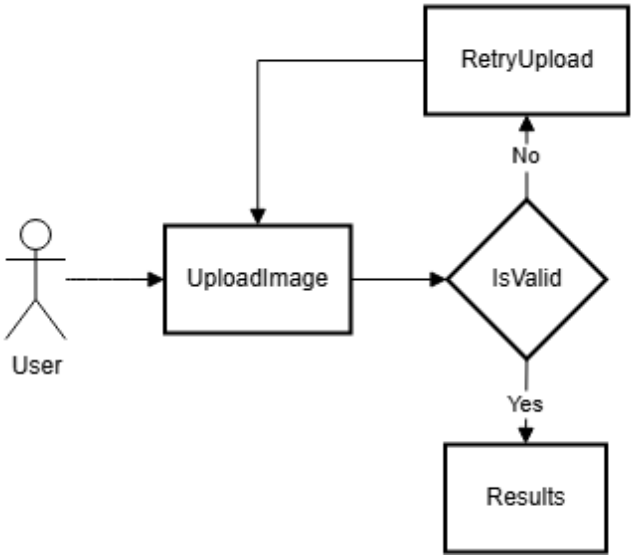

Figure 3.2: User case diagrams for Upload Image for Diagnosis

3.3.3 Use Case 3: View and Download Diagnosis Report

- **Actors:** User
- **Description:** After uploading an image and completing any required payment, the user can view the diagnosis result, including the confidence score and optionally download a full report in PDF format.
- **Preconditions:**
  - The user must be logged in to the system.
  - A valid image must have been uploaded and successfully processed.
  - Payment must be completed
- **Main Flow:**
  1. The system processes the uploaded image using a Deep learning model.
  2. A diagnosis result with a confidence score is generated.
  3. User makes the payment.
  4. The result is displayed to the user along with a brief summary or suggestions.
  5. The system generates a PDF report and provides an option to view or download it.
- **Alternative Flow:**
  - If the diagnosis prediction fails, the system notifies the user and prompts for a new image upload.

– If PDF generation fails, the system displays an error and asks the user to retry.

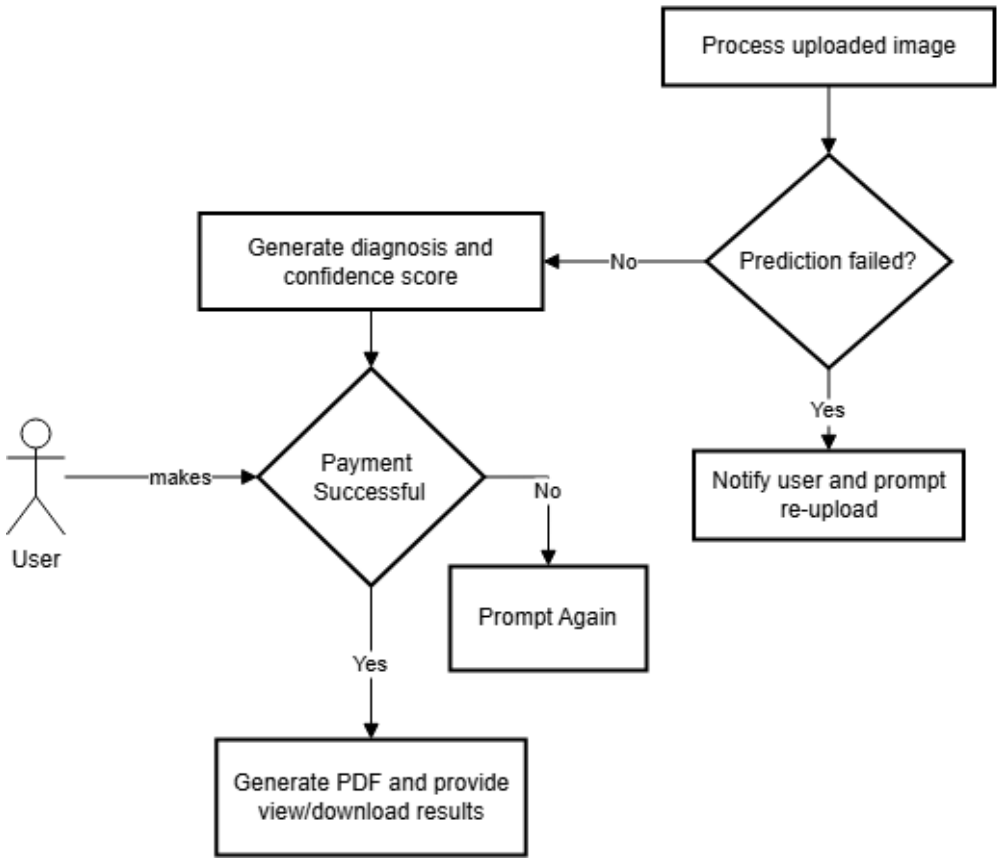

Figure 3.3: Use case diagram for viewing and downloading diagnosis report

4 Other Requirements

4.1 Analysis Team and Roles

The analysis team for this system is shown in figure 4.1 and consists of the following key roles:

- **Project Manager:** Responsible for overseeing the project, ensuring milestones are met and managing the team.
- **Deep Learning Specialist:** Develops and fine-tunes the machine learning model for skin disease prediction.
- **Frontend Developer:** Creates the user interface for the web application, ensuring it's intuitive and accessible.
- **Backend Developer:** Implements the server-side logic, APIs, and integrates the ML model into the system.
- **Quality Assurance (QA) Engineer:** Ensures that the system is tested for functional and nonfunctional requirements and meets quality standards.

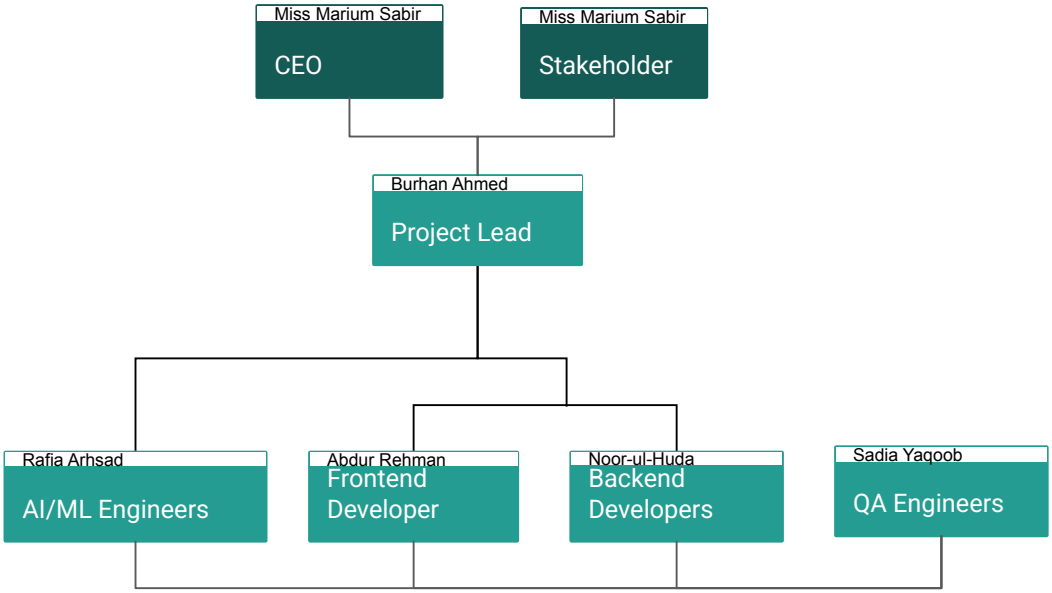

Figure 4.1: Team and Roles

4.2 Analysis Process

The system analysis process followed these key steps:

- **Requirements Elicitation:** The team conducted brainstorming sessions and engaged with stakeholders to gather both functional and nonfunctional requirements. This phase included research and feedback collection to refine the flow and identify additional requirements.
- **Requirement Analysis:** The collected requirements were analyzed for technical feasibility, potential revenue generation and time constraints. Requirements were then prioritized to ensure that the most critical features were developed first.
- **Documentation:** All gathered requirements and meeting discussions were documented in Confluence.

4.3 Dependency Analysis

The system has the following dependencies:

- **Internet Access:** Necessary for making disease identification requests.
- **User Authentication Service:** Relies on third-party authentication (e.g., Auth0) for user registration and login.

4.4 Performance Requirements

The system must meet the following performance requirements:

- **Response Time:** Image uploads and predictions should be completed within 5 seconds for 90% of requests.
- **Throughput:** The system should be capable of processing at least 100 requests per minute under standard load conditions.
- **Scalability:** The system should be capable of scaling horizontally to handle up to 100 concurrent users.

4.5 Security Requirements

The system must adhere to the following security guidelines:

- **Authentication:** Users must authenticate via secure login mechanisms (passwords, 2FA).
- **Data Privacy:** No personal data or images should be stored permanently without user consent.

4.6 Software Quality Attributes

The system must meet the following quality attributes:

- **Reliability:** The system should provide accurate results consistently, with minimal downtime.
- **Maintainability:** The system should be easy to maintain and extend, with clear code documentation.
- **Usability:** The system must have a clean, user-friendly interface for both novice and experienced users.
- **Scalability:** The system should handle an increasing number of users and requests.

4.7 Business Rules

The following business rules govern the behavior of the system:

- **Payment Model:** Users must pay a nominal fee for accessing diagnosis reports.
- **User Consent:** Users must provide explicit consent for storing and analyzing their data.
- **Result Accuracy:** The results should include a confidence score and users should be informed that the diagnosis is not a replacement for professional medical advice.