USMAN INSTITUTE OF TECHNOLOGY

Affiliated with NED University of Engineering & Technology, Karachi

Department of Computer Science

B.S. Computer Science

FINAL YEAR PROJECT REPORT

Batch-2020

TELECLINIC (SKIN)

By

Mahnoor	20B-012-CS
Ameema Farooq	20B-006-CS
Fizza Masooma	20B-120-CS
Muhammad Saad Hassan	18B-117-CS

Supervised by

Dr. Farhan Ahmed Karim

ST-13, Block 7, Gulshan-e-Iqbal, Abul Hasan Isphahani Road, Opposite Safari Park, P.O. Box 75300,

Karachi, Pakistan. Phone: 34978274-5; 34994305; 34982476; http://www.uit.edu

Submission Performa

Name (1) Mahnoor

(2) Ameema Farooq

(3) Fizza Masooma

(4) Muhammad Saad Hassan

Address (1) AS-05, Row-2, Block A (N.C.E.H.S), Gulshan – e – Iqbal, Karachi.

(2) R 530 block 17 FB Area Karachi.

(3) Gulshan – e – Iqbal, 13-D1 Sir Shah Suleman Road Own Heights Karachi.

(4) B-11, Gulshan – e – Ali, Block 19, Al – Noor Society, Federal B Area, Karachi.

TELECLINIC (Skin)

Dr. Farhan Ahmed Karim

This report is submitted as required for the Project in accordance with the rules laid down by the Usman Institute of Technology as part of the requirements for the award of the degree of Bachelor of **Computer Science/Software Engineering**. I declare that the work presented in this report is my own except where due reference or acknowledgment is given to the work of others.

Signatures of students	Date
(1)……………………………..	……………………..
(2)…………………………….	…………………….
(3)……………………………..	…………………….
(4) ……………………………..	…………………….

Acknowledgments

We would like to express our sincere gratitude to Allah the Almighty for giving us strength and guidance to successfully complete our goals. We also like to thank all those who contributed to the successful completion of this project.

First of all, we would like to thank our supervisor Dr. Farhan Ahmed Karim. His expertise and encouragement have been invaluable in the direction and execution of this project.

We thank the dermatologists and healthcare professionals who generously shared their expertise and insights and provided necessary input during the development of the web application and the integration of AI analysis.

We would like to acknowledge the understanding, support, and strength of our friends and families, whose encouragement and patience have always been a source of strength during the difficult phase of this endeavor.

We also thank the Usman Institute of Technology for providing essential resources and materials that facilitated the development of this work.

Last but not least, we are grateful to all participants and stakeholders who contributed their time and input, and ultimately contributed to the outcome of this project well and improved.

This work would not be possible without the collective efforts and support of these individuals and organizations.

Abstract

This report presents and implements a web application designed to facilitate remote access to skin health services. The main goal of this project is to provide a platform where patients can make online appointments with dermatologists, regardless of their location. When appointments are scheduled, patients can use a variety of communication tools to communicate with dermatologists such as video calls and text messaging and also allowing patients to provide skin images so that the AI model can diagnose the kind of skin disease. The focus of the platform is using an AI model that has been trained to recognize and diagnose three common skin diseases: Ringworm (Tinea Corporis), Onychomycosis (Nail Fungus), and Tinea Pedis (Athlete's foot), using Convolutional Neural Networks (CNN) to provide AI Model Diagnostic System. The project approach involves a rigorous development and testing phase to ensure that the web application is functional, secure and usable. User feedback and revisions help create a user-friendly approach and seamless experience for patients and dermatologists. Key findings from the project highlight the potential of AI-augmented telemedicine platforms to bridge the gap between remote populations and urban dermatologists, enabling dermatological health care to be accessible and improve its effectiveness.

Keywords: Dermatology, Telemedicine, Web Applications, AI Diagnostics, Remote Healthcare, Convolutional Neural Network (CNN), Skin Diseases.

TABLE OF CONTENTS

List of tables

List of figures

CHAPTER I

INTRODUCTION

1.1 INTRODUCTION

Using modern technology, TELECLINC (Skin) is an innovative project in the field of medical imaging and healthcare. This research seeks to revolutionize skin disease diagnosis by effectively integrating computer vision, machine learning, and image processing with a particular focus on Convolutional Neural Networks (CNNs) [1]. Understanding the need for early and accurate diagnosis, the approach aims to tackle the particular difficulties presented by skin disorders, whereby allowing people living in remote regions has access to experts in other regions. This enables for early and accurate treatment of skin diseases.

Skin Diseases pose a unique challenge to larger part of the healthcare sector, requiring innovative approaches that go beyond obstacles and provide diagnosis and treatment. An intuitive and effective platform in the form of TELECLINIC (Skin) is been developed. It is an intuitive web application which is developed in conjunction with the Skin Disease Detection System. It functions as a complete solution that enables dermatologists and patients alike, to enable remote skin disease diagnosis and treatment. The project's goal is to eliminate geographic barriers and streamline patient-dermatologist interactions via TELECLINIC (Skin), hence providing on-time and effective care.

In addition to technological innovations, the program recognizes the ongoing challenges faced by those seeking health care, especially when it comes to choosing the right provider. It also acknowledges the historical trend of online physician visits and referrals and critiques the shortcomings of current websites. Many of these websites do not provide comprehensive information about physicians, such as their specialties and success rates. Furthermore, lack of physician participation and the high cost of online chat hinder the effectiveness of these meetings.

To overcome these obstacles, the proposal proposes the development of a state-of-the-art web application that will make it easier for patients to find the right medical professional. Available on any device including a web browser, this web application provides user-friendly communication through information and communication technologies. Intuitively, patients can peruse a carefully selected list of physicians, each complete with detailed information about their expertise and previous results. The web application offers a way to streamline the search process by symptoms distribution. Additionally, because the information can only be safely transmitted and viewed by authorized healthcare professionals, patients do not need to submit written reports. This assures a complete and safe clinical experience by reinforcing patient privacy and allowing physicians to make changes to documentation.

1.2 AIM OF PROJECT

Using modern technology, TELECLINIC (Skin) hopes to revolutionize skin health care. This study aims to provide a means of accurately detecting skin diseases at their earliest stages, especially in remote areas without access to primary health care.

This project aims to develop and implement a web-based telemedicine platform equipped with AI diagnostic capabilities to increase access to dermatology healthcare services This platform is designed to allow patients to make appointments for virtual appointments with dermatologists, through various communication tools, including videocall.

and text messaging. The web application also has an AI CNN [1] to detect and diagnose common skin diseases like Ringworm (Tinea Corporis), Onychomycosis (Nail Fungus), and Tinea Pedis (Athlete's foot). Leveraging AI technology the project seeks to improve efficiency and accessibility of dermatology consultations, especially for remote people.

1.3 PROBLEM STATEMENT AND SOLUTION

Multiple variables, including geographic barriers, inaccessibility to dermatologists, and reliance on diagnostic methods, make skin disease a significant health burden, especially in isolated areas where it is difficult for traditional health care systems to provide rapid and accurate diagnosis. This project seeks to address these issues by developing a comprehensive system for the diagnosis and management of peripheral skin diseases.

The importance of accurate and accessible dermatology treatment largely determines the scope of the project. The project addresses these issues by integrating state-of-the-art technologies such as image processing, computer vision and machine learning. Acting as a link between patients and dermatologists, the TELECLINIC (Skin) facilitates remote consultations. Creative web app further improves accessibility by adding a secure way to post records and provide comprehensive physician information. Establish an integrated healthcare system, some of this work is to build a Skin Disease

Detection System, TELECLINIC web app will reduce the gap between patients and dermatologists, increase diagnostic accuracy, and speed up patient-dermatologist interactions.

Following are the problem identified by the stakeholders:

1.3.1 PATIENTS

1. Many patients, especially in rural or underserved areas, face significant barriers to accessing dermatology care due to lack of access to their specialist locally.

2. Patients often wait longer to see dermatologists, and diagnosis and treatment take longer.

3. Traveling to urban primary care facilities can be quite time-consuming and expensive, imposing a significant burden on patients.

4. Patients may feel uncomfortable talking about or actually disclosing skin conditions, delaying care seeking.

5. Failure to seek immediate advice and respond quickly can exacerbate skin conditions that require timely intervention.

1.3.2 DERMATOLOGISTS

1. Dermatologists in urban areas often have too many cases, leading to burnout and reduced quality of care.

2. Handling administrative tasks, scheduling, and tracking manually can be time-consuming and reduce patient care.

3. Dermatologists may not have access to advanced diagnostic tools that can help make a quick and accurate diagnosis.

4. Obtaining a complete patient history and image can be difficult without a well-designed digital platform, which can lead to incomplete diagnosis.

5. Ensuring regular follow-up and monitoring of patient status is difficult, especially for those who live far away or have mobility issues.

1.3.3 HEALTHCARE SYSTEMS

1. Inadequate resource allocation and uneven distribution of dermatology services, often resulting in overburdened urban areas and underserved rural areas.

2. The costs associated with individual consultations, with businesses and professionals, are substantial, making it difficult to provide comprehensive skin care.

3. Traditional health systems are struggling with scaling services to meet rising demand, especially in specialty areas like dermatology.

4. Ensuring patient data management, privacy and security while providing seamless access to healthcare providers is a major challenge.

5. Integrating AI and other advanced technologies into existing healthcare systems is difficult and often faces resistance due to cost, training and adaptation barriers.

6. Healthcare systems often struggle to effectively communicate and educate patients about the importance of skin care and available services.

The project aims to provide affordable, effective, and high-quality skin health care for all involved parties by resolving these issues through a web-based telemedicine platform that combines AI and diagnostics.

1.4 PROJECT DESCRIPTION AND OBJECTIVES

The objective of this project is to develop and implement an advanced web-based telemedicine system designed to increase access to skin disease health services. The platform is specifically designed to facilitate long-distance engagement with dermatologists, addressing the challenges of geographic barriers and lack of access to primary care.

Patients can schedule appointments with dermatologists and consult via a variety of communication tools, including video calls, and messaging. A key feature of the platform is patient potential post pictures of their skin condition for professional evaluation.

To further improve the evaluation process, the platform includes an AI-powered diagnostic tool. The tool uses CNN [1] to detect and diagnose three common skin conditions: Ringworm (Tinea Corporis), Nail Fungus (Onychomycosis), and Tinea Pedis (Athletes foot) and the AI model provides preliminary analysis provides to assist dermatologists with greater diagnostic accuracy.

The development process requires extensive testing and optimization to ensure that the web application is functional, secure and user-friendly. User feedback is central to the iterative process, ensuring that the platform adequately meets the needs of patients and dermatologists.

1.4.1 OBJECTIVES

1. To create a web-based system that also allows patients to schedule remote appointments with dermatologists, geographic barriers are removed.

2. Integrate multiple methods of communication, including video calling and messaging, to provide seamless communication between patients and dermatologists.

3. Patients can upload images of their skin condition for detailed review by dermatologists.

4. To use a CNN [1] based AI Model that can identify and diagnose ringworm, nail fungus and tinea pedis, to provide dermatologists with initial diagnostics to aid diagnosis.

5. Conduct a rigorous testing phase to ensure web application functionality, security and usability, to ensure a reliable and secure experience.

6. Continue to improve the platform by incorporating feedback from users, improving the interaction, and enhancing the overall user experience based on real-world applications on the snow.

7. Demonstrate the potential of AI-enabled telemedicine platforms to bridge the gap between remote populations and urban dermatologists, thereby improving the accessibility and the effective has improved.

1.5 EXPECTED OUTCOME

Expected outcomes of a project generally include both tangible and intangible results expected from the completion of a project. These outcomes often include the achievement of specific goals, such as increased productivity, increased customer satisfaction, or financial returns. Tangible outcomes may include the delivery of a new product or service, completed infrastructure, or the implementation of a new system. Intangible outcomes can include improved team morale, strengthened stakeholder relationships, or enhanced brand reputation. In addition, expected results are often formulated in terms of measurable objectives, such as achieving a certain percentage increase in sales, reducing costs by a certain amount, meeting project deadlines role or all expected outcomes must be consistent with the strategic objectives of the project and provide clear benefits for the organization and stakeholders.

1.5.1 FOR PATIENTS

1. Patients, especially those in underserved or remote or unaccessible areas, will have better access to dermatologists, reducing the need to travel long distances.

2. The platform will facilitate faster scheduling of appointments and consultations, allowing faster diagnosis and treatment of skin conditions.

3. Patients will benefit from convenient remote counseling, which can be done from the comfort of their homes. This reduces the stress and hassle associated with visiting in person.

4. Patients can comfortably and privately discuss their conditions through digital communication tools, encouraging them to seek appropriate medical advice.

5. By providing an initial diagnosis with an AI diagnostic tool, patients can receive a faster initial diagnosis, potentially speeding up treatment.

1.5.2 FOR DERMATOLOGISTS

1. It will be a streamlined platform for managing dermatology appointments, patient records and follow-up, reducing administrative burden and increasing focus on patient care.

2. AI diagnostic tools will help dermatologists provide initial diagnosis and thus support more accurate and effective diagnosis.

3. By facilitating remote communication, dermatologists can better manage their caseload, which can reduce burnout and improve job satisfaction.

4. The platform will enable better communication with patients through a variety of digital tools, improving both patient-provider relationships and care follow-up.

1.5.3 FOR HEALTHCARE SYSTEMS

1. The platform will help distribute dermatology care more equitably across communities, improve utilization of healthcare resources and reduce congestion in city centres.

2. By reducing the need for physical interventions and personal visits, the health care system can reduce the costs associated with the provision of skin care.

3. The telemedicine platform can easily be scaled to meet the growing demand for dermatology services, making it a sustainable solution for future healthcare needs.

4. Using a digital platform, patient data can be handled more efficiently and securely, ensuring better privacy protection and easier access for healthcare professionals.

5. Successful integration of AI into the research process will demonstrate the potential of advanced technology to improve healthcare delivery, and set a precedent for future innovation.

6. The platform can be used to educate patients about skin health and the importance of timely care, leading to better health outcomes and increased awareness.

The use of this web-based telemedicine platform with AI diagnostics is expected to bridge the gap between remote people and dermatologists This will lead to more convenient dermatology health services, it works well and is good. By leveraging technology, the platform will create a more equitable healthcare environment, ensuring that all patients, regardless of location, can access effective skin care in a timely manner.

1.6 METHODOLOGY

A project's methodology refers to the systematic approach to achieving its objectives, including the specific methods, tools, and techniques used throughout its lifecycle Typically beginning in the planning phase in, including defining scope, setting objectives, and developing a detailed project plan. This is followed by the execution phase, where resources are allocated, transactions are executed, and milestones are executed as planned. Key features is typically risk management, quality assurance, and communication strategies to ensure that all stakeholders are informed. Regular reviews and evaluations are conducted to monitor progress, make adjustments, and ensure that the project is aligned with its objectives. The methodology includes a closing phase, where deliverables are finalized, and detailed assessments are conducted to capture lessons learned and evaluate overall performance.

1.6.1 RESEARCH

1. Conduct comprehensive research on existing telemedicine platforms, AI diagnostics in dermatology, and CNN [1] applications in medical imaging. Identify best practices, potential challenges and gaps in current solutions.

2. Interact with patients, dermatologists and healthcare professionals through interviews, surveys form and feedback focus groups to understand, visualize the work flow and their pain point, needs and preferences. Gather detailed platform requirements.

3. Find legal and regulatory requirements for telemedicine and Abased diagnostic tools. Ensure compliance with healthcare standards such as General Data Proctection Regulation (GDPR) [2] and Health Insurance Portability and Accountability Act (HIPAA) [3].

4. Evaluate the feasibility of integrating AI diagnostics into telemedicine by considering factors such as available data, computing resources, and existing technology infrastructure.

1.6.2 DEVELOPMENT

1. Perform all web application design, including user interface, backend processing, database management, and AI integration. Create a detailed technical description and data flow diagram.

2. Collect highly annotated images for three target skin diseases (ringworm, nail fungus, and tinea pedis). First process the data and create a CNN [1] model using a framework like TensorFlow [4] or PyTorch [5]. Train the model on the labeled data, ensuring that it achieves high accuracy in detecting the specified conditions.

3. Build web applications using modern web development technologies like HTML, CSS and JavaScript for the front-end and PHP for the back-end. Basic functionality such as user registration, appointment scheduling, use of communication tools (video calls, messaging), and image upload capabilities.

4. Add the trained CNN [1] model to the web application. Developing APIs to enable seamless communication between the AI research tool and the rest of the platform. Ensure that the AI delivers a clear initial diagnosis to dermatologists.

1.6.3 TESTING

1. Unit testing individual components of the platform ensures that each function is working properly. Use automated testing tools to facilitate this process.

2. Perform integration testing to ensure that all components of the platform work together as expected. Focus on the interaction between the web application and the AI analysis tool.

3. Engage a group of users, including patients and dermatologists, to test the platform in real-world scenarios. Collect information about usage, use, and overall experience. Identify if there are any issues or areas for improvement and address them.

4. Conduct comprehensive security testing to ensure the platform meets all regulatory requirements for patient data security. Implement procedures to protect against data breaches, unauthorized access, and other security risks.

1.6.4 DEPLOYMENT

1. Run the application in a staging environment that reflects the production process. Perform final testing to ensure that the platform is ready for actual deployment without disrupting existing systems.

2. Provide comprehensive training for dermatologists and administrative staff to familiarize them with the platform. Develop implementation manuals and supporting documentation.

3. Take the platform into a live production environment. Monitor the deployment process closely to ensure a smooth transition and resolve any issues that arise immediately.

4. Provide ongoing support and maintenance to address any technical issues, provide updates, and add new features based on user feedback. Monitor system performance and security regularly to ensure proper operation.

1.7 FUNCTIONALITY

Functionality refers to the set of functions and features that a system or application is designed to perform. It includes specific tasks, the ease with which users can perform those tasks, its functionality and effectiveness. In the case of software, the functionality includes a range of functions and services provided to the user, such as data processing, user interfaces, and other integration systems or platforms. Not only does the product fulfill its intended purpose but it does so in a capable way reliable, user-friendly and efficient. This is an important aspect of product development and development, as it directly affects user satisfaction, performance and overall experience. Ensuring robust performance requires comprehensive design, rigorous testing, and constant feedback to refine and enhance the capabilities of a product or system.

1.7.1 USER REGISTRATION AND AUTHENTICATION

1. Allow users (patients and dermatologists) to create an account using a flexible registration method. Collect important information such as name, contact details and professional credentials of dermatologists.

2. Use a secure login system with email/username and password.

3. Enable users to easily manage their profiles, update their personal information, change their passwords, and set preferences.

1.7.2 APPOINTMENT SCHEDULING

1. Provide a user-friendly approach for patients to schedule appointments with dermatologists. Display available times and let users choose their preferred day and time.

2. Allow patients and dermatologists to check, reschedule, or cancel appointments.

3. Maintain a history of prior appointments of patients and dermatologists, including details and consultation results.

1.7.3 COMMUNICATION TOOLS

1. Include secure video conferencing for real-time discussions between patients and dermatologists.

2. Use a secure messaging system for text-based communication. Allow users to exchange messages, ask questions, and share updates.

1.7.4 IMAGE UPLOAD AND ANALYSIS

1. Allowing patients to upload high-definition images of their skin.

2. Use an integrated CNN [1] model to analyze the uploaded images and provide preliminary recommendations for three common skin conditions: Ringworm (Tinea Corporis), Nail Fungus, and Tinea Pedis (Athlete's foot).

3. Create an AI assessment report and present it for review by dermatologists. Emphasize the reliability of the AI and any known conditions.

1.7.5 DERMATOLOGIST INTERFACE

1. Providing dermatologists with a comprehensive dashboard to manage their patients, view appointment schedules and access patient histories.

2. Allow dermatologists to create and deliver electronic prescriptions to patients, ensuring compliance with regulatory standards.

By incorporating this functionality, the web application aims to provide patients and dermatologists with a comprehensive and seamless telemedicine experience, increasing the accessibility and quality of dermatology care.

CHAPTER II

BACKGROUND AND LITERATURE REVIEW

2.1 INTRODUCTION

This project TELECLINIC (Skin) is divided into two – parts. First part consists of a web application designed for individuals living in remote areas. The second part is the Skin Disease Detection System, which is responsible for identifying foot fungus, nail fungus and ringworm, among other skin diseases. People who live in remote areas and cannot afford good, qualified healthcare can find ease in the web application. As a result, this application offers a platform for communication between the dermatologists and patients. Patients can select any male or female dermatologist of their choosing from the appointment pannel, which displays a variety of dermatologist options. As a result, once the dermatologist has been selected, the patient can upload his / her skin image, and the Skin Disease Detection System which has been trained using a Convolutional Neural Network (CNN) [1] model can diagnosis the condition in that image. Once the image has been diagnosed, the dermatologist will review it and, based on his analysis, prescribe a test or treatment. The patient will pay the fee at the payment panel. Thus, the web application allows for communication between the dermatologist and the patient. In many rural areas of today's world, access to quality healthcare and hospitals for skin diseases is limited. Therefore, the web application makes life easier for both dermatologists and patients. Dermatologists can use it from the comfort of their homes, and patients can consult with any dermatologist of their choosing. The dermatologist will then prescribe treatment and tests through the platform after the patient pays a fee.

2.2 BACKGROUND AND LITERATURE REVIEW

The Skin Diseases Detection System is an innovative project that combines medical imaging, computer vision, and machine learning for accurate and early detection of skin diseases. Prior research has shown promising results with CNNs [1] and Region-Based Convolutional Neural Networks (RCNNs) [1], which can differentiate various skin diseases based on images. Although there have been significant advances in medical treatment such as lasers and photonic. These technologies have made diagnostics of skin diseases much faster and more accurate, but still these are expensive [6]. By using computer vision, the problem can be solved but is is expensive and inaccessible to people. This project aims to fill these gaps by creating a sophisticated system capable of detecting and classifying different skin diseases. It aligns with previous work in machine learning, libraries, image processing, and classification techniques.

A growing body of evidence is discussed in some telemedicine research papers to support the validity of telemedicine in dermatological diagnosis and treatment recommendations, as well as in providing dermatological care to underserved areas [7]. While they are developing the video conferencing technology, which funcnatility is to assess skin tumors. Pervious studies on this type of application have focused on telemedicine applications, which help patients and dermatologists communicate. Image processing techniques are also used for skin disease detection, which facilitaties patient treatment for dermatologists. Roughness and acne are determined after patients submit pictures of their skin [8]. According to a different study based on past telemedicine research, telemedicine is growing in COVID-19 during emergency situations. In some research papers it mentions that family dermatologists can easily access specialists using telemedicine enabling them to monitor their patients clearly.

Within the telemedicine domain, the focus is on leveraging mobile technology to facilitate communication via text, images, videos and other channels. Ultimately, there was a reduction in the risk of COVID-19, and patients who use WhatsApp and SMS for round-the-clock support during treatment continue to receive dermatological services [9]. In a different study, the objective was to assess how well telemedicine performed penicillin allergy skin testing. It assesses how telemedicine affects maximizing and enhancing allergy resource accessibility. In a particular study, the diagnosis of a patient's condition depended heavily on physical examinations. Imaging studies, a physical examination, a lab test, and the patient's medical history must all be included [10]. In some research paper it also notes that family physicians can closely monitor their patients using telemedicine to easily access specialists.

The article highlights that telemedicine should be used in addition to in-person dermatologist visits rather than as a substitute, and it explores the advantages and difficulties of practicing medicine remotely.

This project is about development of a web application for people living in remote areas who cannot afford to see highly qualified medical professionals. This application will be made available online so that dermatologists can communicate with patients living in remote areas. As supervised learning is what this project focus on, but first the dataset is labeled. The K-NN algorithm has been used to work on unsupervised learning in numerous earlier studies. Following labeling, the skin image data set to train a CNN [1] model. The model can identify every pattern in an image.

In previous research papers CNN [1] has been implemented and Support Vector Machine (SVM), etc. But this project implement’s CNN [1] model as they are more powerful than other classification algorithms because it has the ability to learn more complex features from images as all steps of image processing will be applied like Image-Preprocessing, Segmentation, Thresholding and Disease Classification. Then after training the model, the system has the ability to diagnose a skin disease. So patient will provide an input of his / her skin image and the system will diagnosie by extracting the features and after diagnosing the type of skin disease for the image, a report is send to the dermatologist along with the actual skin image uploaded by the patient. The dermatologist will prescribe treatment and test according to disease. While patients can select any dermatologist of their choosing from the appointment’s pannel. The patients can sort dermatologist based on their experence, qualification, gender, specilzation, and etc. Following the appointment, they will consult the dermatologist and pay the fee of their appointment. Thus, the web application will make life of thousands of patients easier, specially for those who live in rural areas. Where high quality medical facilities are not available. Therefore, patients can use this web applications to connect themselves to highly qualified dermatologist from the comfort of their own home and can consulte with male or female dermatologists online. In essence, it is a web application designed to facilitate communication between patients and dermatologists.

2.3 PREVIOUS WORK

Prior research and development in the field of skin disease detection have primarily focused on the use of computer vision, machine learning, and image processing techniques. Some tried to detect Eczema using computer vision [11]. Some used deep learning to train their model [12]. Some tried to detect six different kinds of skin disease [13]. Various studies have explored the use of Convolutional Neural Networks (CNNs) [1] and Region-Based Convolutional Neural Networks (RCNNs) [1] for automated skin disease diagnosis. These studies have shown promising results in the accurate identification of skin diseases and distinguishing between various types. These investigations have yield encouraging findings regarding the precise diagnosis and differentiation of skin conditions.

Since the web application is launched for the users living in remote areas, prior research on this kind of application has concentrated on telemedicine application that facilitate communication between patients and medical professionals. To make it easier for dermatologists to treat patients, skin detection is also carried out using image processing techniques. Once patients provide skin images, roughness and acne are obtained. Another study about earlier telemedicine research shows that, in COVID-19, telemedicine is expanding during emergency situations. In the field of telemedicine, they are concentrating on using mobile technology as a means of communication through text, photos, videos among other mediums. In the end, the risk of COVID-19 was decreased, and dermatological services are still provided to patients who use WhatsApp and SMS for round-the-clock assistance during treatment. They treat both acute and long-term illnesses. The purpose of the study was to determine whether telemedical wound care could be implemented with a new generation of smartphones that have built-in cameras [14]. The study discovered that mobile phone images can be used for remote evaluation of leg ulcers, and that there is an agreement of high degree between evaluations made remotely and in person. Most of the time, participants comfortalbely made diagnosis based solely on the pictures, and the image quality was deemed to be good in most of the cases and felt confident and comfortable about the system.. This study shows how telemedical wound care can lower costs while enhancing quality of care in healthcare systems [10].

2.4 ALGORITHMS

The primary algorithms used in many research papers include CNNs [1] for image classification and Region-Based Convolutional Neural Networks (RCNNs) [1] for object detection [15][16]. Ther are many research papers where the authors used a different algorithm such as Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Random Forest (RF) [17]. While some only used SVM [18]. Similar to previous research papers, the Pretrained CNNs algorithms are employed in conjunction with the AlexNet model. Its primary purpose is to extract features from resized color images of skin diseases. An additional algorithm for classifying extracted features is the Multi Support Vector Machine algorithm.

2.5 USAGE OF ALGORITHMS

Algorithms are used to perform the following tasks which enable the AI Skin Disease Detection System to perform accuratly.

2.5.1 FEATURE EXTRACTION

Feature extraction is an important step in data processing, where raw data is transformed into measurable traits or features. These features are used to simplify the data and make it more manageable for analysis. For image processing, feature extraction is the identification of key characteristics from images that can help distinguish between different classes or objects.

a. DIMENSIONALITY REDUCTION

Feature extraction reduces the number of variables considered, enabling more efficient data manipulation by retaining important information.

b. TECHNIQUES

Typical methods include edge detection, texture analysis (e.g., gray level co-occurrence matrix), and color histogram analysis. Machine learning uses advanced techniques such as Principal Component Analysis (PCA) and CNNs [1].

c. PURPOSE

The main objective is to extract the most relevant information from the data in order to improve the performance of the subsequent classification task.

2.5.2 CLASSIFICATION

Classification is the process of assigning input data to predefined categories or categories. Once the features are extracted from the data, classification algorithms are applied to classify the data based on these features.

a. ALGORITHMS

Popular classification algorithms include Support Vector Machines (SVM), Neural Networks, Decision Trees, and K-Nearest Neighbors (KNN) [17].

b. SUPERVISED LEARNING

Most classification applications involve supervised learning, where the algorithm is trained on labeled data with pairs of input and output.

c. OBJECTIVE

The objective is to develop a model that can accurately predict the class labels of other unseen data based on the extracted features.

d. APPLICATIONS

Classification is widely used in various fields such as medical diagnostics, image detection, spam detection and sentiment analysis. By combining efficient feature extraction with complex classification algorithms, powerful models can be developed that can interpret and make predictions from complex data.

2.6 IMPLEMENTATION OF ALGORITHMS

Several other algorithms, such as the support vector machine, C-Means, Watershed, Grey Level Co-occurrence Matrix, and Neural Networks, are employed in various research publications. These algorithms are put into pratice.

2.6.1 GREY LEVEL CO-OCCURRENCE MATRIX (GLCM) IMPLEMENTATION:

The gray level co-occurrence matrix (GLCM) is a mathematical technique for investigating the spatial relationships between pixels in an image. Texture analysis is a common method for image processing. GLCM measures the frequency of occurrence of pairs of pixels in an image with a specific value and in a specific spatial relationship. Following are the functionalities of GLCM Algorithm.

1. **Matrix construction:** The GLCM for visualization is the frequency with which a pixel of intensity i is close to a pixel of intensity j in a given direction (e.g. horizontal, vertical, diagonal).

2. **Texture features:** Various texture features can be extracted from GLCM, such as contrast, correlation, energy, and uniformity, which describe the texture properties of the image.

2.6.2 C-MEANS ALGORITHM

C-means, commonly known as Fuzzy C-means (FCM), is a clustering algorithm that generates a data block into two or more clusters. This is useful for dealing with cluster uncertainty and overlapping clusters. Following are the functionalitoes of C-Means Algorithm.

1. **Fuzzy partitioning:** Each data point belongs to a cluster to the extent specified by the membership grade.

2. **Objective function:** The algorithm generates an objective function (generalized least squares function) that represents the minimum distance from any given data point to the cluster center, weighted by the membership rank of that data point.

3. **Iterative Process:** FCM repeats cluster centers and membership grades until convergence is reached, resulting in optimal group performance.

2.6.3 WATERSHED ALGORITH

The watershed algorithm is a powerful segmentation method for image processing. It is particularly effective in separating residues or overlapping objects. Following are thee functionalities of Watershed Algorithm.

1. **Gradient Map:** The algorithm starts with a gradient map of the image, where higher values represent edges.

2. **Flooding Process:** Treat the gradient map as a map. The algorithm simulates the flow from the lowest points (called minima) until different drainage points converge at flow channels, which represent the boundaries between blocks.

3. **Markers:** Markers or seeds are often used to guide the flooding process, improving separation accuracy by defining areas of interest in advance.

2.6.4 NEURAL NETWORK

A neural network (NN) is a computational model inspired by the neurons in the human brain. It is widely used in machine learning and artificial intelligence for tasks such as classification, regression, and pattern recognition. Following are the functionalities of Neural Networks.

1. **Layers and Neurons:** Neural networks have an input layer, one or more hidden layers, and an output layer. Neurons in each layer, and connected by weights.

2. **Activation Functions**: Neurons apply an activation function on weighted input levels to introduce nonlinearity.

3. **Training Process:** The network is trained with a large amount of data. In training, weights are adjusted based on the error between the predicted and actual results, usually using back-propagation and gradient descent methods.

2.6.5 SUPPORT VECTOR ALGORITHM

Support Vector Machine (SVM) is a supervised learning algorithm for classification and regression tasks. It is known for its excellent performance in high-dimensional environments and its ability to design complex systems. Following are the functionalities of SVMs.

1. **Hyperplane:** The goal of SVM is to find the best hyperplane that efficiently separates the data into different classes. In binary classification, this hyperplane maximizes the margin between the two classes.

2. **Support vectors:** The data points adjacent to the hyperplane, known as support vectors, are important for defining the position and orientation of the hyperplane.

3. **Kernel Trick:** SVM can handle nonlinear data efficiently by using a kernel function (e.g., linear, polynomial, radial basis function) that transforms the data to a higher level where a linear separation is possible.

4. **Margin Maximization:** The algorithm maximizes the margin (the distance between the hyperplane of each class and the nearest data points) to improve the generalizability of the classifier.

Each of these methods and algorithms plays an important role in different visualization and machine learning tasks, offering unique benefits and applications.

TABLE 2.1 COMPARISION OF ALGORITHMS

Attribute	**Support Vector Machine (SVM)**	**Neural Network (NN)**	**Watershed Algorithm**	**C-means Algorithm**	**Gray Level Co-occurrence Matrix (GLCM)**
Purpose	Classification and regression tasks	Classification, regression, and pattern recognition	Image segmentation and watershed transform	Clustering and pattern recognition	Texture feature extraction
Approach	Finds hyperplane that best separates data classes	Layers of neurons for deep learning and pattern extraction	Uses gradient of image intensity for segmentation	Clusters data based on similarity	Analyzes spatial relationships between pixels
Data Handling	Effective with high-dimensional data	Effective with large and complex datasets	Effective with grayscale or gradient images	Effective with feature vectors and spatial data	Effective with image texture and spatial data
Training Method	Supervised learning	Supervised learning (Backpropagation)	Unsupervised (image processing)	Unsupervised (clustering)	Statistical analysis of image matrices

Complexity	Moderate to high, depending on kernel choice	High complexity, especially with deep networks	Moderate, involves image processing techniques	Moderate, involves iterative optimization	Low to moderate, depends on matrix computation
Accuracy	High accuracy with appropriate kernel	Very high with sufficient training data	High accuracy in well-defined boundaries	High accuracy with appropriate parameters	Dependent on texture patterns and window size
Computational Cost	High, especially with non-linear kernels	High, especially with deep layers and large data	Moderate, involves image transformations	Moderate, depends on data size and iterations	Moderate, involves matrix calculations
Interpretability	Moderate, can be complex with non-linear kernels	Low, often seen as a black box	High, visually interpretable segmentation	Moderate, cluster centers and memberships	High, clear numerical texture features
Scalability	Scales well with kernel trick	Scales with additional layers and nodes	Scales with image size and resolution	Scales with data size	Scales with image resolution and number of textures
Applications	Image classification, face recognition	Image recognition, natural language processing	Medical image analysis, object segmentation	Image segmentation, anomaly detection	Texture classification, medical imaging

2.7 SIGNIFICANCE OF THE ALGORITHMS

Research papers discuss the importance of various algorithms, such as Pre-trained CNN – AlexNet, a potent deep learning model with impressive image recognition performance. It makes it possible to extract features from photos of skin conditions [7]. In order to handle high dimensional data, the Support Vector Machine Algorithm is useful algorithm. Algorithm for unsupervised learning and data point clustering is the K-Means Clustering Algorithm. It can serve as a grouping tool for comparable features or patterns in the image. For precise and automated diagnosis, Artificial Neural Network (ANNs) can be used to classify different disease types based on features taken from color images.

2.8 MACHINE LEARNING AND LIBRARIES

Machine learning plays a pivotal role in the project, particularly through the use of CNNs [1] and RCNNs [1]. These algorithms will be implemented using popular machine learning libraries in Python, such as TensorFlow [4] and PyTorch [5], to develop, train, and optimize the skin disease detection model [19]. Some researchers have use MATLAB for image processing [20].

2.9 LIST OF PREVIOUS SIMILAR SOFTWARE AND APPLICATIONS

Several previous software applications have attempted to address skin disease detection. Examples include DermNet NN [21] [22] [23], AI dermatologist skin cancer, and Skin Vision [24] [25] [26], but they may lack the sophistication and automation intended for this project. The various similar software for TELECLINIC (Skin) have various products. For example; Direct to consumer medicine, Teledermatology for the diagnosis and management of skin cancer, Mobile applications in dermatology and Artifical Intelligence in: Present, Past and Future, Fundamentals of Telemedicine and Telehealth.

This project falls into two categories: the first one is telemedicine, and the second one is skin disease detection. For example; this project is going to implement a web application to facilitate communication between the remote area residents and dermatologists. Once the system has diagnosed a disease, the dermatologist will then prescribe an online medication and treatement. Thus, applications are accepted for both groups.

Similar applications that leverage computer vision and machine learning for image analysis can be found in various domains, including medical image analysis, facial recognition, and object detection. However, this project is tailored specifically for skin disease detection. There are application based on MobileNet [18]. Some of them have used YOLO in real time for detecting skin cancer [28]. There are researcher who have used GLCM features for classifying skin pictures [29]. Several comparable telemedicine uses CNNs [1] and Region-Based CNNs (RCNNs) [1] have been investigated in a number of studies for their potential application in automated skin diagnosis. These investigations have yielded encouraging findings regarding the precise diagnosis and differentiation of skin conditions. Since the web application is launching for the users in remote areas, prior efforts in this field have concentrated on telemedicine applications that facilitate communication between patients and medical professionals. To make it easier for dermatologists to treat patients, skin detection is also carried out using image processing techniques. Once patients provide skin images, roughness and acne are obtained.

2.10 DEPLOYMENT STRATEGIES

Medical diagnostic errors in telemedicine are a challenge that is highlighted in another research paper. Its main objective is to design a multimedia system that uses the Grey Level Co-occurrence Matrix to automatically recognize skin textures and test it on three different dermatological conditions. The findings demonstrate that, given skin images, GLCM is capable of diagnosing three differnet dermatological conditions.

The strategies and advantages of offering effective answers to emerging telemedicine challenges through multimedia-based computer-aided and computer-supported cooperative diagnosis between medical professionals in disparate locations are highlighted as it discusses deployment strategies. The feasibility of telemedical wound care using a new generation of mobile phones with integrated cameras is another deployment strategy covered in the research paper.

2.11 MAIN FEATURES AND TECHNICAL INTERFACES

The study contrasted the assessments of leg ulcers made by in-person dermatologists with those made by remote dermatologists (the gold standard). Another research papers have different main features Dermal thermography, hyperspectral imaging, digital photographic imaging, and audio/video/online communication are the four modalities that are the focus of this review. The Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines were followed in the conduct of the systematic review. In addition to defining the population of interest, the intervention, and the results, clinical questions (PICOs) were developed and assessed for their clinical applicability.

The use of pretrained CNNs, such as Alex Net, which enables feature extraction from skin images with automatic learning of discriminative features, is a common feature of various research papers. Integration with Support Vector Machines (SVM) is another key feature that enables accurate skin disease classification by combining CNN [1] and SVM capabilities. The emphasis on image processing techniques, which is the method's focus on image processing techniques to identify and analyze skin disease images, is another key component of research papers. The telemedicine research papers examine various aspects. A feasibility study of telemedical wound care is presented in the paper using a new generation of cell phones with built-in cameras. The telemedicine research papers examine various aspects. Technical interfaces are examined in the research paper: The technical interface of a new generation of mobile phones is used in the research paper "Telemedical Wound Care Using a New Generation of Mobile Telephones: A Feasibility Study" to take pictures of leg ulcers. These pictures were sent right away by email to distant dermatologists for their consideration. The aim of the research was to assess the feasibility of using this new generation of mobile phones and direct email transfer for telemedical wound care.

The technical interface for this research paper is based on some other technical interfaces of research papers on telehealth and telemedicine applications for the diagnosis, monitoring, treatment, and prevention of diabetic foot disease. The review was carried out in accordance with the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines, and the articles were obtained exclusively from the MEDLINE database. The modalities that were taken into consideration were photographic imaging, dermal thermography, hyperspectral imaging, and audio/video/online communication.

The technical interfaces of the research paper on skin diseases include image processing, computer vision, color image analysis, and machine learning. The method that is recommended resizes a digital image of the affected skin area and applies a pretrained convolutional neural network to extract features. Subsequently, the features are classified using a multiclass SVM to identify the kind of skin disease. The only pricey pieces of equipment required for this quick and simple method are a computer and a camera. Several other research papers have also provided an explanation of the technical interface. Adaptive thresholding, edge detection, K-means clustering, and morphology-based image segmentation are some of the image processing techniques used to identify skin diseases from a given image set.

CHAPTER III

HARDWARE, SOFTWARE ANALYSIS AND REQUIREMENTS

3.1 INTRODUCTION

This is the app for those who use remote derma and wants to make communication between patients and dermatologists easier. Secondly, it falls under the detection category, wherein the system uses images to diagnose and identify skin diseases. Software requirements are therefore essential to the completion of any AI/ML project. Since both hardware and software requirements are essential. In order to program well, these two components must be added. Since a dataset is a prerequisite for the software, gathering a data set of skin images is a necessary software requirement for programming purposes and creating software like skin disease detection. The project will use these skin disease photos to train a convolution neural network model. Hardware for this project also includes a tablet computer that will be used as a dermatologist's pad. The dermatologists can prescribe any medication with ease using this online platform, and they can write any medication with a digital pen on a graphical tablet.

3.2 SOFTWARE REQUIREMENT ANALYSIS

The software requirement analysis for the project:

3.2.1 SKIN IMAGE

The first requirement of the project is to obtain images of the skin, which are the primary data entry of the system. Patients take these pictures with their smartphones or other digital cameras and upload them to a telemedicine platform. High-quality images are important because they contain the visual information needed for an accurate diagnosis and diagnosis by both the AI diagnostic system and the dermatologist**.**

3.2.2 PREPROCESSING

The preprocessing consists of a series of steps to prepare the skin images for further analysis. It includes functions like resizing, normalization, noise reduction and color adjustment to enhance the quality of the image. Preprocessing ensures that images are standardized and free of artifacts, which can improve the quality and accuracy of subsequent images and machine learning tasks.

3.2.3 SEGMENTATION

Segmentation is the process of dividing a preprocessed skin image into logical areas or sections, usually isolating the area of interest (such as a lesion) from the background This step is important to analytics are focused on relevant parts of the image, so AI can detect and accurately assess skin conditions Can be done.

3.2.4 FEATURE EXTRACTION

Feature extraction is a technique for identifying and quantifying specific quantities from segmented skin images. These factors may include the color, shape, texture, and size of the lesions. The excluded features act as input to the classification system, encompassing the critical information needed to distinguish between skin conditions.

3.2.5 CLASSIFICATION

Classification is the final step in which the extracted features are used to classify skin images into different skin states. Using machine learning models such as K-Nearest Neighbors (KNNs), Support Vector Machines (SVMs), or Convolutional Neural Networks (CNNs) [1], the system can pedis predict the presence and type of skin diseases (e.g. , ringworm, nail fungus, Tinea). This automated test helps dermatologists make informed decisions and provides patients with early detection.

3.3 HARDWARE

Numerous hardware platforms, including PCs and laptops, are compatible with the web application. Second, the hardware component is being used in the web application to make things easier for the dermatologists. The doctor pad is a graphical tablet that is part of this hardware. A dermatologist can use a digital pen to write on a graphical tablet to prescribe any medication. The patients can view the web application on their smartphones.

3.4 SOFTWARE

This telemedicine platform integrates many leading-edge technologies to provide comprehensive skin health care services. The front end is built with HTML, CSS, and JavaScript, ensuring responsiveness and ease of use. HTML provides the layout of web pages, CSS provides styling for consistent and visually appealing design, and JavaScript provides interactivity and dynamic functionality such as real-time video calling and responsive document presentation

Behind the scenes, PHP is used to handle server-side operations, including user authentication, form processing, and secure communication between the front end and the server. MySQL acts as a relational database system framework, storing and organizing critical data such as patient records, dermatologist profiles, medical images, etc., ensuring data integrity and flying query processing effective Furthermore, the platform has an AI diagnostic system powered by CNNs [1]. investigations, providing valuable diagnostic aids for dermatologists Together these technologies provide robust, safe and effective telemedicine solutions.

3.5 FRONTEND

Frontend of the web application is developed using HTML, CSS and JAVASCRIPT, inorder to create a user friendly and responsive interface. HTML (Hyper Text Markup Language) used for structuring and defining the layout and how the content is displayed on the website, it also ensures that all the content is organized correctly. CSS (Cascading Style Sheet) is used to style the entire web application to have a modern and minimalistic look, which allows for a aesthetically pleasing, uniformed and consistent look across the entire web application, on all types of devices. JavaScript is used to add interactive and dynamic functionality to the web application. It enables features like text-messaging, responsive form submission and video calling.

By using all of the above mentioned technologies, the web application ensures that the platform provides a easy to use, visually pleasing, and accessible platform for both the users; dermatologists and patients.

3.6 BACKEND

The backend of this web application is build using PHP (Hypertext Preprocessor). PHP is an open source programming language which is used world wide to implement backend. PHP is used to handle all server side operations like user authentication on login, it also process user submitted forms. PHP is used to interact with the database to insert, retrive, update and delete the data from the database. PHP is responsible to handle all communication between fronted and server. PHP manages appointment schedulling, uploading skin image and prescriptions and real time messaging.

By using PHP for backend, the project ensures that the web application is secure, robust and performant while handling dynamic content and all sorts of complex functionalities which are required for any reliable telemedicine web application.

3.7 MACHINE LEARNING

Machine learning plays a pivotal role in the project, particularly through the use of CNNs [1]. These algorithms will be implemented using popular machine learning libraries in Python, such as Tensor Flow [4] and PyTorch [5], to develop, train, and optimize the skin disease detection model.

The machine learning component of this telemedicine platform includes developing and integrating an AI diagnostic system using CNNs [1] This AI model is trained to recognize and diagnose three common skin diseases:Ringworm (Tinea Corporis), nail fungus and Tinea Pedis (Athlete's foot) uploaded by patients By analyzing images of the skin, the CNN [1] model can accurately detect these conditions and provide early diagnosis. This machine learning application aims to support dermatologists by providing rapid, reliable diagnostic assistance, and enabling remote dermatology consultations well and precisely done. The AI-powered approach helps bridge the gap between patients in remote areas and urban dermatologists, enabling timely and effective skin health care is easy to do.

3.8 DATABASE

This web application uses MYSQL as its primary database for storing user information. MYSQL is an open source Relational Database Management System (RDMS). It is known for its robustness and easy of use. MYSQL fulfills all the data storage needs. MYSQL handles the stroage and retrival of all the curcial data, which include patient records, dermatologists profiles, appointment schedules, and medical images like skin images and prescriptions. MYSQL ensures data integrity and security, supports efficient query processing and transaction management. Using MySQL, the platform can efficiently manage large amounts of structured data, providing seamless communication between the back-end and front-end components. This ensures that all user data is securely stored and easily accessible, contributing to the efficiency of the telemedicine services offered by the platform.

3.9 LIBRARIES

This project uses the following two libraries for creating CNN [1].

3.9.1 TENSORFLOW

TensorFlow [4] is an open source machine learning framework, which is developed by Google. This provides a comprehensive framework for technical education to build, provides forth and measure for construction, training and the doing of the two, which supports and supports the development of developed countries. Skilled training in darshas can provide the answer. TensorFlow [4] also includes visualization, debugging, and optimization tools, which help streamline the development process and improve model performance.

3.9.2 KERAS

Keras [30] is an open source deep learning library for building and training neural networks. Designed to be user-friendly, modular, and extensible, it allows developers to build complex neural network models with minimal code. Keras [30] provides a high-level API that streamlines the definition, training, and analysis of deep learning models, making them accessible to both beginners and experts Supports multiple backends, including TensorFlow [4], for consumption role for low-level calculations. Keras [30] includes many pre-built layers, optimizers, and loss functions, enabling rapid prototyping and testing in deep learning projects.

3.10 ALGORITHMS

In order to detect skin diseases, CNN [1] will be used. It is primarily used to classify images and extract or learn features from them. It is capable of handling very complex, big data sets. The algorithm will automatically learn and extract features from skin images when we apply it to the skin images. The collection of skin image data automatically determines the type of disease based on its features. When compared to other algorithms like SVM, KNN, etc., CNN [1] is the most potent algorithm.

3.10.1 CONVOLUTION NEURAL NETWORK (CNN)

A CNN [1] is a specific type of deep learning model which are specifically designed to process network data sets, such as fig. CNNs [1] have many layers, including convolutional layers that apply filters to input data to capture spatial hierarchy, pooling layers which help in reducing dimensionality, and fully connected layers for classification CNNs [1] are more effective for finding patterns and features in images various Becoming models are known for their ability to automatically learn feature representations from raw data with minimal preprocessing.

3.11 ACTIVITY DIAGRAM

The system's activity diagram given below in Figure 3.1. Here, the user logs in to the system initially, and it verifies their identity. The patient can then upload a picture of their skin at that point. The picture can be in either the PNG or JPEG format. A database contains the image. Following the patient's upload, a pre-processing apply that allowingthe user to rotate, resize, and crop the image that was inserted. The image is sent for feature extraction once the pre-processing step has been applied. Subsequently, the picture is given to the model to determine the kind of skin condition. The model forecasts whether the patient has fungal nails, foot fungus, or ringworm. A report detailing the skin condition is produced after the image has been evaluated and the disease has been determined. Next, the physician reviews the report's findings and decides whether to accept or reject them. The patient is then given treatment and medication by the physician.

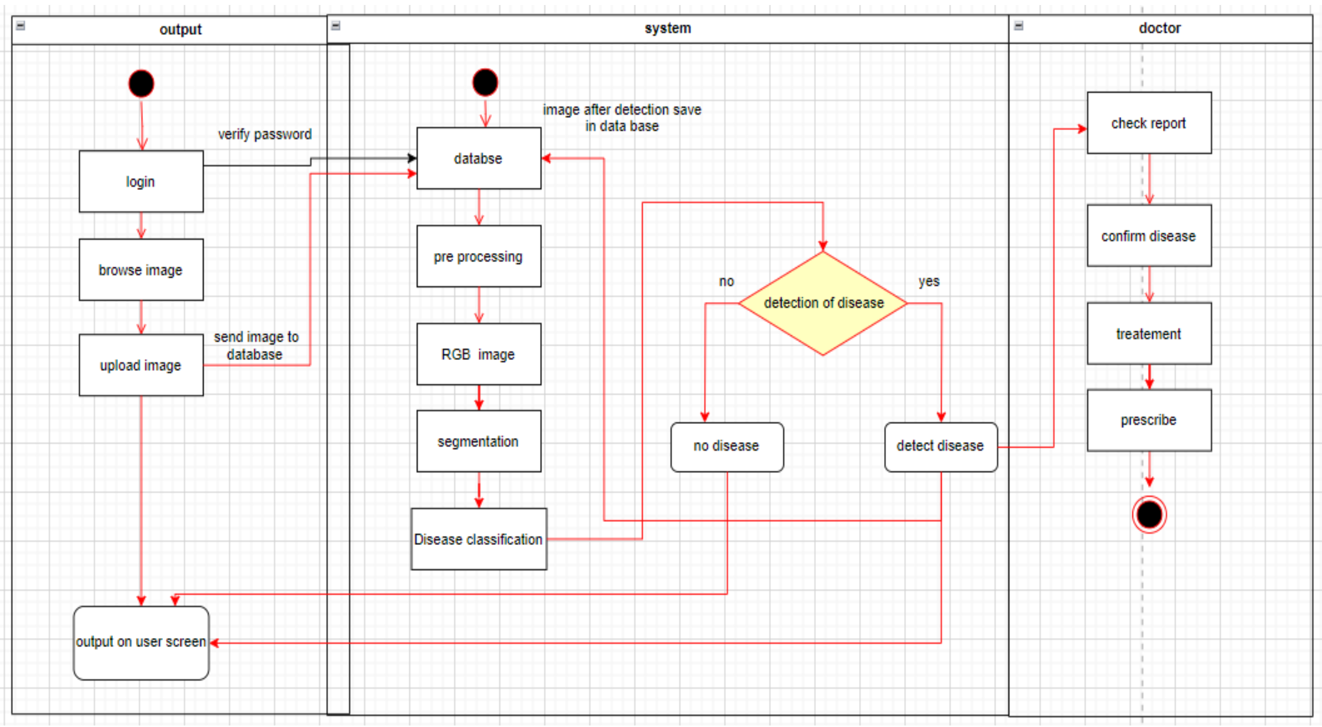


Figure 3. 1 ACTIVITY DIAGRAM

3.12 ACTOR USE CASE DIAGRAM

Figure 3.2 shows all the different use cases of the system related to registration. Here both actors dermatologist and patient can sign up and sign in to the system. They both can edit, update, view their profile and edit their profile and sign out.

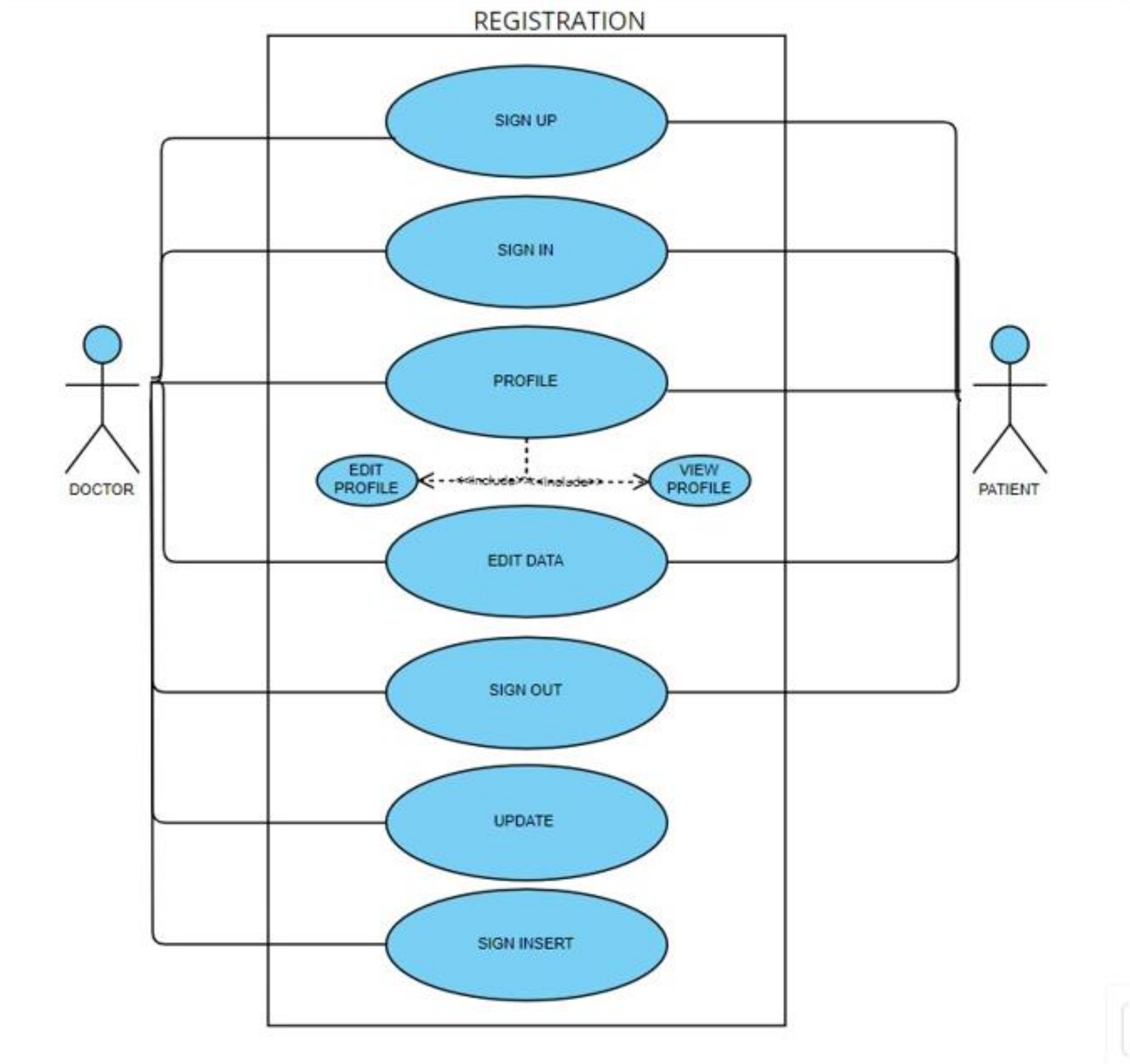


Figure 3. 2 ACTOR USE CASE DIAGRAM (REGISTRATION)

Figure 3.3 describes and shows all the different use cases of the system related to image processing. Where a patient can upload an image of his/her skin. The image can be of any one of the two format JPEG or PNG. Ones the user has uploaded his image then a pre-processing tab opens which allows the user to crop, resize and rotate the image that the user has inserted. After the user has performed pre-processing on the image, the image is then sent for feature extraction. After that the image is send to the model for classification of the type of skin disease. Model predicts whether the patient has ringworm, foot fungus or nail fungus. Ones the image is assessed and the disease is been identified, a report is generated describing the skin disease. Then the dermatologist check the result of the report and confirm or rejected the result of the report. The dermatologist then prescribes treatment and medicine to the patient.

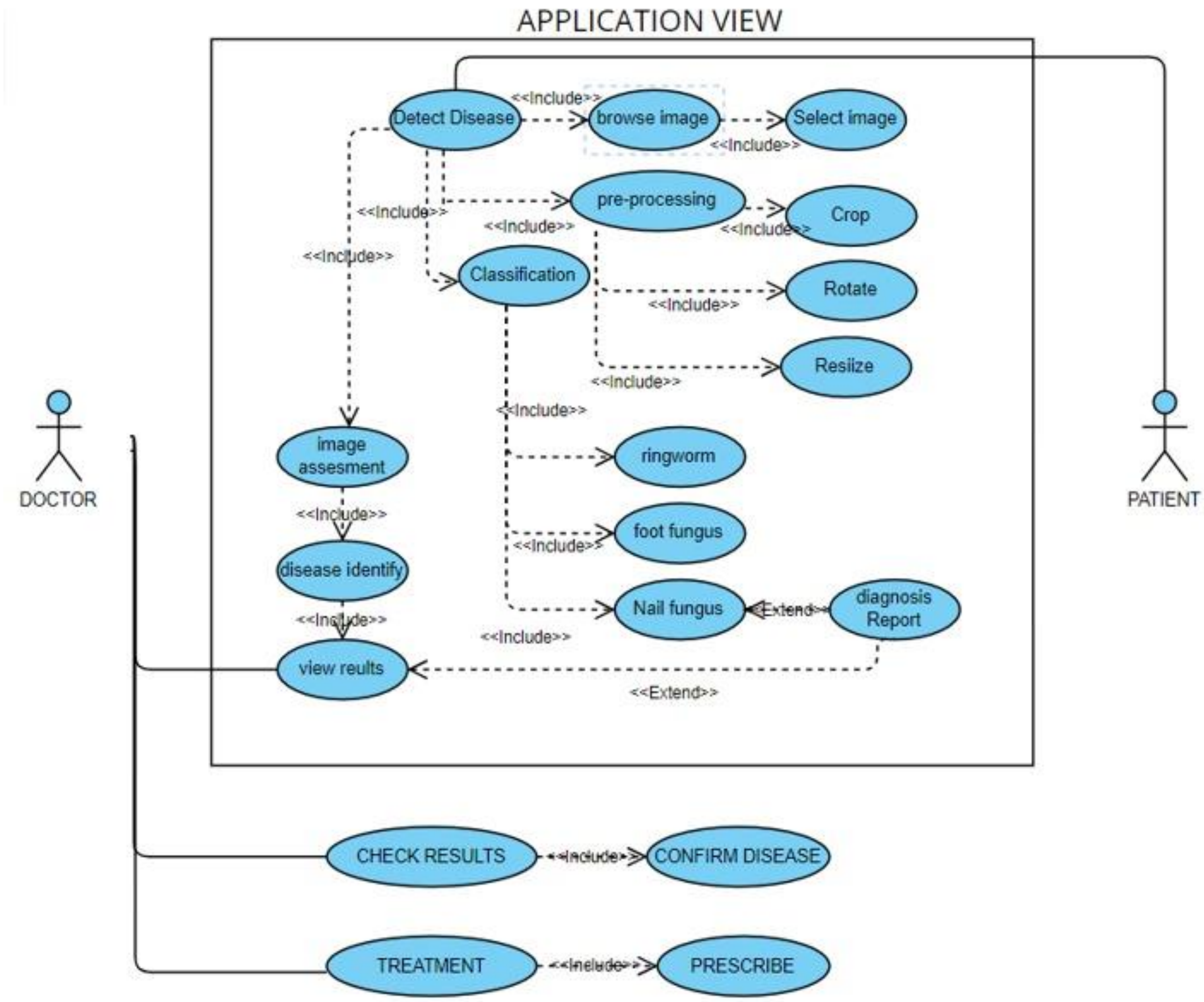


Figure 3. 3 ACTOR USE CASE DIAGRAM (APPLICATION VIEW)

Figure 3.4 shows all the different use cases of the system related to appointment. Here patient is able to check the availability of dermatologist, can fix an appointment with the dermatologist and can modify the appointment like time, date, etc.

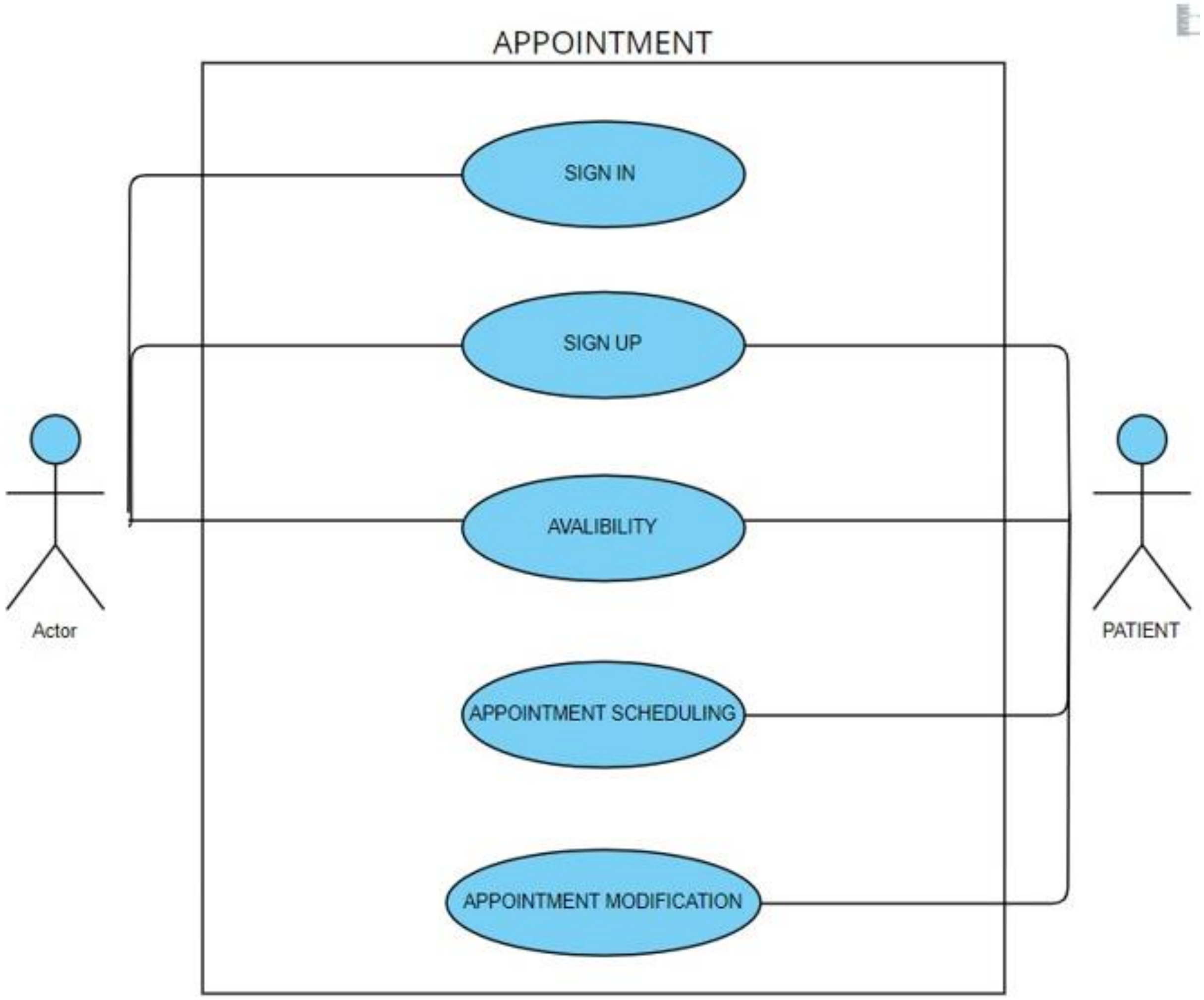


Figure 3. 4 ACTOR USE CASE DIAGRAM (APPOINTMENT)

3.13 SYSTEM DIAGRAM

Figure 3.5 displays the system diagram. This diagram illustrates the entire system, including how all of the skin image data sets were sent for training, how the convolution neural network algorithm was applied to the skin disease images, and how image processing operations were applied to the skin images. The data set is first preprocessed by the system. This system can resize and crop an image by preprocessing it and then applying a segmentation part. This system is able to segment the primary portion of that illness through segmentation. Following segmentation, the system categorizes the illness according to its type. This system has been trained to identify the illness. The user then inputs an image of their skin condition, which is sent to a trained model, enabling the model to identify the type of disease. After the model has classified the disease, the image is sent to a dermatologist via a web application. Once the dermatologist views the user's image and its details, the dermatologist can prescribe medication using a graphical tablet connected to the doctor pad. Once the medication is prescribed, the patient receives treatment.

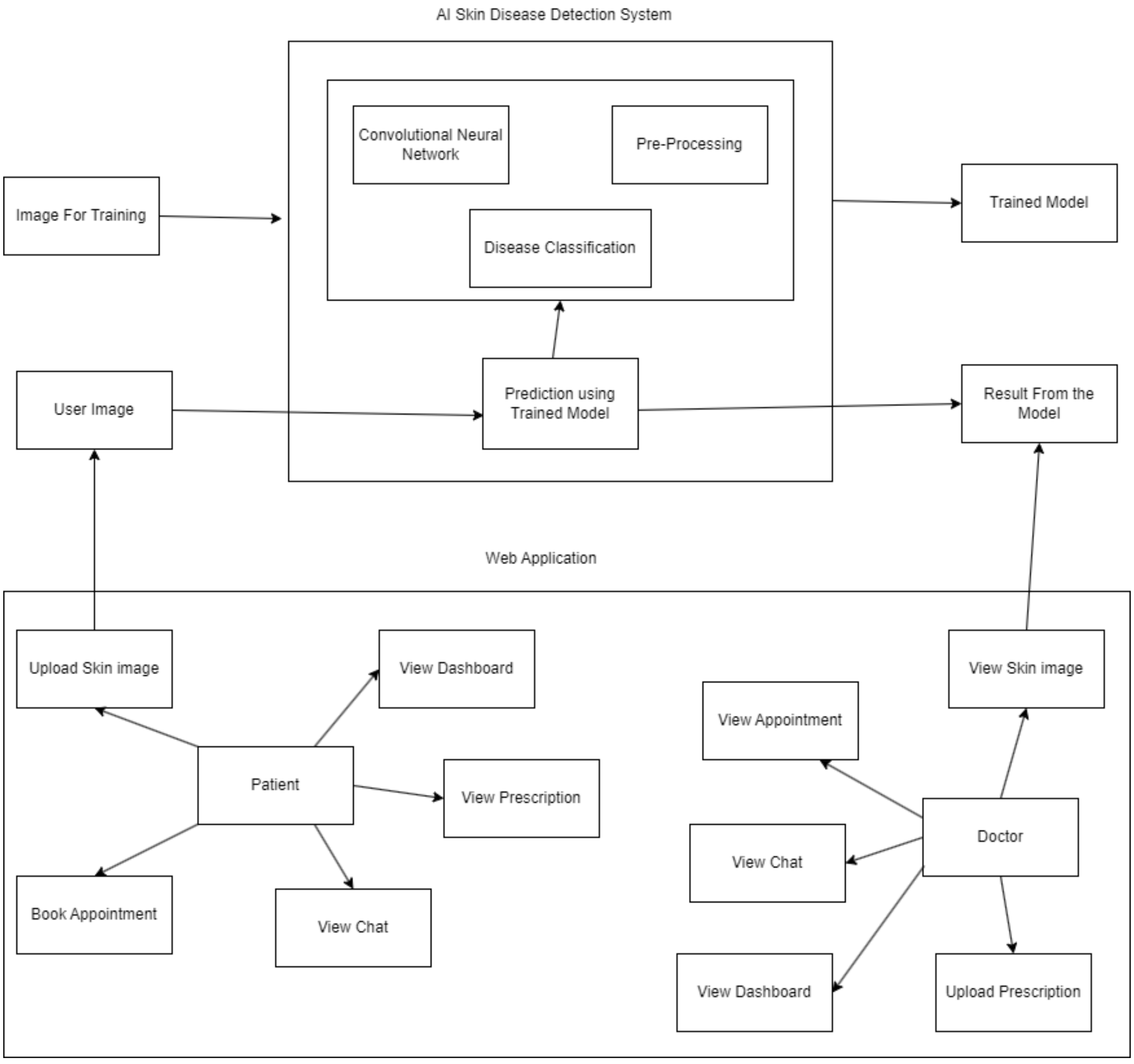


Figure 3. 5 SYSTEM DIAGRAM

CHAPTER IV

SOFTWARE DESIGN AND MODELING

4.1 INTRODUCTION

This project is TELECLINIC (Skin) and this is the web application for remote areas peoples and the system can detect three skin diseases mainly that are Ring Worm, Nail Fungal and Foot fungal . This chapter, discusses about various diagrams for this project.

4.1.1 USE CASE DIAGRAM

In the Unified Modeling Language (UML), a use case diagram is a visual representation that shows how users (actors) and a system interact in terms of particular use cases, which are discrete functionality or activities. It shows how users interact with the system to accomplish particular goals and portrays the behavior of the system from the viewpoint of the user.

4.1.2 ACTIVITY DIAGRAM

In UML (Unified Modeling Language) convention, their is a diagram called activity diagrams which are usually used to show how activities or actions are performed through the workflow of a system. It shows how the decision-making process, proceeds randomly, and how control transfers from one task to another.Activity diagrams help illustrate how different functions or processes interact and are useful for modeling dynamic system components.

4.1.3 CLASS DIAGRAM

A class diagram is a type of UML (Unified Modeling Language) diagram that shows how classes, their types, relationships, and methods are used to organize a system It provides a static representation of the system, focusing on relationships between classes.

4.1.4 ERD DIAGRAM

A visual data model that depicts entities (objects), their relationships and attributes among them is called an entity-relationship diagram, or ERD. ERDs are commonly used in database design to describe the relationships and and organization of various items inside a database.

4.1.5 SEQUENCE DIAGRAM

A sequence diagram is a type of UML diagram in which interactions are shown and the order in which message between objects or components are passed in a system across time. It emphasizes the order of events and the flow of control, representing the dynamic behavior of a system.

4.1.6 SYSTEM DIAGRAM

System diagram are like the models that visually express forces acts on the components of process and some interactions between that force.

4.2 CLASS DIAGRAM

Figure 4.1 shows all the class in this system. There are six classes Dermatologist, Patient, Appointment, Prescription, SkinPhoto, and Login. The Dermatologist class has attributes of id, name, gender, contact_number, email and specialization. The Dermatologist class contain mthods like signup(), appointment(), diagnosis(), record(), chat(), prescribe_medicine(). The Patient class has attributes of id, name, gender and contact_number. The Patient class has methods like appointment(), get_diagnosis(), get_record(), get_chat(). SkinPhoto class has attributes likes id and image_url. The SkinPhoto class has methods like get_diagnosis(). Prescription class has attributes like prescription_id, dermatologist_id, patient_id, medication, test and the class contains methods like get_prescription(). The Appointment class has attributes like id, patient_id, dermatologist_id, datetime and skin_photo. The Appointment class has methods like schedule_appointment(), cancel_appointment() and view_appointment(). The Login class has attributes like name, contact_number, email and address. The Login class has methods like get_verify(), username_password(), get_number().

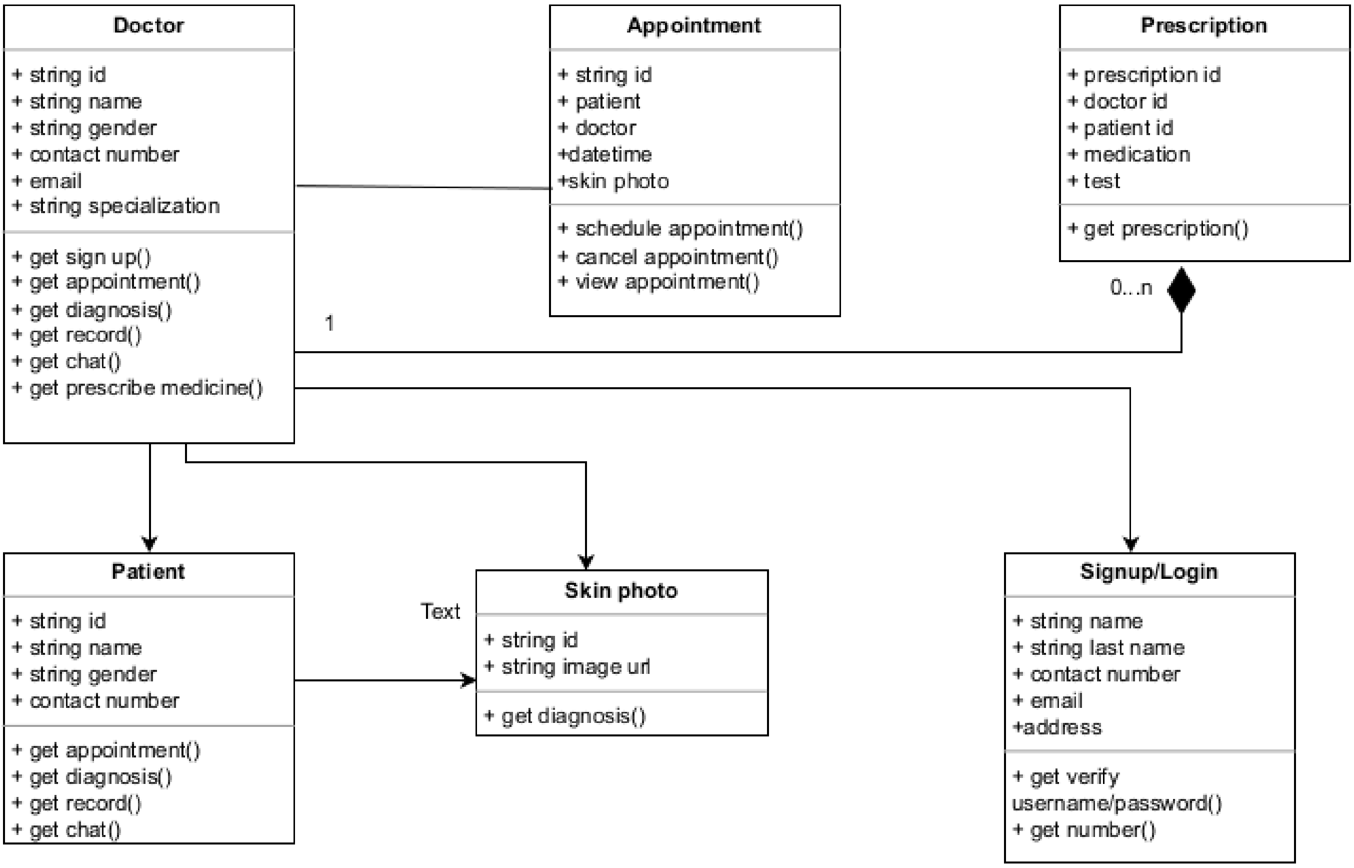


Figure 4. 1 CLASS DIAGRAM

4.3 ERD DIAGRAM

Figure 4.2 shows the database design of the system. This system's database consists of six tables. These tables include Role, Admin, Dermatologist, Patient, Appointment, Payment. The Role table has two attributes role_id and privileges. The Admin table has three attributes admin_id, admin_name and role_id. Dermatologist table has nine attributes, dermatologist_id, dermatologist_age, dermatologist_name, dermatologist_gender, dermatologist_qualification, dermatologist_specialization, dermatologist_contact_number, dermatologist_account_number and role_id. Patient table has eight attributes, patient_id, patient_age, patient_name, patient_contact_number, patient_gender, patient_medical_history, patient_account_number and role_id. Payment table has three attributes, payment_id, account_number and payment_status. Appointment table has six attributes, appointment_id, patient_id, dermatologist_id, date, appointment_status and payment_id.

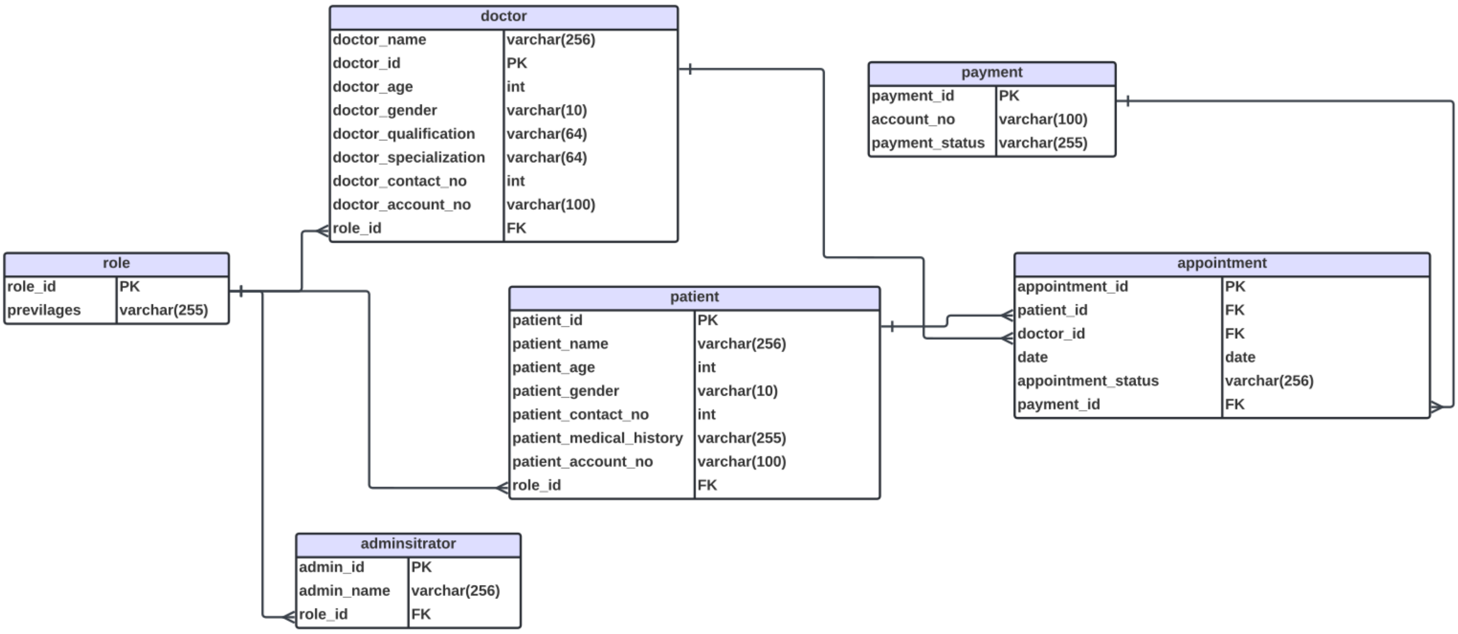


Figure 4. 2 ERD DIAGRAM

4.4 SEQUENCE DIAGRAM

Figure 4.3 and figure 4.4 show sequence diagram of the system with respect to actors; dermatologist and patient. In these sequence diagram the interaction between both actors is shown with respect to the system. The sequence diagram shows how both of them login. The patient can upload skin image and the dermatologist can view it. The patient can make a appointment and the dermatologist can take that appointment. The dermatologist can prescribe medicine and test to the patient and the patient can take a look at the prescription.

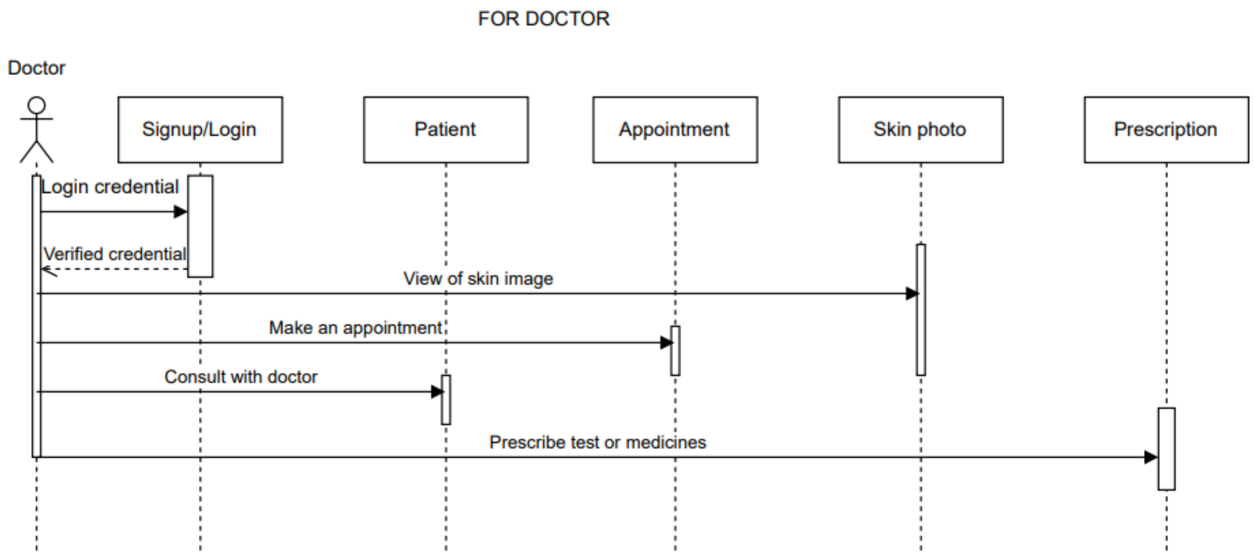


Figure 4. 3 SEQUENCE DIAGRAM (FOR DERMATOLOGIST)

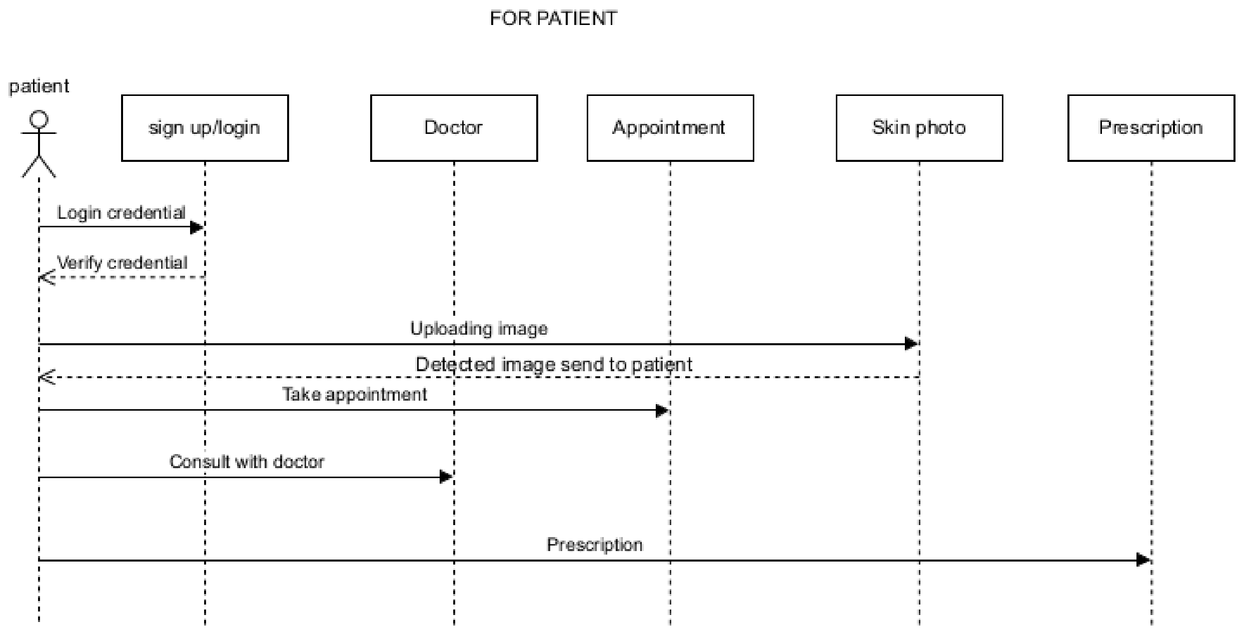


Figure 4. 4 SEQUENCE DIAGRAM (FOR PATIENT)

CHAPTER V

ALGORITHM ANALYSIS AND COMPLEXITY

5.1 INTRODUCTION

This project, TELECLINIC (Skin), is a web application designed to facilitate communication between physicians and patients and cater to individuals living in remote areas. This system is capable of detecting the first three diseases: foot fungal, ring worm, and skin nail fungal.therefore, for the training of the dataset images of skin disease, this project will be using “Convolution Neural Network (CNN) [1] Algorithm”. Compared to other algorithms, the CNN [1] algorithm operates extremely quickly.numerous algorithms, such as the Random Forest, K-Nearest Neighbor, Support Vector Machine (SVM), [17] and others, were used in earlier research projects. While some research papers used svm alone. The pretrained CNN [1] Algorithm is used in conjunction with the AlexNet model, much like in earlier research papers. Extracting features from resized color images of skin diseases is its main goal. CNN's [1] unique features make it suitable for skin disease detection task and feature extraction, so it is an additional algorithm for classifying extracted features. Several different algorithms are used in research papers, but one of the most effective algorithms for managing large data sets is CNN [1].

5.2 PURPOSE

CNN [1] can process complex images from a data set quickly. When the data set of pictures of skin diseases will be subjected to the CNN [1] algorithm. After the algorithm is applied and the model is trained, it will be able to identify the type of skin disease by identifying its features, which it will extract from the images of the disease. High dimensional image data processing and analysis benefit from its use. Its objective is to automatically extract features from the value of pixels. Features based on color variation, texture, and shape extraction will be used in the skin disease detection system. These features are crucial for precise disease classification. Given that the high-resolution images in the data set of skin diseases are handled by the CNN [1] model, in an efficient manner.

5.3 SIGNIFICANES

CNN [1] has significant role in various fields especially in image analysis and video analysis. CNN [1] achieved success in image recognize task as it can learn features automatically from the data set of skin disease images and gives accurate classification of textures, patterns within the image. It has the significant role mainly in detection task and diagnosis of disease in medical images of data set. It is highly effective for recognizing the complex patterns within the image. CNN [1] is capable of learning hierarchal features or representations from data set as a powerful tool for image analysis and pattern recognition. It works on large datasets and complex too. CNN [1] also widely implemented on facial recognition tasks for automatically recognize facial features. However, for pictures of skin conditions CNN [1] automatically identify and extract characteristics from skin images in order to accurately classify the type of disease that is present.

5.4 VISUAL GEOMETRY GROUP 16 (VGG16)

This project uses Visual Geometry Group 16 (VGG16) [31]. VGG16 [31] is an CNN [1] architecture. It has 16 layers, including 13 convolutional layers and 3 fully connected layers, using small 3×3 convolution filters and 2 × 2 max-pooling level. VGG16 [31] is known for its flexibility and performance in image classification tasks when trained on ImageNet dataset. It is mostly used for transfer learning, where pre-trained models are optimized for a new task. Despite its computing and memory capacity, VGG16 [31] is a popular choice for various applications such as feature extraction, object recognition, image segmentation and style transfer.

5.5 CHARACTERISTICS OF VGG16

1. All convolutional layers use a small receptive field of 3×3 with stride of 1 and padding to preserve spatial resolution.

2. With a stride of 2, there are also 2×2 maximum pooling levels, which are used to reduce the spatial dimensions of the feature maps.

3. In addition to the pooling and convolutional layers, there are three fully overlapping layers: the first two have 4096 neurons each, and the last one has 1000 neurons which is corresponding to the number of classes that are found in the ImageNet dataset.

4. After each fully connected and convolutional layer, a Rectified Linear Unit (ReLU) is applied as an activation function.

5. The last layer is the softmax level used for classification.

5.6 ADVANTAGES OF VGG16

1. **Simplicity:** Uniform architecture (using only 3×3 convolution) makes it easy to use and understand.

2. **Depth:** 16 layers allows the neural network to learn complex and abstract features from an image.

3. **Transferability:** Pre-trained VGG16 models are ubiquitous and a good starting point for a variety of image-related tasks.

5.7 LIMITATIONS OF VGG16

1. **Computational Expensive:** Due to its depth and number of parameters, VGG16 requires significant computational resources for training and inference.

2. **Memory Intensive:** The network has a large memory footprint, which can be limiting when working with hardware with limited memory capacity.

5.8 APPLICATIONS OF VGG16 WITH CNNS

Following are the applications of VGG16 [31] used along with CNNs [1].

5.8.1 IMAGE CLASSIFICATION

VGG16 [31] is mainly used for image classification tasks. It was initially trained on an ImageNet dataset with over a million images in 1000 categories. The network is able to classify images in these categories with high accuracy.

5.8.2 FEATURE EXTRACTION

The VGG16 [31] can be used as a feature extractor. Convolutional layers can provide rich feature representations of the input image, which can be used for various downstream tasks such as object image segmentation, detection, and other methods for image analysis.

5.8.3 TRANSFER LEARNING

The transfer learning uses a new but previously trained VGG16 [31] model for the same task. Instead of training a CNN [1], which requires a lot of data and significant computational resources, the pre-trained VGG16 [31] model can be optimized for new tasks General steps include:

1. Removing the last fully connected layer.

2. Adding a new fully connected layer which is corresponding to the number of classes in the new dataset.

3. Freezing the weights of previous layers and training only the new layer or fine-tuning all layers with a lower learning rate.

5.8.4 OBJECT DETECTION AND SEGMENTATION

The modified version of VGG16 [31] can be used in complex models (such as R-CNN, Fast R-CNN, and Faster R-CNN) [1] and image segmentation (such as Fully Convolutional Networks).

5.8.5 IMAGE STYLE TRANSFER AND SUPER-RESOLUTION

VGG16's [31] convolutional layers capture various levels of features, from edges to textures to objects, making it useful in applications like neural style transfer, where the style of one image is copied to the content of another image, and image super-resolution, where the resolution of images is enhanced.

5.9 ALGORITHM

The following algorthim is taken from the original paper written on VGG16 [31].

Define the VGG16 network

```
function VGG16(input_image):
    # Input layer
    input = Input(input_image)
    # Block 1
    conv1_1 = Conv2D(input, filters=64, kernel_size=(3, 3), padding='same', activation='relu')
    conv1_2 = Conv2D(conv1_1, filters=64, kernel_size=(3, 3), padding='same', activation='relu')
    pool1 = MaxPooling2D(conv1_2, pool_size=(2, 2), strides=(2, 2))
    # Block 2
    conv2_1 = Conv2D(pool1, filters=128, kernel_size=(3, 3), padding='same', activation='relu')
    conv2_2 = Conv2D(conv2_1, filters=128, kernel_size=(3, 3), padding='same', activation='relu')
    pool2 = MaxPooling2D(conv2_2, pool_size=(2, 2), strides=(2, 2))
    # Block 3
    conv3_1 = Conv2D(pool2, filters=256, kernel_size=(3, 3), padding='same', activation='relu')
    conv3_2 = Conv2D(conv3_1, filters=256, kernel_size=(3, 3), padding='same', activation='relu')
    conv3_3 = Conv2D(conv3_2, filters=256, kernel_size=(3, 3), padding='same', activation='relu')
    pool3 = MaxPooling2D(conv3_3, pool_size=(2, 2), strides=(2, 2))
    # Block 4
    conv4_1 = Conv2D(pool3, filters=512, kernel_size=(3, 3), padding='same', activation='relu')
    conv4_2 = Conv2D(conv4_1, filters=512, kernel_size=(3, 3), padding='same', activation='relu')
    conv4_3 = Conv2D(conv4_2, filters=512, kernel_size=(3, 3), padding='same', activation='relu')
    pool4 = MaxPooling2D(conv4_3, pool_size=(2, 2), strides=(2, 2))
    # Block 5
    conv5_1 = Conv2D(pool4, filters=512, kernel_size=(3, 3), padding='same', activation='relu')
    conv5_2 = Conv2D(conv5_1, filters=512, kernel_size=(3, 3), padding='same', activation='relu')
    conv5_3 = Conv2D(conv5_2, filters=512, kernel_size=(3, 3), padding='same', activation='relu')
    pool5 = MaxPooling2D(conv5_3, pool_size=(2, 2), strides=(2, 2))
    # Fully Connected Layers
```

```
    flatten = Flatten(pool5)
    fc1 = Dense(flatten, units=4096, activation='relu')
    fc2 = Dense(fc1, units=4096, activation='relu')
    fc3 = Dense(fc2, units=1000, activation='softmax')
    # Output layer
    output = fc3
    return output
# Example usage
input_image = LoadImage('path/to/image.jpg')
model = VGG16(input_image)
predictions = model.predict(input_image)
```

5.9.1 EXPLANATION

1. **Input Layer:** Takes the input image.

2. **Convolutional Blocks:** There are 5 blocks, each with 2 or 3 convolutional layers along with ReLU activation, which is usally followed by a max-pooling layer.

3. **Fully Connected Layers:** Three fully connected layers, along with the first two layers using ReLU activations and the last one using softmax for classification.

4. **Output Layer:** Provides the classification results.

This pseudocode provides a high-level overview of the VGG16 architecture, focusing on the sequence of operations and layer configurations.

5.10 COMPLEXITY OF THE ALGORITHM VGG16

The complexity of the VGG16 [31] network can be analyzed in terms of computational complexity (temporal complexity) and spatial complexity (memory complexity) These complexities mainly depend on the operations performed at the convolutional levels, which are very computationally intensive.

5.10.1 COMPUTATIONAL COMPLEXITIES

Each convolutional layer involves multiplying each filter by a local region of the input feature map and summing the results. The computational complexity for a single convolutional layer can be expressed as:

$O((K^2 \cdot C_{in} \cdot H_{out} \cdot W_{out}) \cdot N)$

Where:

- K is the size of the filter (e.g., 3 for a 3×3 filter).
- C_{in} is the number of input channels.
- H_{out} and W_{out} are the height and width of the output feature map.
- N is the number of filters.

In the VGG16 design, the complexity varies for each layer due to different input sizes and number of filters. All computational complexity can be calculated by summing the complexity of all variables.

5.10.2 WORST CASE COMPLEXITY

The worst-case computational complexity occurs when the input image is large, requiring all convolutional layers to process the maximum number of operations.

For a typical input size of 224×224×3 and the VGG16 configuration, the approximate number of operations is:

$$\text{Total Operations} \approx 15.5 \text{ billion (G) floating point operations (FLOPs)}$$

5.10.3 BEST CASE COMPLEXITY

Theoretically, the ideal scenario would include the smallest possible inputs that still make sense for the architecture. However, the practical lower limit is constrained by the fixed design of VGG16, which requires a minimum input size. The optimal complexity is still appropriate due to the fixed depth and number of filters in the grid.

5.10.4 MEMORY COMPLEXITY

Memory Complexity includes the amount of storage required for:

a. MODEL PARAMETER

Model Parameter includes weights and biases for each layer. The total number of parameters in VGG16 is approximately 138 million. This is calculated by summing the parameters of all convolutional and fully connected layers.

b. INTERMEDIATE ACTIVATION

Intermediate Activation includes feature maps which are generated after each layer during forward propagation. Memory requirements for activations may be important, especially during training when activations are also stored for external propagation. Memory requirements for activations depend on the input size and the number of layers.

In summary, the complexity of VGG16 [31] is strongly influenced by its robust design and a number of its parameters, making it computation-memory intensive for both worst-case and best-case scenarios.

CHAPTER VI

IMPLEMENTATION

6.1 INTRODUCTION

This project consists of a web application and Skin Disease Detection System. Using the web application patients and dermatologists can create their accounts. When a patient creates an account, he or she get access to a dashboard where they can browse all dermatologist and can book an appointemnt with them. After booking an appointment, the patient can upload his or her skin image. The Skin Disease Detection System takes that image and identifies the disease. Once the disease is identified. It is passed to the dermatologist to prescribe treatment and test. Patients can chat with the dermatologist via video call and text messages. When a dermatologist creates an account, he or she get access to a dashboard where they can see all their present, past and future appointments. The dermatologist can also chat with their patients. Dermatologist can view patient skin images and prescribe test and treatment using the build in doctor pad.

6.2 CODE FOR TRAINING DETECTION MODEL

```
# Load and preprocess training and testing data

X_train, Y_train = load_and_preprocess_data(folder_path_training, labels, image_size)

X_test, Y_test = load_and_preprocess_data(folder_path_testing, labels, image_size)



# Shuffle the data

X_train, Y_train = shuffle(X_train, Y_train, random_state=101)



# Perform one-hot encoding

Y_train = tf.keras.utils.to_categorical(Y_train, num_classes=len(labels))

Y_test = tf.keras.utils.to_categorical(Y_test, num_classes=len(labels))
```

```
# Data augmentation using ImageDataGenerator
datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)


# Load VGG16 model with pre-trained weights
base_model = VGG16(weights='imagenet', include_top=False, input_shape = (image_size, image_size, 3))


# Freeze the layers
for layer in base_model.layers:
    layer.trainable = False


# Add your custom layers
x = base_model.output
```

```
x = Flatten()(x)

x = Dense(512, activation='relu')(x)

x = Dropout(0.3)(x)

predictions = Dense(len(labels), activation='softmax')(x)


# Create a new model

model = Model(inputs=base_model.input, outputs=predictions)


# Compile the model

model.compile(optimizer=Adam(lr=0.0001), loss='categorical_crossentropy', metrics=['accuracy'])


# Train the model

history = model.fit(datagen.flow(X_train, Y_train, batch_size=32), epochs=10,
validation_data=(X_test, Y_test))


# Save the model

model.save('C:\\Users\\AB\\Desktop\\Detection model\\skin_disease_detection_model.hdf5')

model.save('C:\\Users\\AB\\Desktop\\Detection model\\skin_disease_detection_model.h5')
```

6.2.1 DESCRIPTION

The code first loads and preprocess the training ang testing dataset from the specified folder path. Images are resized to a constant size (‘image_size’) and labeled accordingly. The training data is shuffled to ensure that the model trains more effectively without any order bias. Class labels are converted to one-hot encoded vectors using the TensorFlow’s utility function. This is important for hierarchical classification tasks. An ImageDataGenerator is used to enhance the training images.Augmentation techniques include random rotations, shifts, shears, zooms, and horizontal flips. This helps to improve the generalization of the model by producing versions of the training images. The VGG16 [31] model is weighted using pre-trained weights from the ImageNet dataset, except for the top (fully connected) layer. The pre-trained VGG16 [31] layers have been frozen to maintain their known properties during the training process. Custom layers have been added on top of the VGG16 [31] base model. These include a Flatten layer, a Dense layer with ReLU activation, a Dropout layer to prevent overfitting, and a final Dense layer with softmax activation for classification into the number of classes defined by the labels. The model is compiled using the ADAM optimizer with a learning rate of 0.0001, and the loss function used is categorical crossentropy. Metric for analysis and accuracy. The model is trained using the augmented training data for 10 epochs. The validation data is used to monitor the model's performance on unseen data. The trained model is saved to the specified directory in both HDF5 and H5 formats.

6.3 CODE FOR SKIN DISEASE CLASSIFICATION MODEL

```
# Load and preprocess training and testing data

X_train, Y_train = load_and_preprocess_data(folder_path_training, labels, image_size)

X_test, Y_test = load_and_preprocess_data(folder_path_testing, labels, image_size)



# Shuffle the data

X_train, Y_train = shuffle(X_train, Y_train, random_state=101)

# Perform one-hot encoding

Y_train = tf.keras.utils.to_categorical(Y_train, num_classes=len(labels))

Y_test = tf.keras.utils.to_categorical(Y_test, num_classes=len(labels))



# Data augmentation using ImageDataGenerator

datagen = ImageDataGenerator(
```

```
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

# Load VGG16 model with pre-trained weights
base_model = VGG16(weights='imagenet', include_top=False, input_shape = (image_size, image_size, 3))

# Freeze the layers
for layer in base_model.layers:
    layer.trainable = False

# Add your custom layers
x = base_model.output
x = Flatten()(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.3)(x)
```

```
predictions = Dense(len(labels), activation='softmax')(x)


# Create a new model

model = Model(inputs=base_model.input, outputs=predictions)


# Compile the model

model.compile(optimizer=Adam(lr=0.0001),loss='categorical_crossentropy', metrics=['accuracy'])


# Train the model

history = model.fit(datagen.flow(X_train, Y_train, batch_size=32), epochs=10, validation_data=(X_test, Y_test))


# Save the model

model.save('C:\\Users\\AB\\Desktop\\Detection model\\skin_disease_classification_model.hdf5')

model.save('C:\\Users\\AB\\Desktop\\Detection model\\skin_disease_classification_model.h5')
```

6.3.1 DESCRIPTION

The code first loads and preprocess the training ang testing dataset from the specified folder path. Images are resized to a constant size (‘image_size’) and labeled accordingly. The training data is shuffled to ensure that the model trains more effectively without any order bias. Class labels are converted to one-hot encoded vectors using the TensorFlow’s utility function. This is important for hierarchical classification tasks. An ImageDataGenerator is used to enhance the training images.Augmentation techniques include random rotations, shifts, shears, zooms, and horizontal flips. This helps to improve the generalization of the model by producing versions of the training images. The VGG16 [31] model is weighted using pre-trained weights from the ImageNet dataset, except for the top (fully connected) layer. The pre-trained VGG16 [31] layers have been frozen to maintain their known properties during the training process. Custom layers have been added on top of the VGG16 [31] base model. These include a Flatten layer, a Dense layer with ReLU activation, a Dropout layer to prevent overfitting, and a final Dense layer with softmax activation for classification into the number of classes defined by the labels. The model is compiled using the ADAM optimizer with a learning rate of 0.0001, and the loss function used is categorical crossentropy. Metric for analysis and accuracy. The model is trained using the augmented training data for 10 epochs. The validation data is used to monitor the model's performance on unseen data. The trained model is saved to the specified directory in both HDF5 and H5 formats.

6.4 CODE FOR SKIN DISEASE DETECTION SYSTEM

```
class SkinImage(Resource):
    def post(self):
        # Load and preprocess the image for prediction
        file = request.files['image'].read()
        npimg = np.fromstring(file, np.uint8)
        img = cv2.imdecode(npimg,cv2.IMREAD_COLOR)
        img = cv2.resize(img, (150, 150))  # Resize the image to match the input size of the models
        img_array = np.array(img)  # Convert image to numpy array
        img_array = img_array.reshape(1, 150, 150, 3)  # Reshape array to match expected input shape
        # Perform prediction using skin disease detection model
        skin_disease_prediction = detection_model.predict(img_array)
        skin_disease_label = "Skin Disease" if np.argmax(skin_disease_prediction) == 1 else "No Disease"
        # Initialize the classification result
        combined_prediction = ""
```

```
    # Check if skin disease detected
    if skin_disease_label == "Skin Disease":
        msg = "Skin Disease Detected"
        # Check if the detected disease can be classified
        classification_prediction = classification_model.predict(img_array)
        classification_label = np.argmax(classification_prediction)


        # Check if the classification label is valid
        if classification_label == 0:
            msg += "\nClassified as: Not Classified\nConsult with a dermatologist."
        elif classification_label in [1, 2, 3]:
            class_labels = ["Onychomycosis", "Tinea Corporis", "Tinea Pedis"]
            msg += f"\nClassified as: {class_labels[classification_label - 1]}. \nDon't rely solely on this result. Please consult a dermatologist if you're feeling unwell."
        else:
            msg += "\nNot trained. Consult with a dermatologist."
    else:
        msg = "No Skin Disease Detected"
    return jsonify({"output": msg})
```

6.4.1 DESCRIPTION

The SkinImage class inherits from Resource, making it a RESTful resource managed by Flask-RESTful. The post method handles POST requests to this endpoint. The image file is read from the request, converted to a NumPy array, and decoded using OpenCV. The image is resized to 150x150 pixels to match the input size expected by the models. The image is reshaped into a 4D array with dimensions (1, 150, 150, 3) to fit the model's input requirements. The preprocessed image is fed into a pre-trained detection_model to predict the presence of a skin disease. The result is interpreted to label the image as either "Skin Disease" or "No Disease". If a skin disease is detected, a further classification is performed using a classification_model. The classification result is checked to determine the specific type of skin disease:

1. 0: Not classified, user advised to consult a dermatologist.

2. 1, 2, 3: Corresponds to specific diseases ("Onychomycosis", "Tinea Corporis", "Tinea Pedis"), with a recommendation to consult a dermatologist for confirmation.

3. Any other label: Indicates the model is not trained for that classification, with advice to consult a dermatologist.

A message is constructed based on the detection and classification results. The message is returned as a JSON response using Flask's jsonify function.

6.5 CODE FOR DERMATOLOGISTPAD

```
const canvas = document.getElementById("paintCanvas");

   const ctx = canvas.getContext("2d");

   let isDrawing = false;

   let tool = "pen";



   function setTool(newTool) {

     tool = newTool;

   }
```

```
function startDrawing(e) {
  isDrawing = true;
  draw(e);
}

function stopDrawing() {
  isDrawing = false;
  ctx.beginPath();
}

function draw(e) {
  if (!isDrawing) return;

  // Set the pen and eraser sizes
  const penSize = 5;
  const eraserSize = penSize * 2;

  ctx.lineCap = "round";

  if (tool === "pen") {
    ctx.globalCompositeOperation = "source-over";
    ctx.strokeStyle = "black";
```

```
      ctx.lineWidth = penSize;
    } else if (tool === "eraser") {
      ctx.globalCompositeOperation = "destination-out";
      ctx.strokeStyle = "rgba(255,255,255,1)";
      ctx.lineWidth = eraserSize;
    }

    ctx.lineTo(e.clientX - canvas.offsetLeft, e.clientY - canvas.offsetTop);
    ctx.stroke();
    ctx.beginPath();
    ctx.moveTo(e.clientX - canvas.offsetLeft, e.clientY - canvas.offsetTop);
  }

  function saveCanvas() {
// Generate a unique name using timestamp and random number
const uniqueName = `image_${Date.now()}_${Math.floor(Math.random() * 1000)}`;

// Create a new canvas to avoid affecting the original drawing canvas
const saveCanvas = document.createElement("canvas");
saveCanvas.width = canvas.width;
saveCanvas.height = canvas.height;
const saveCtx = saveCanvas.getContext("2d");
```

```
// Set the background color to white

saveCtx.fillStyle = "white";

saveCtx.fillRect(0, 0, saveCanvas.width, saveCanvas.height);


// Draw the content from the original canvas onto the new canvas

saveCtx.drawImage(canvas, 0, 0);


// Save the new canvas as an image

const image = saveCanvas.toDataURL("image/png");


// Create a link and trigger a click to download the image

const link = document.createElement("a");

link.href = image;

link.download = `${uniqueName}.png`;

link.click();


// Save the image data to the input file element

const imageFileInput = document.getElementById("imageFile");

const blob = dataURItoBlob(image);

const file = new File([blob], `${uniqueName}.png`, { type: 'image/png' });
```

```
    // Create a FileList containing the file
    const fileList = new DataTransfer();
    fileList.items.add(file);

    // Assign the FileList to the input file element
    imageFileInput.files = fileList.files;

    sendImageToServer();
}

function sendImageToServer() {
    const patient_id = document.getElementById("patient_id").value;
    const dermatologist_id = document.getElementById("dermatologist_id").value;
    const imageFileInput = document.getElementById("imageFile");

    // Create FormData object to send data
    const formData = new FormData();
    formData.append("patient_id", patient_id);
    formData.append("dermatologist_id", dermatologist_id);
    formData.append("image", imageFileInput.files[0]);
    // Send POST request using fetch API
    fetch("upload_prescription.php", {
```

```
        method: "POST",
        body: formData
    })
    .then(response => {
        if (response.ok) {
            console.log("Image sent successfully.");
            // Optionally, you can perform some action after sending the message
            // For example, clear input fields, update UI, etc.
        } else {
            console.error("Failed to send image.");
        }
    })
    .catch(error => {
        console.error("Error:", error);
    });
}

// Convert data URI to Blob
function dataURItoBlob(dataURI) {
    const byteString = atob(dataURI.split(',')[1]);
    const mimeString = dataURI.split(',')[0].split(':')[1].split(';')[0];
    const ab = new ArrayBuffer(byteString.length);
```

```
    const ia = new Uint8Array(ab);

    for (let i = 0; i < byteString.length; i++) {

        ia[i] = byteString.charCodeAt(i);

    }

    const blob = new Blob([ab], { type: mimeString });

    return blob;

}

      function clearCanvas() {

        ctx.clearRect(0, 0, canvas.width, canvas.height);

      }



      canvas.addEventListener("mousedown", startDrawing);

      canvas.addEventListener("mousemove", draw);

      canvas.addEventListener("mouseup", stopDrawing);

      canvas.addEventListener("mouseout", stopDrawing);
```

6.5.1 DESCRIPTION

The canvas element is selected from the HTML document, and its 2D rendering context is obtained. Initial variables are set: isDrawing to track the drawing state and tool to store the current drawing tool (pen or eraser). The setTool function allows changing the drawing tool between "pen" and "eraser". startDrawing(e): Activates drawing mode and starts drawing. stopDrawing(): Deactivates drawing mode and resets the drawing path. draw(e): Draws on the canvas if drawing mode is active. The pen and eraser sizes are set, and different drawing behaviors are defined for each tool using the globalCompositeOperation property. saveCanvas(): Saves the current canvas content as an image.

1. A unique name for the image file is generated.

2. A new canvas element is created to avoid altering the original canvas.

3. The new canvas is filled with a white background, and the original canvas content is drawn onto it.

4. The image is saved as a PNG data URL, and a download link is created and triggered.

5. The image is also stored in an input file element for subsequent upload.

sendImageToServer(): Sends the saved image to a server.

1. Retrieves patient_id and dermatologist_id from input fields.
2. Creates a FormData object and appends the necessary data.
3. Sends the data to a server endpoint (upload_prescription.php) via a POST request using the Fetch API.
4. Logs the result of the image upload process.

dataURItoBlob(dataURI): Converts a data URI to a Blob object for file handling. clearCanvas(): Clears the entire canvas. Adds event listeners to handle mouse interactions with the canvas:

1. mousedown to start drawing.

2. mousemove to draw.

3. mouseup and mouseout to stop drawing.

6.6 OPERATIONAL DIAGRAM

In figure 6.1 Operational Diagram, when user uses the TELECLINIC (Skin) Web Application, an http request is send through the internet by the user to the web server. The web server receives the request send by the client. The web server send a reponse to the client and displays the html pages of the web application. An HTTP connection between client and web server is established. Web server fetches data for signup, login, chats, appointments, skin images and prescription. The web server send request to the skin disease detection system and receive back a response.

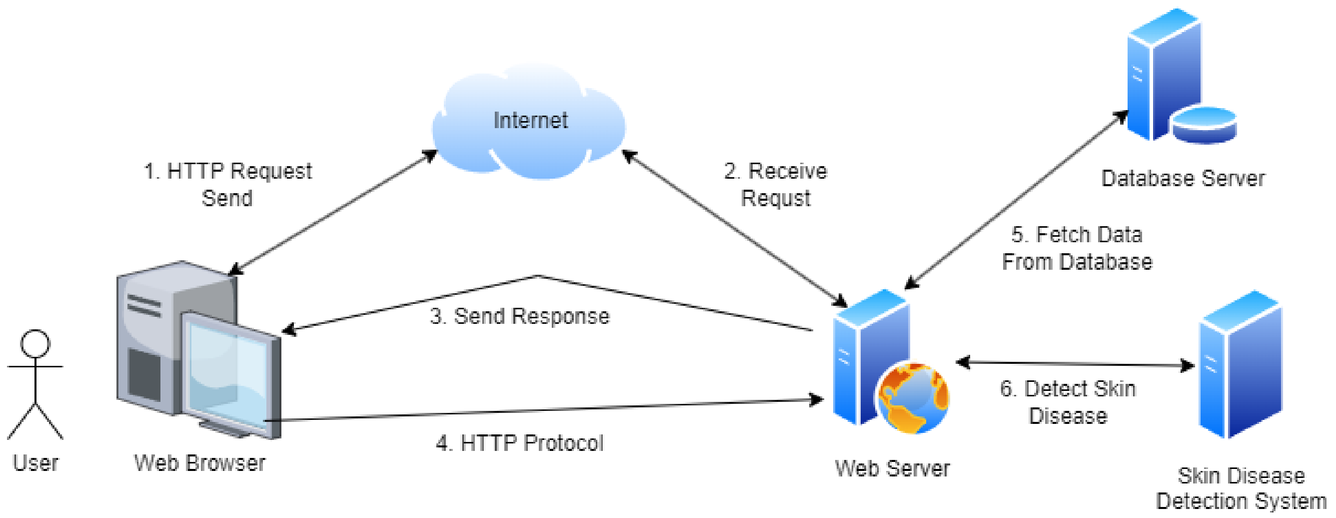


Figure 6. 1 OPERATIONAL DIAGRAM

6.7 COMPONENT DIAGRAM

In figure 6.2 in this web application these are the components. This web application has the following seven main components.

a. DERMATOLOGIST

The dermatologist component provides functionality like createProfile(); which allows the dermatologist to create and update their profile. viewAppointments(); retrives a list of all upcoming appointments. chatWithPatient(patient_id); starts the chat with the specified patient. writePrescription(patient_id); creates prescription for the specified patient. analyzeSkinImage(patient_id); analyzes an uploaded skin image and provides results generated by the Skin Disease Diagonstic System.

b. PATIENT

The patient component provides functionality like createProfile(); which allows the patient to create and update their profile. bookAppointment(dermatologist_id, date_time); books an appointment with a specified dermatologist at the specified date and time. viewPrescription();

retrives all the prescriptions. uploadSkinIamge(image); uploads a skin image for analysis. chatWitDermatologist(dermatologist_id); starts a chat with the specified dermatologist.

c. CHAT

The chat component provides functionality like sendMessage(patient_id, dermatologist_id, sender, message); sends a message from one user to another. getMessages(id); retrives messages from the database.

d. APPOINTMENT

The appointment component provides functionality like scheduleAppointment(dermatologist_id, patient_id, date_time); creates a new appointment for specified dermatologist and patient for the specified date and time. cancelAppointment(appointment_id); cancels existing appointment. getAppointment(appointment_id); retrives details about the specified appointment. listAppointment(patient_id, dermatologist_id); lists all appointment for a specified patient or for a specified dermatologist.

e. PRESCRIPTION

The prescription component provides functionality like createPrescription(dermatologist_id, patient_id); creates a new prescription. getPrescription(prescription_id); retrives the specified prescription. listPrescription(patient_id); lists all the prescription for the specified patient. updatePrescription(prescription_id); updates an existing prescription.

f. SKIN IMAGE

The skin_image component provides functionality like uploadSkinImage(patient_id, image); uploads a skin image for a specified patient. analyzeSkinImage(image); analyzes the uploaded skin image using AI model and provides the results. getImageAnalysis(image_id); retrives the results of the specific skin image. saveImage(image); saves the skin image to the database.

g. DERMATOLOGISTPAD

The dermatologistpad component provides functionality like createPrescription(dermatologist_id, patient_id); creates a new prescription. updatePrescription(prescription_id); updates an existing prescription.

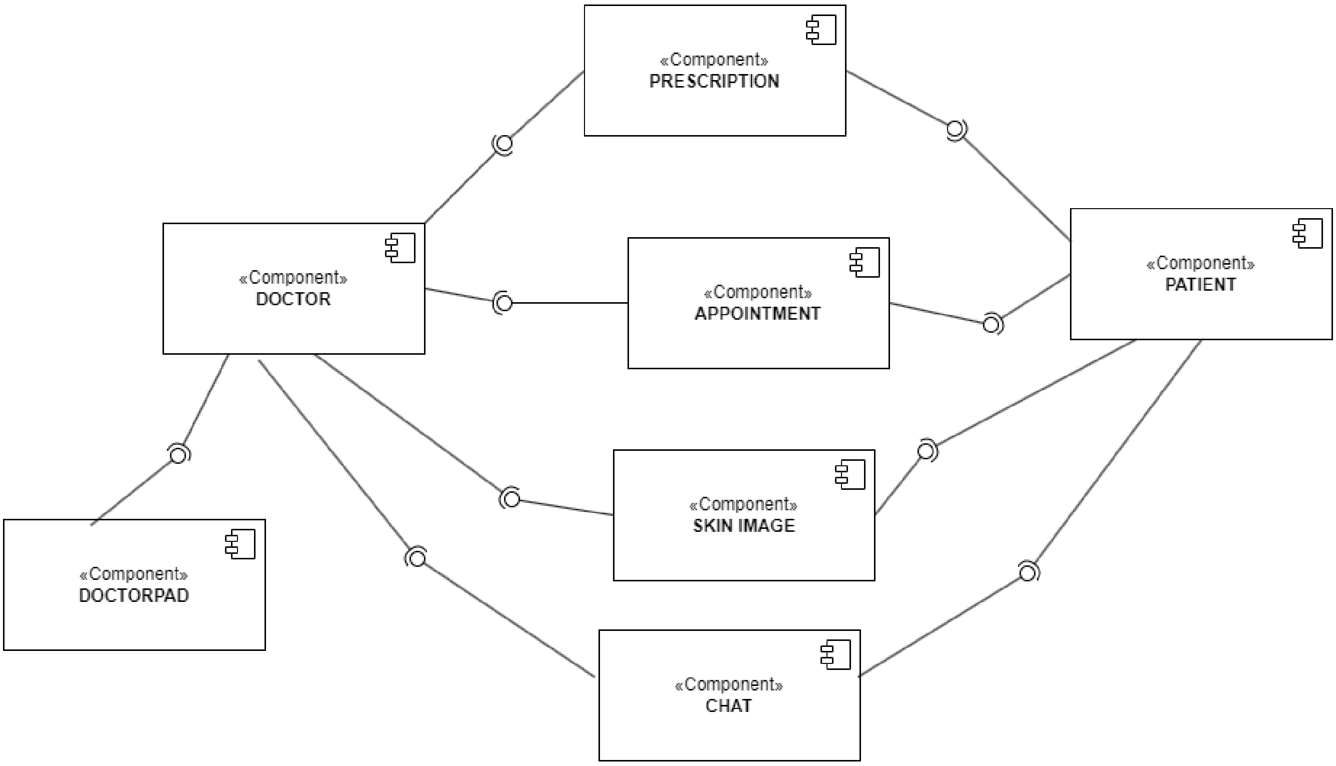


Figure 6. 2 COMPONENT DIAGRAM

6.8 DEPLOYMENT DIAGRAM

In figure 6.3 web application to deployed using two servers. One web server is used to server frontend clients while the other server which host the AI based Skin Disease Detection System is used internally by the backend to detect what kind of disease the patients has. Database server handles all the storing of data like patient and dermatologist record, chats, appointment, prescription and skin images.

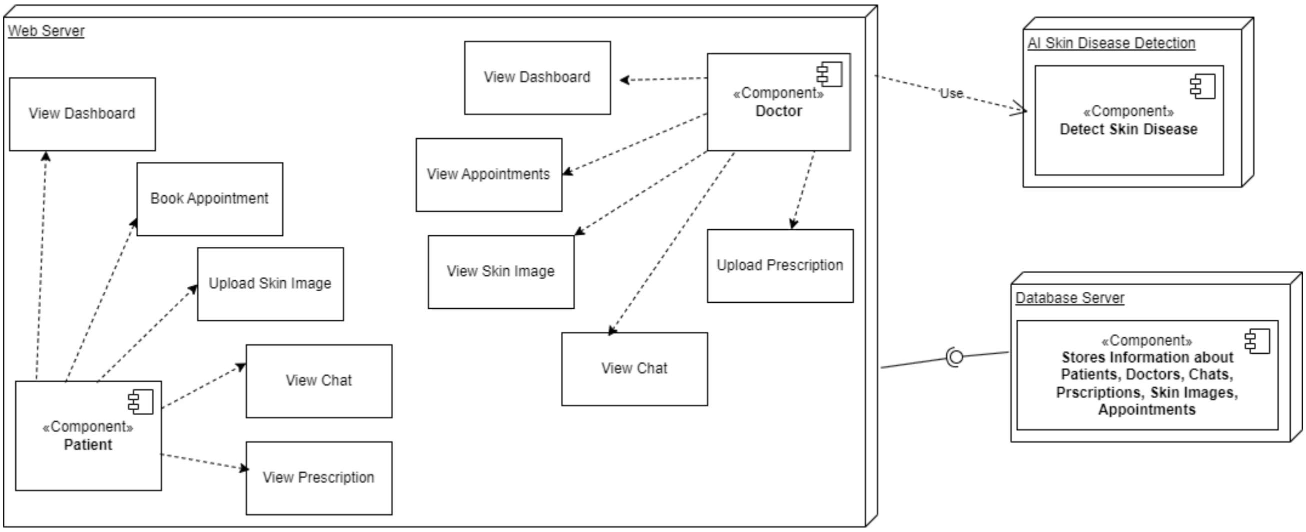


Figure 6. 3 DEPLOYMENT DIAGRAM

6.9 TECHNICAL INTERFACES:

Following are the technical interfaces of this project, which highlight the core functionality of the web application.

6.9.1 PATIENT DASHBOARD

In figure 6.4 is the dashboard screen which the patient gets to see when they signup or login.

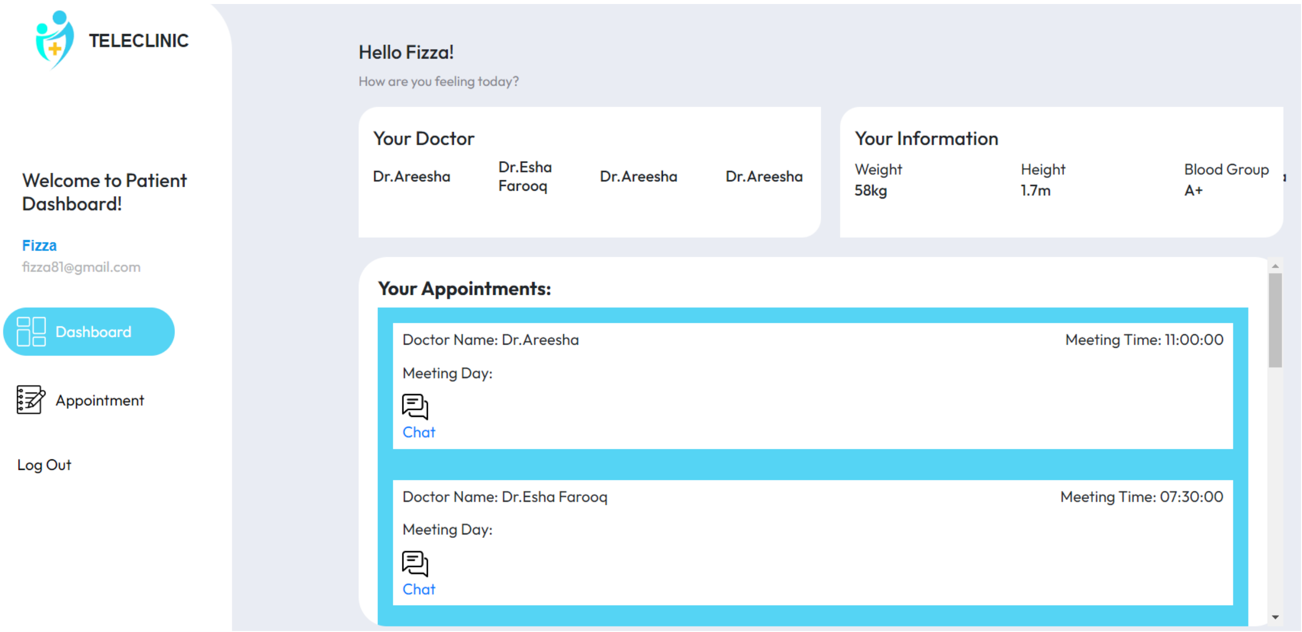


Figure 6. 4 PATIENT DASHBOARD

6.9.2 Dermatologist Dashboard:

In figure 6.5 is the dashboard screen which the dermatologist gets to see when they signup or login.

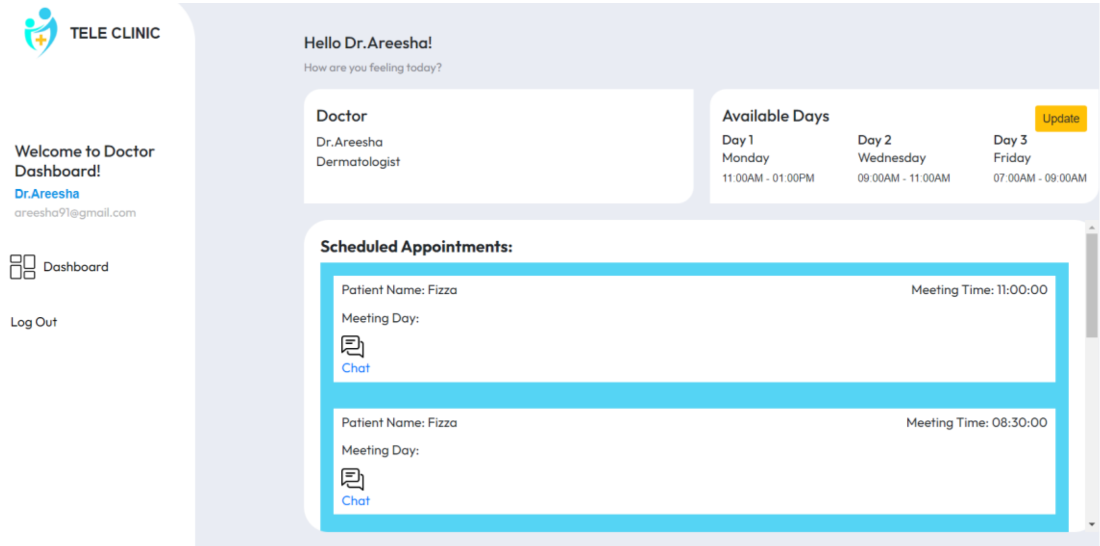


Figure 6. 5 DERMATOLOGIST DASHBOARD

6.9.3 Patient Booking Appointment:

In figure 6.6 and figure 6.7 patient can book appointnment for any available dermatologist.

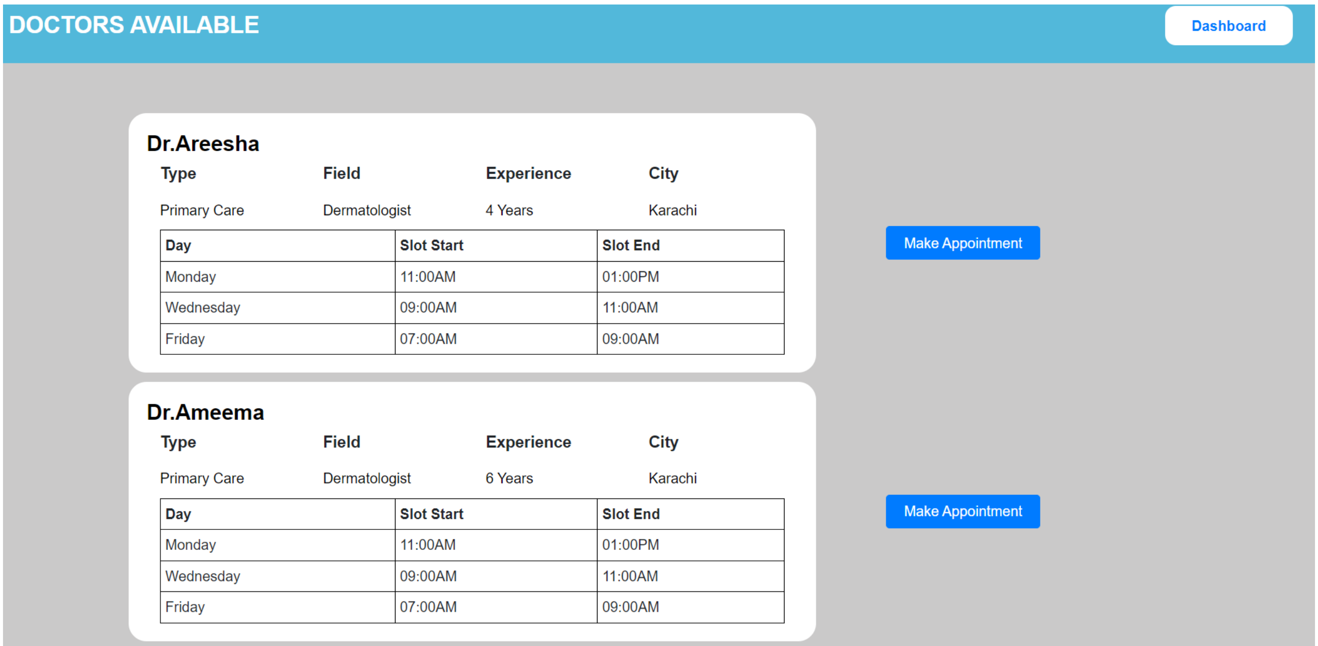


Figure 6. 6 PATIENT BOOKING APPOINTMENT

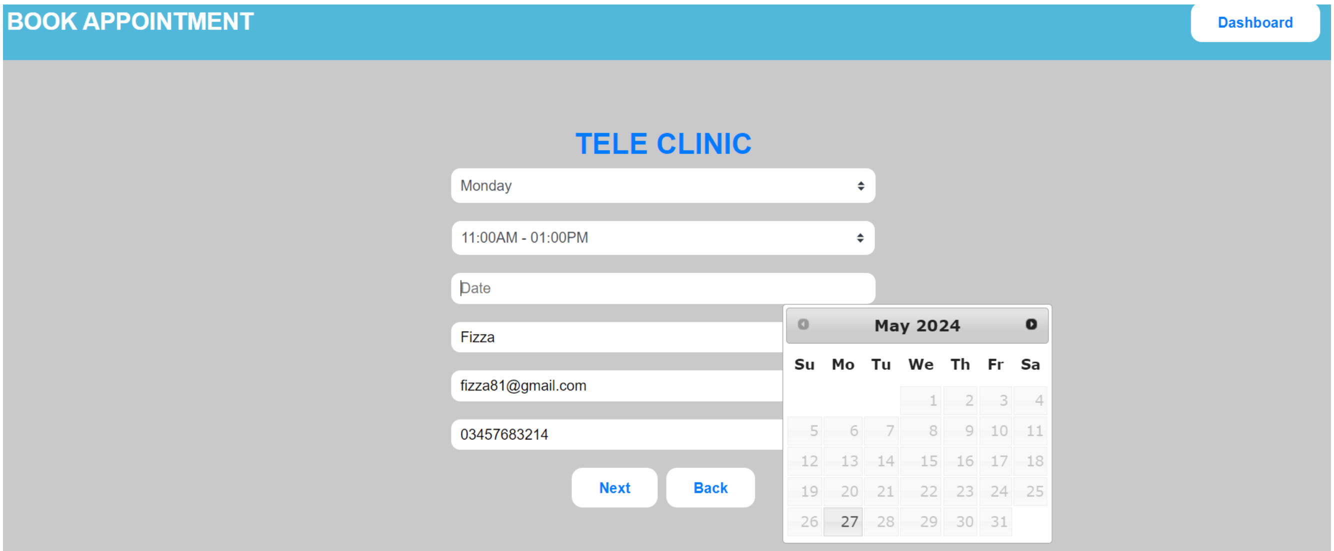


Figure 6. 7 PATIENT APPOINTMENT

6.9.4 CHAT BETWEEN PATIENT AND DERMATOLOGIST:

In Figure 6.8 and figure 6.9 interfaces shows chat between dermatologist and patient.

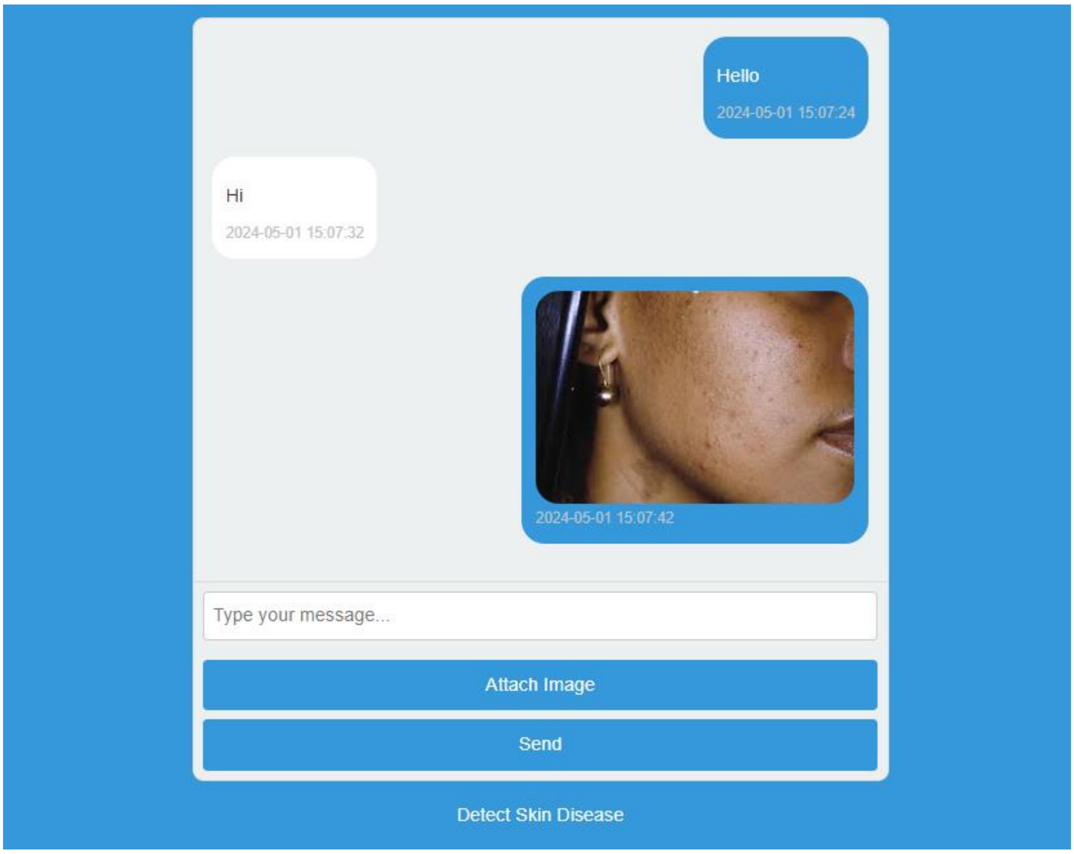


Figure 6. 8 PATIENT CHAT

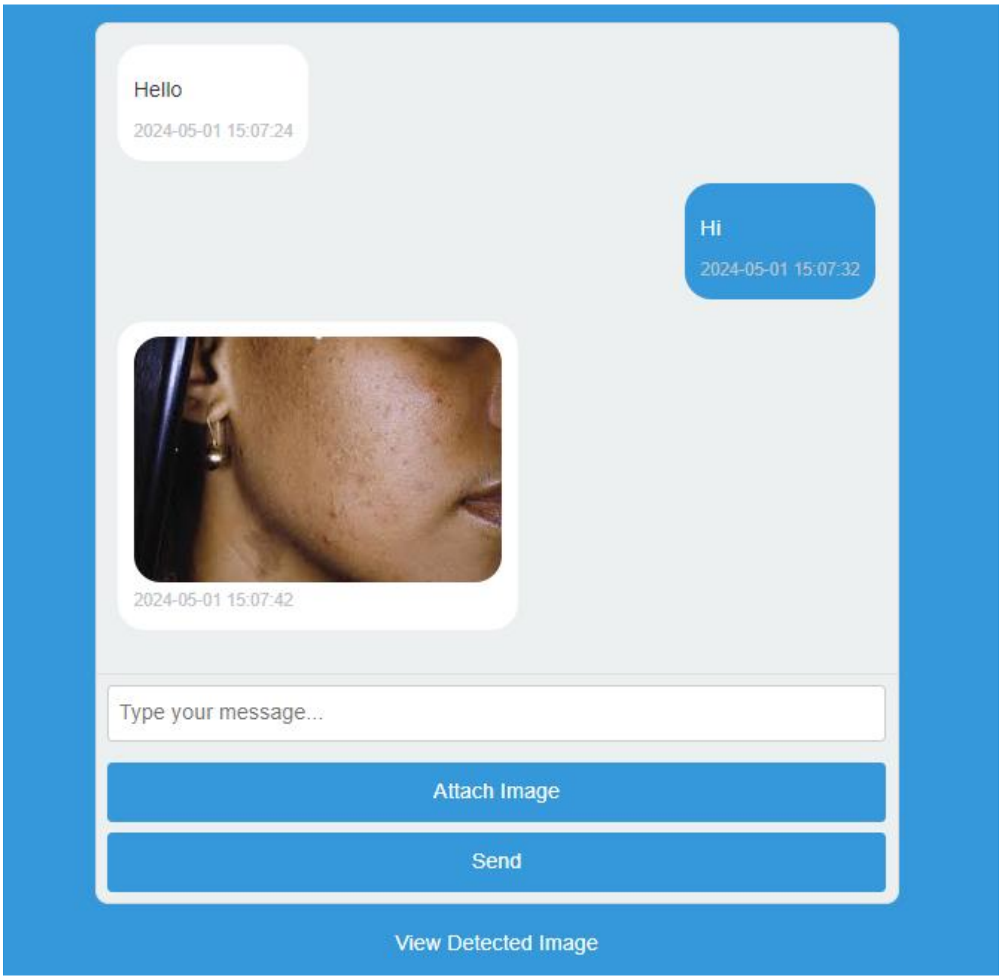


Figure 6. 9 DERMATOLOGIST CHAT

6.9.5 SKIN DISEASE DETECTION SYSTEM:

In Figure 6.10, figure 6.11 and figure 6.12 interfaces show how a patient can upload his/her skin image and the AI based Skin Disease Detection System will detect the skin disease.

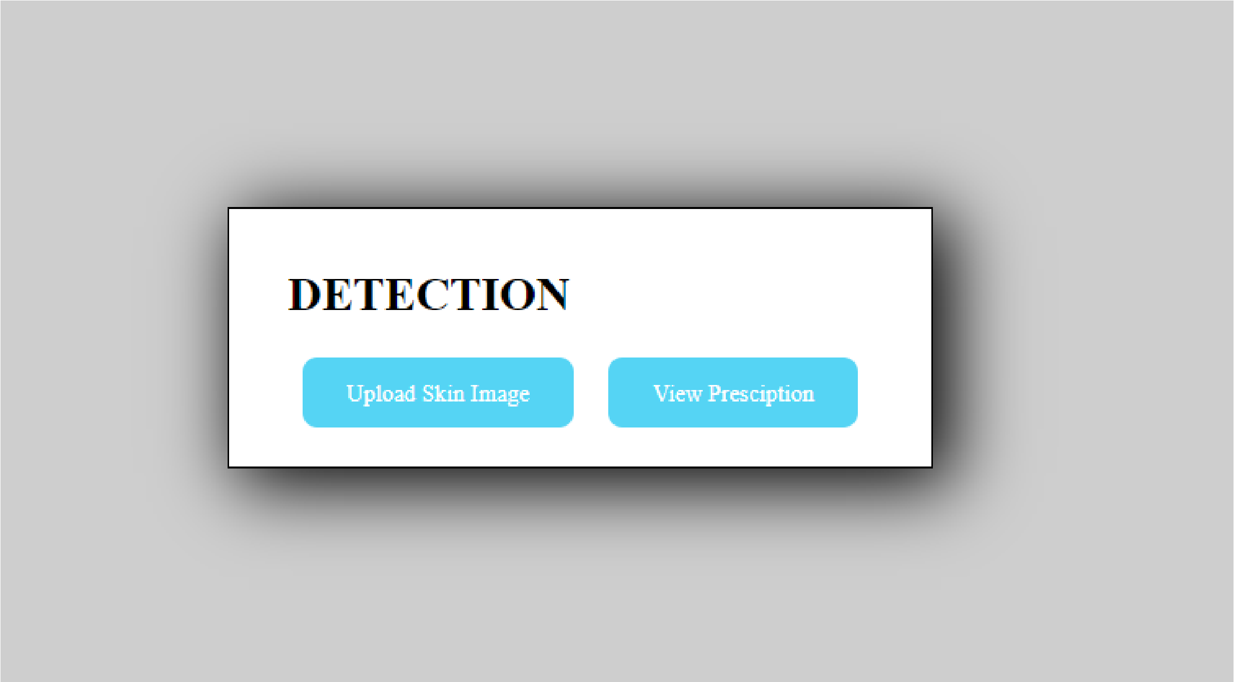


Figure 6. 10 PATIENT UPLOAD IMAGE AND VIEW PRESCRIPTION

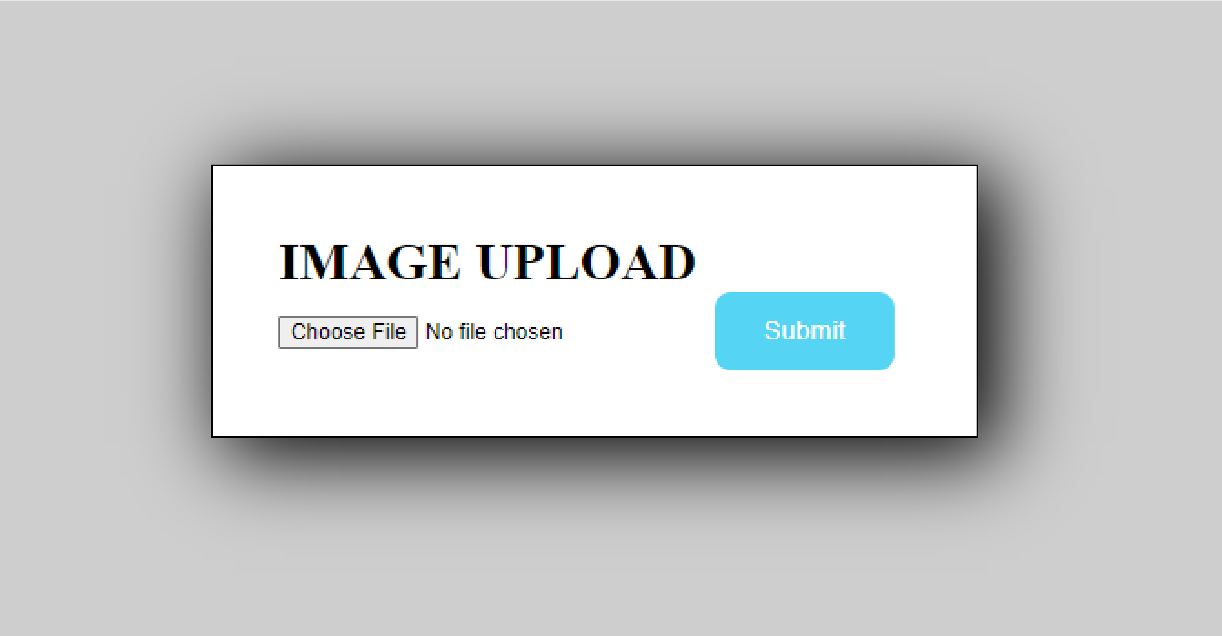


Figure 6. 11 PATIENT SKIN IMAGE UPLOAD

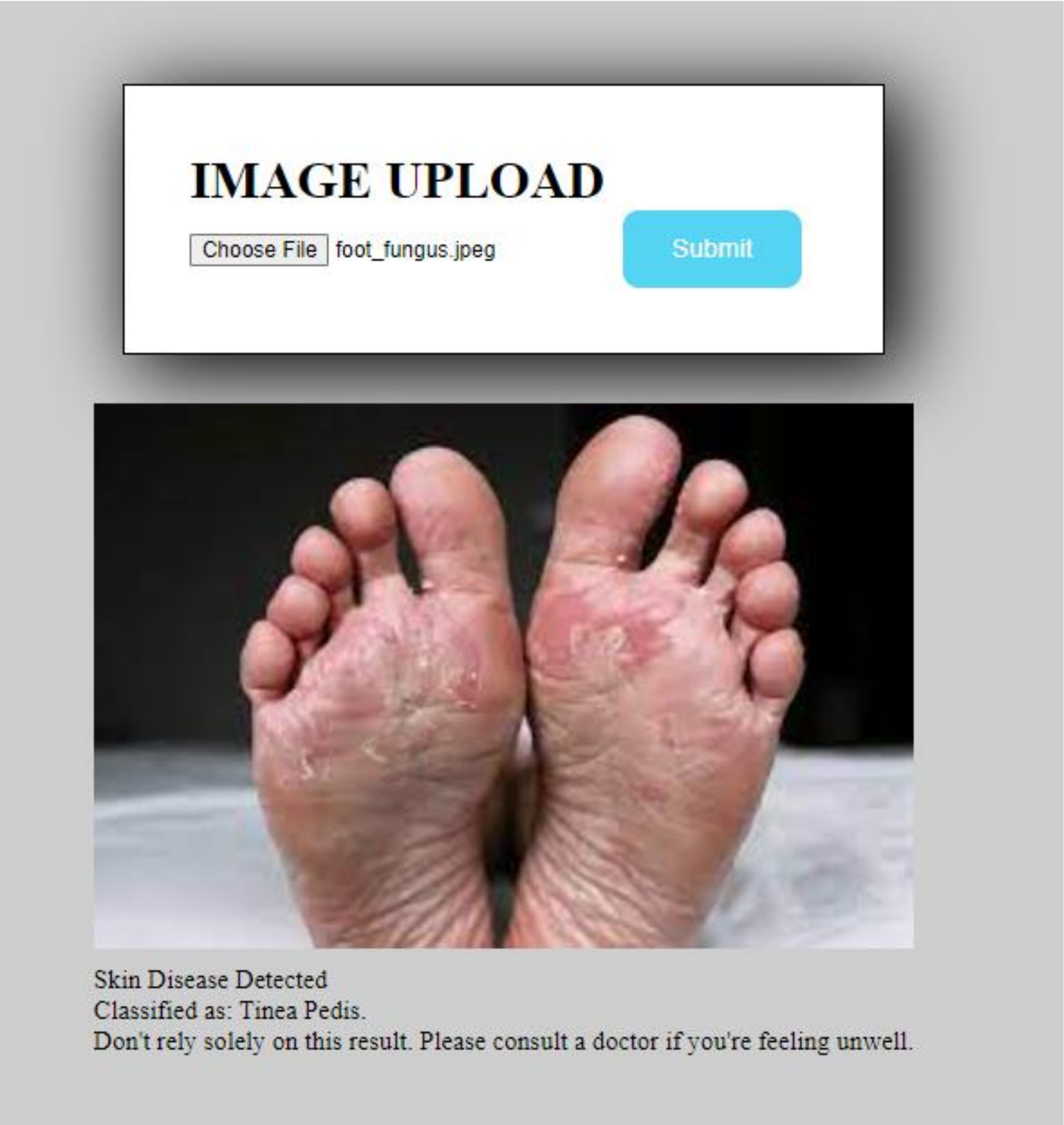


Figure 6. 12 AI SKIN DISEASE DETECTION SYSTEM

6.9.6 DERMATOLOGIST USING DERMATOLOGISTPAD:

In figure 6.13 the dermatologist will use this dermatologistpad to prescribe any test or treatment to the patient.

Figure 6. 13 DERMATOLOGISTPAD

6.10 STATE DIAGRAM:

In figure 6.14 and 6.15 is the state diagram that describe the overall view of the functionality of what patients and dermatologist can do in the web application.

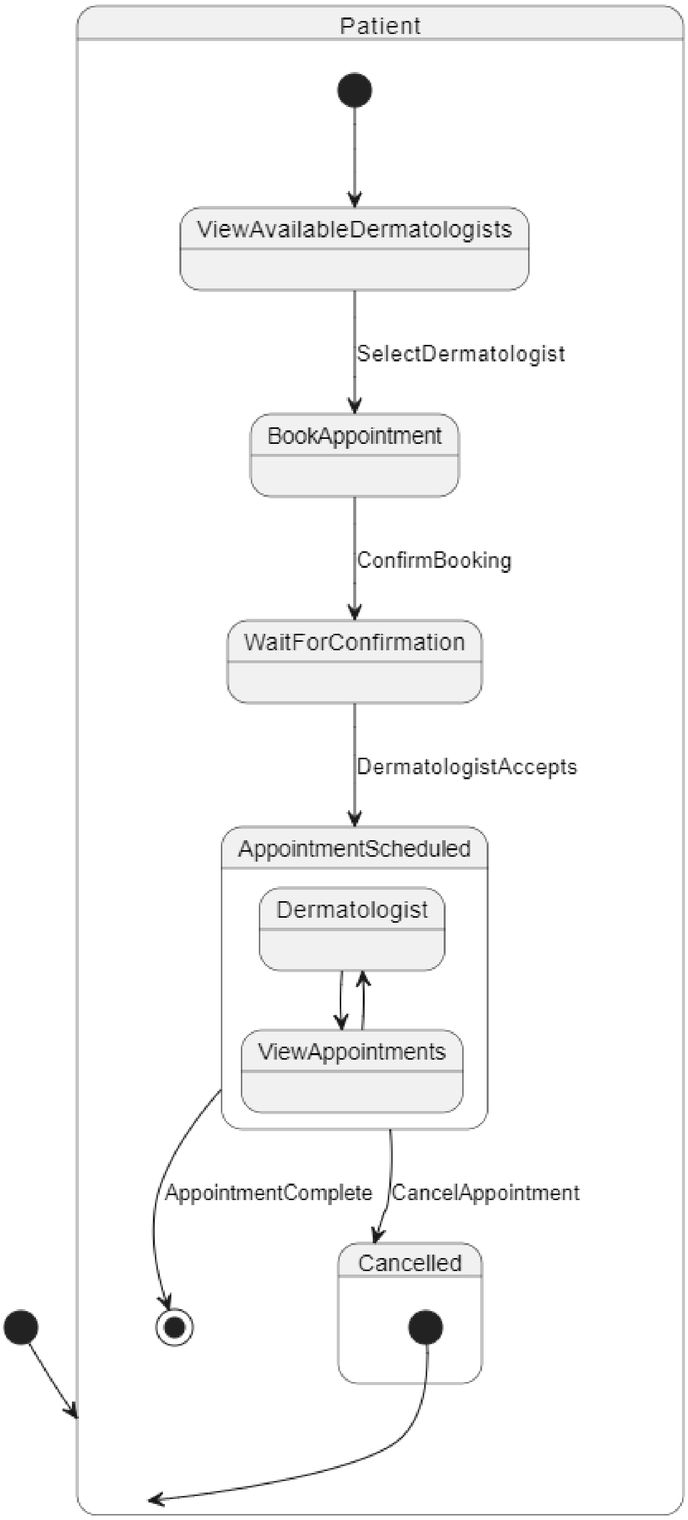


Figure 6. 14 STATE DIAGRAM FOR PATIENTS

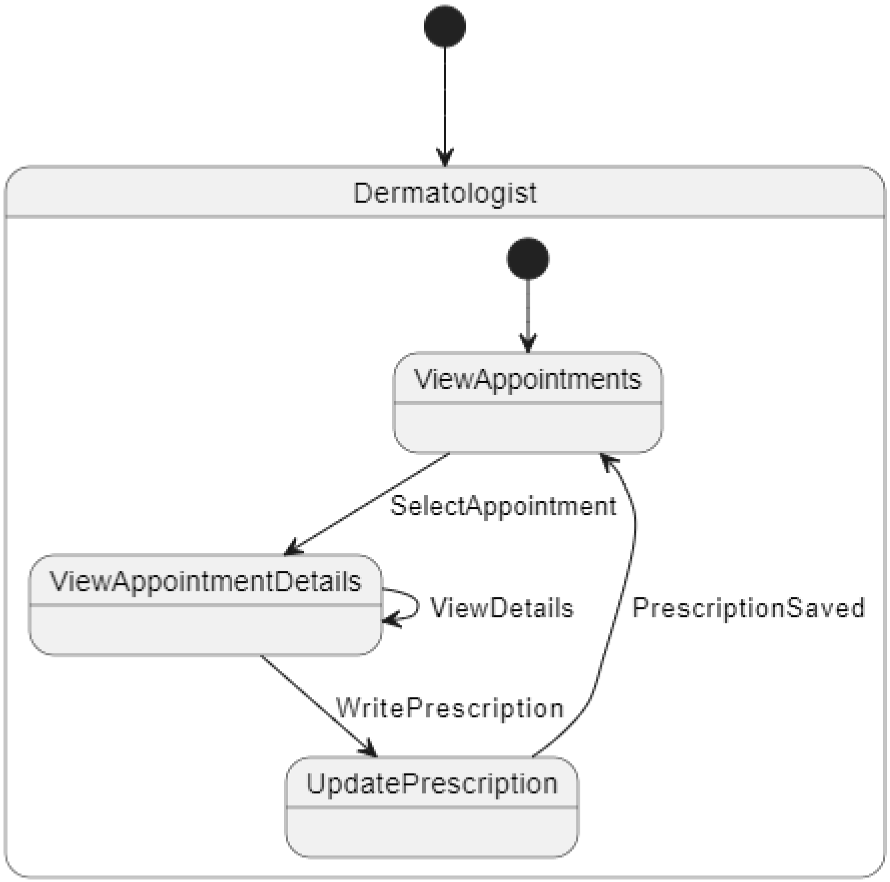


Figure 6. 15 STATE DIAGRAM FOR DERMATOLOGIST

CHAPTER VII

TESTING

7.1 INTRODUCTION

Testing is one of phases that are crucial in the development of any software / application. In this process, a thorough evaluation of software is done to ensure that the software reliably performs as intended. All errors are identified and corrected before proceeding to publish / deploy the code.

7.2 TESTING METHODS

In this chapter, mainly two types of testing methods will be discussed.

1. White Box Testing
2. Black Box Testing

7.2.1 WHITE BOX TESTING

White box testing, also known as clear box testing, glass box testing, or structure testing, is a method of software testing in which the configuration, internal structure, and usability of the software under test is determined by the tester who is aware about how the system works. White box testing involves analyzing and testing the source code at various levels, including integration testing, unit testing, and system testing. This helps in identifying issues such as logical errors, code inefficiencies, and security weakness of the system. The main advantage of white box testing is that it provides complete information and a deeper understanding of the code-base, resulting in more robust and reliable software.

7.2.2 BLACK BOX TESTING

Black Box Testing is a software testing technique in which the tester who is conducting these test, examines the functionality and behavior of the software without having any knowledge of internal implementation specifications, code structure, or internal procedures. Testers generate inputs and receive output compared to expected results, checking for correct behavior, error handling, and compliance with functional specifications. Black box testing is typically conducted at various levels, including functional testing, system testing, and acceptance testing. Finally the conclusion is particularly useful for validating the operating system, user interface, and overall system behavior, and provides an unbiased assessment of software quality from the user's perspective.

7.3 TESTING LEVELS

There are mainly two main levels of testing:

1. Functional Testing

2. Non – Functional Testing

7.4 NON – FUNCTIONAL TESTING

Non-Functional testing is a kind of software testing which focuses on testing passive aspects of a system, such as functionality, usability, reliability, and scalability Unlike functional testing, which means software perform specific tasks well.

7.4.1 PERFORMANCE TESTING

Checks the response of the system to various loads including stress testing, load testing and endurance testing.

7.4.2 USABILITY TESTING

Evaluates the interface’s simplicity and user-friendliness, and ensures that users can interact with the system effectively.

7.4.3 RELIABILITY TESTING

Assesses the stability and fault handling ability of the system over time.

7.4.4 SECURITY TESTING

Identifies vulnerabilities and ensures that the system remains secure from unauthorized access and threats.

7.4.5 SCALABILITY TESTING

Determines how well the system can be scaled up or down in response to requirements.

Non-Functional testing is essential to verify that the software meets quality standards and provides a good user experience, even in real-world situations.

7.5 FUNCTIONAL TESTING

Functional testing is a type of software testing that focuses on ensuring that a software application properly performs its intended functions according to specified requirements. This testing method involves evaluating system performance through input if appropriate to deliver and test results. Basic functional testing examines the overall API, user interface, security, database, application workflow and client/server application.

Some of the key features of functional testing include:

7.5.1 UNIT TESTING

Individual software components or modules are tested.

7.5.2 INTEGRATION TESTING

Verify that the integrated parts of an application work together as expected.

7.5.3 SYSTEM TESTING

Tests the complete and integrated software system and ensures that it meets specified requirements.

7.5.4 ACCEPTANCE TESTING

Validate the software as required by the project and ensure that it is ready for use.

Functional testing is essential to verify that the software works properly and fulfills users' needs, providing confidence that the application behaves as intended under various circumstances.

TABLE 7.1 FUNCTIONAL TESTING

Test Case Scenario	**Pre-Condition**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
User click on signup/login button.	User should have opened the URL.		GUI Work effectively.	Signup / login form is opened.	Pass
User enters an invalid email.	User should be on signup or login form.	majidgmail .com	Reject User form.	Error is generated telling the user to fill proper email.	Fail / Reject
User enters an invalid contact number.	User should be on signup form.	+9233232398	Reject User Form.	Error is generated telling the user to fill valid contact number.	Fail / Reject
User books an appointment with a dermatologist how is	User must sign up or login and on appointment page.	Dermatologist data (id, name, availability)	Successfully book appointment.	Generated success messages about	Pass

available on that time.				appointment booking.	
User books an appointment with a dermatologist how is not available on that time.	User must sign up or login and should be on appointment page.	Dermatologist data (id, name, availability)	Reject book appointment.	Generates an error telling the user that the dermatologist is not available on that time slot.	Fail / Reject
User uploads skin image.	User must be login and Image must be valid.	validate_image.jpg Or validate_image.png	Image will be uploaded.	Image uploaded successfully.	Pass
User enters a message.	User must be login.	Text message entered by the user.	Text message will be sent successfully.	Text messages is sent successfully.	Pass
Dermatologist upload a prescription.	Dermatologist must be login and the prescription image should be valid.	Prescription_1.png	Prescription will be uploaded successfully.	Prescription is uploaded successfully.	Pass
User password reset.	User must have an account.	Registered email.	Password reset successfully.	Password is reset successfully.	Pass

Dermatologist updates availability.	Dermatologist must be logged in.	Availability times, dates.	Availability updated successfully.	Availability is updated successfully.	Pass
Real-time chat functionality.	Both patient and dermatologist must be logged in.	Text messages, Attachments.	Messages exchanged in real-time.	Messages are displayed on both sides.	Pass
User cancels an appointment.	User must have a booked appointment.	Appointment ID.	Appointment canceled successfully.	Appointment is canceled successfully.	Pass
Dermatologist reschedules an appointment.	Dermatologist must have a booked appointment.	New date and time.	Appointment rescheduled successfully.	Appointment date and time is changed.	Pass
Skin image analysis result.	User must upload a valid skin image.	Valid skin image.	Analysis result provided.	Result is displayed.	Pass

7.5.5 CONCLUSION OF FUNCTIONAL TESTING

Functional testing is an important part of the software testing life cycle that focuses on ensuring that each function of a software application works as required. The main goal is to ensure that the application behaves as expected and meets performance and technology on the basis of requirements.

By systematically testing the above components, functional testing ensures that the software application not only meets its functional requirements but also provides a reliable and user-friendly experience. This technique is crucial for identifying and fixing problems early in the development cycle, ultimately resulting in superior product and increased user satisfaction.

7.6 ERROR HANDLING TESTING

Error Handling Testing focuses on ensuring that the system gracefully handles and responds to a variety of error conditions and special circumstances. The main goal is to ensure that the application can handle unexpected situations and provide meaningful information to the user. In Error Handling testing, test are run to understand areas where potential errors can be hidden.

TABLE 7.2 ERROR HANDLING TESTING

Interface	**Bug Description**	**Fix / Not Fix**
Image	When users browses and uploads an unsupported image format (eg; TIFF, BMP).	Fix
Image	User uploads an image with excessive file size.	Not Fix
Video	When a user uploads a video instead of image. There will be a bug.	Not Fix
Chat	User send messages to dermatologist.	Fix
Chat	User sends a message exceeding the maximum allowed length.	Fix
Chat	User sends a message with special characters or scripts.	Not Fix
Form	User enter valid or invalid detail in the form. The form instruct the user to insert valid details.	Fix
Form	User submits the form with missing required fields.	Fix
Form	User enters invalid email or phone number format.	Fix
Prediction	User uploads a skin image to get detection from the AI Skin Disease Detection System.	Fix
Prediction	User uploads an unsupported file format for skin image (e.g., PDF, DOCX).	Fix

Prediction	User uploads a corrupted or unreadable image file.	Fix
Prediction	AI model returns an error due to server or processing issues.	Not Fix
Prescription	Dermatologist uploads a prescription for the patient.	Fix
Prescription	Dermatologist uploads a prescription in an unsupported file format (e.g., DOCX instead of PDF).	Fix
Prescription	Dermatologist uploads a prescription with excessive file size.	Not Fix
Signup/Login	User attempts to sign up or log in with an invalid email or password.	Fix
Signup/Login	User attempts multiple failed login attempts in a short period.	Not Fix
Appointment	User attempts to book an appointment outside available hours or on a booked slot.	Fix
File Upload	User uploads a file while the server is temporarily down.	Fix
Data Processing	User's data fails to process due to backend issues.	Fix

7.6.1 CONCLUSION OF ERROR HANDLING TESTING

Error Handling Testing ensures that the application responds gracefully to unexpected situations and user error. This involves checking for error conditions such as invalid input, unsupported files, network problems, and unauthorized access. The goal is to ensure that the application provides meaningful error messages and maintains stability without degradation. Effective error handling enhances the overall user experience by providing guidance to users for troubleshooting and ensuring applications are robust and secure.

7.7 REGRESSION TESTING

Regression testing is an important process in software development whose goal is to ensure that updates or enhancements to the application do not introduce new bugs or regression. In addition to revalidating existing functionality, looking for changes impacting, managing test suites, automating test cases, comparing versions, detecting and reporting errors, and optimizing testing efforts. By creating systematic regression testing, organizations can maintain the robustness and independence of their software products increase, reduce the risk of post-implementation issues, and increase user satisfaction.

Regression testing of the web application is done each time changes are made to codebase. The web application is tested many times.

TABLE 7.3 REGRESSION TESTING

Test Case Scenario	**Test Case Steps**	**Test Case Data**	**Expected Result**	**Actual Result**	**Status**
User upload an image input on browser button or attach image button.	User selects input in form of image.	Image stored on device	Saved in Database.	Stored in directory and database.	Pass
User upload an image input for disease detection by detect button.	User clicks on detect button to start the detection process.	Image stored on device	Detection of Disease	Disease is accurately detected, and the name is displayed.	Pass
User uploads video instead of image.	Input should be in form of image.	Video stored on device.	Reject	System correctly rejects the video upload and displays an error message.	Fail

User selects low quality image.	Load image from device.	Image stored on device.	Detection of disease is difficult.	Disease name is displayed.	Pass
User selects a high quality image.	Load image from device.	Image stored on device.	Detection of disease is possible.	Disease name is displayed.	Pass
User attempts to upload an image with an unsupported file format.	User uploads a file in an unsupported format (e.g., PDF, DOCX).	Unsupported file	Reject	System correctly rejects the upload of unsupported file formats and displays an error message.	Pass
User attempts to upload an image with a file size exceeding the limit.	User uploads an image with a file size larger than the specified limit.	Large image file.	System rejects the upload, displaying an error message about exceeding the file size limit.	System Accepts the large image file.	Fail

7.7.1 CONCLUSION OF REGRESSION TESTING

Regression testing used includes verifying various issues related to image uploading and diagnosis processing. This includes ensuring proper image storage, accurate diagnosis, rejecting unsupported file formats and oversized files, handling low-quality images, and using appropriate error messages about the issues.

7.8 INTEGRATION TESTING

Integration testing is an important part of software development where individual software modules are combined and tested as a group to ensure they work together seamlessly. The main objective of integration testing is to identify communication faults between integrated components, ensure that they interact properly and fulfill intended functions. This testing procedure involves connecting modules and subsystems together sequentially, testing their connections, and verifying the data flow between them. Integration testing helps identify issues such as connection failures, data inconsistencies, and connection inconsistencies early in the development lifecycle, and ultimately ensures that the entire software system is reliable, robust and functional.

Integration testing is carried out by combining multiple components of a software as a group and then testing them. The aim of integration testing is to make sure that all different components work properly when integrated together.

TABLE 7.4 INTEGRATION TESTING

Test Cases	**Attributes**	**Description**	**Expected Result**	**Result**
Browse button / attach image button with image input.	Accept	Integrating the browse button to allow users to upload images.	Image uploaded successfully.	Pass
Detect Disease button integrated with image browsed	Succeed	Integrating the detect disease button for disease detection.	Successful detection of skin disease.	Pass
Integration of dermatologist pad with prescription upload.	Succeed	Integrating the dermatologist pad for prescription upload functionality.	Dermatologists can write prescriptions and treatments.	Pass
Video integration in appointment.	Succeed	Integrating video functionality for appointments.	Patients can conduct video calls with dermatologists.	Pass

Back button is integrated in the end.	Accept	Integrating a back button for navigation.	Returns user to the main browse image page.	Pass
Chat integration with appointment booking.	Succeed	Integrating chat functionality within appointment booking.	Patients can communicate with dermatologists for queries.	Pass
AI model integration with chat for instant diagnosis.	Succeed	Integrating AI model for instant diagnosis within chat.	Provides instant diagnosis and recommendations to patients.	Pass
Appointment scheduling integration with calendar system.	Succeed	Integrating appointment scheduling with a calendar system.	Automatically schedules appointments and updates the calendar.	Pass
User authentication integration with all modules.	Succeed	Integrating user authentication across all modules.	Ensures secure access and data integrity throughout the system.	Pass

7.8.1 CONCLUSION OF INTEGRATION TESTING

Integration testing of a project is testing the seamless integration of components and functionality in a web application. It includes browse detect patient button integration for image upload and disease detection, Dermatologistpad with prescription upload, video functionality for appointments, back button for navigation, chat integration and appointment booking, AI model integration for instant diagnosis in chat, A calendar system, and user authentication integration across all modules. Successful completion of integration tests ensures that all components work together in harmony, providing a seamless and efficient user experience.

7.9 UNIT TESTING

Unit testing is a basic process in software development where individual units or components of a software application are tested in isolation to ensure correctness and reliability The main purpose of unit testing is to ensure that each piece of code, like functions, methods, or classes , behave as expected Unit tests are typically automated and focused on testing small, granular units of code, and enable developers to identify and fix bugs early in the development process in By isolating and testing individual units, unit testing helps improve code quality, code maintenance It also enables re-components, and contributes to the overall stability and robustness of the software.

TABLE 7.5 UNIT TESTING

Buttons	Attribute	Description	Expected Result	Status
Attach Image / Browse Image Button	Functionality	Clicking the button triggers image upload functionality.	User successfully uploads an image through the browse button.	Pass
Detect Disease Button	Functionality	Clicking the button initiates skin disease detection.	The uploaded image is successfully processed, and the detected disease is displayed.	Pass
Prescription Upload Button	Functionality	Clicking the button enables prescription upload functionality.	Dermatologist successfully uploads a prescription.	Pass
Back Button	Navigation	Clicking the button redirects the user to the previous page.	User is redirected to the skin image upload page.	Pass

Chat Functionality	Functionality	Sending a message through the chat interface.	Message is successfully sent to the recipient.	Pass
AI Model	Functionality	Using AI model for instant diagnosis within chat.	AI model provides accurate diagnosis and recommendations.	Pass
User Authentication	Security	Logging in with valid credentials.	User gains access to authorized features and data.	Pass
Appointment Scheduling	Functionality	Creating an appointment with valid details.	Appointment is successfully scheduled in the system.	Pass
Image Processing Algorithms	Functionality	Testing individual image processing algorithms.	Algorithms correctly process images and return desired outputs.	Pass
Database Integration	Integration	Performing CRUD operations on the database.	Database operations are executed without errors.	Pass

7.9.1 CONCLUSION OF UNIT TESTING

Unit testing for a project involves testing individual components and their functions in isolation to ensure accuracy and reliability. This includes testing buttons for graphics, diagnosis, and prescription, as well as navigation functionality such as background buttons. Additionally, unit tests cover more complex features such as chat functionality, AI model integration, user authentication, appointment scheduling, image processing algorithms, database integration, etc. Performing unit tests well verifies that each component behaves as expected, contributing to the overall quality and robustness of the software application.

7.10 DECISION TESTING

Decision testing, also known as branch testing, is a software testing technique that focuses on evaluating different possible outcomes of decision points in the logic of a system. The primary purpose of decision testing is to implement and test scenarios all branches have been checked, such as if-else statements and switch-case statements covering true and false branches, boundary conditions and error paths at each decision point. By systematically evaluating decision points, developers can detect errors reasonably possible, ensuring the correctness of the conditional behavior, and increasing the overall reliability and quality of a software application.

TABLE 7.6 DECISION TESTING

Test Case	Conditions to execute	Input	Expected Result	Actual Result	Status
Browse: User selects image from the device.	Condition 1: If user browses for image.	Image in (.jpg)	User can browse and select skin images from the device.	Skin images are displayed on the screen.	Pass
Select Image: User will select which image to upload.	Condition 1: If selected image is of type (.jpg)	Selected image file (.jpg)	Selected image appears on the screen.	Image appears on screen and proceeds to next steps.	Pass
	Condition 2: If selected image is of type (.png)	Selected image file (.png)	Selected image appears on the screen.	Image appears on screen and proceeds to next steps.	Pass

	Condition 3: If selected iamge is not of type (.jpg) or (.png).	Selected image file (other format).	Selected image does not appear on the screen.	Image does not appear on screen.	Pass
Upload Prescription: Dermatologist selects prescription image to upload.	Condition 1: If selected image is of valid type.	Selected image file (valid format)	Selected prescription image appears on the screen.	Prescription image appears on screen.	Pass
Input Image for Detection: User uploads image for detection.	Condition 1: If uploaded image is of valid type.	Selected image file (valid format)	Uploaded image appears on the screen.	Image appears on screen and proceeds to next steps.	Pass
	Condition 2: If uploaded image is of invalid type.	Selected image file (invalid format)	Uploaded image does not appear on the screen.	Image does not appear on screen.	Pass
Appointment Scheduling:	Condition 1: If appointment time is available.	Valid appointment details	Appointment is successfully scheduled.	Appointment is successfully scheduled.	Pass
AI Model Integration:	Condition 1: If AI model successfully detects skin disease.	Valid image for detection	AI model detects skin disease successfully.	Skin disease is detected by the AI model.	Pass

User Authentication:	Condition 1: If user provides valid credentials.	Valid user credentials	User gains access to authorized features.	User successfully logs in with valid credentials.	Pass
Chat Functionality:	Condition 1: If message is successfully sent.	Valid message content	Message is successfully sent to the recipient.	Message is successfully sent to the recipient.	Pass
Image Processing Algorithms:	Condition 1: If image processing algorithm successfully detects disease.	Valid image for detection	Disease is detected by the algorithm.	Disease is detected by the algorithm.	Pass
Database Integration:	Condition 1: If database CRUD operations are successful.	Valid database operations	Database operations are executed without errors.	Database operations are executed without errors.	Pass

7.10.1 CONCLUSION OF DECISION TESTING

Decision testing of the test cases above focuses on analyzing the various states and consequences of user interaction in a web application. Each test represents a specific user action, such as searching images, selecting image files, uploading prescriptions, and planning. Implementation conditions include conditions, including valid and invalid inputs, file types, and system availability. By systematically testing these decision points, developers ensure that the application behaves as expected under various circumstances, processes user input correctly and provides appropriate responses or actions. Expected results verify the correct behavior of the application in response to each situation, while actual results confirm whether the application meets these expectations or ensures dependability to the web application, it works well, and is usable.

CHAPTER VIII

CONCLUSION

This project aims to develop and implement a web-based telemedicine platform equipped with AI diagnostic capabilities to increase access to dermatology healthcare services This platform is designed to allow patients to make appointments for virtual appointments with dermatologists, through various communication tools, including videocall and text messaging. The web application also has an AI Convolutional Neural Networks (CNN) [1] to detect and diagnose common skin diseases like Ringworm (Tinea Corporis), nail fungus, and Tinea Pedis (Athlete's foot). Leveraging AI technology the project seeks to improve efficiency and accessibility of dermatology consultations, especially for remote people.

This telemedicine platform application addresses the critical need for accessible and efficient dermatology healthcare services, especially for remote and underserved populations. With advanced AI and analytics capabilities that combine a user-friendly web interface with robust backend support, the platform offers comprehensive solutions for remote skin health consultation HTML, CSS, JavaScript a used for the front end ensures a responsive and interactive user experience, while PHP and MySQL provide a secure and reliable backend infrastructure and are A.I.

The project demonstrates the ability to use modern web technologies and machine learning to fill the gaps between patients and dermatologists, making skin care more accessible and effective. With proposed future enhancements, such as expanded AI capabilities, real-time counseling features, mobile application development and advanced security measures, the platform can continue to evolve and meet the growing demand for telemedicine is addressed This initiative not only strengthens healthcare but also sets a precedent for integrating AI into medical diagnostics.

CHAPTER IX

FUTURE ENHANCEMENT

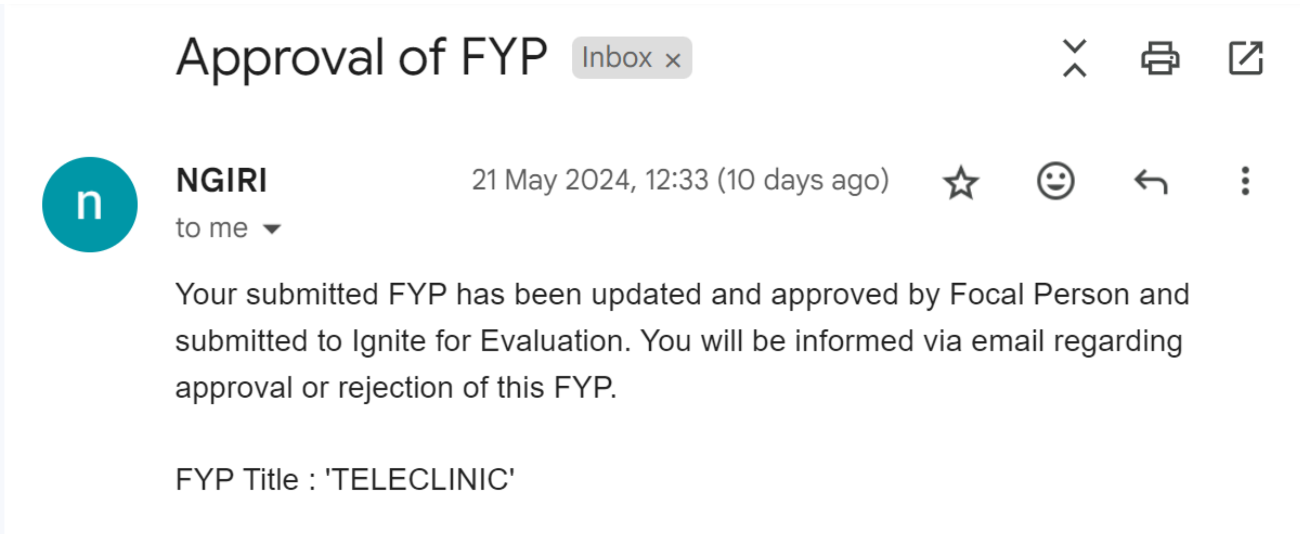
Since the scope of the project is big. The project has lot of possibilities. This project can be enhanced by expanding AI Skin Diesease Diagontic System to detect and diagnose more skin conditions beyond the first three (ringworm, nail fungus, and tinea pedis) This may require AI to be trained on large and diverse datasets to provide leveled and covered. Another improvement can be that the system can combine real-time video consulting features with advanced tools such as augmented reality (AR), dermatologists can flag areas of concern and comment directly on a live video feed on. This can increase the communication and effectiveness of remote conversations. Another imporvement is that the system can develop native mobile applications for iOS and Android platforms to provide users with a seamless and intuitive experience. Mobile apps can leverage device capabilities such as high-definition cameras and touchscreens to improve skin image capture and user interaction. Another improvement is that the system can implement advanced security measures such as end-to-end encryption of all communications, multi-factor authentication, and compliance with standards and health regulations such as Health Insurance Portabiliy and Accountability Act (HIPAA) [3] to ensure the privacy of patient data and security.

ACHIEVEMENTS

In this chapter, a breif summary of all the participations in different competitions, activities and exhibitions are listed.

APPLICATION TO IGNITE NATIONAL TECHNOLOGY FUNDS

TELECLINIC (Skin) has been successfully submitted for funding under the "Igniting University Innovation (NGIRI 2023-24)" initiative by the Ignite National Technology Fund. The application has been thoroughly reviewed and approved for the evaluation phase. The team is now awaiting the final results, which will determine whether the project will receive the funding support. This achievement highlights the innovative potential and the significant impact of the project, as recognized by a prestigious national platform.



3RD RESEARCH AND TECHNOLOGY SHOWCASE 2024 BY HIGHER EDUCATION COMMISSION (HEC)

The team successfully participated in the 3rd Research and Technology Showcase 2024 organized by HEC, where the team displayed the final year project. Initially scheduled for 18 April 2024 and later postponed to 1 June 2024, the event provided us with valuable lessons. The showcase allowed us to network with industry professionals and peers, receive valuable feedback, and gain recognition for the project's innovation. This experience has motivated us to pursue further excellence in the work.


Registration form for **3rd Research and Technology Showcase** by Sindh Higher Education Commission - 2024

Thank you for your response.

This form was created outside of your domain. Report Abuse - Terms of Service - Privacy Policy

Google Forms

REFERENCES

[1] Keiron O'Shea and R. R. Nash, "An Introduction to Convolutional Neural Networks," arXiv (Cornell University), Nov. 2015, doi: https://doi.org/10.48550/arxiv.1511.08458.

[2] Intersoft Consulting, "General data protection regulation (GDPR)," General Data Protection Regulation (GDPR), 2018. https://gdpr-info.eu/

[3] Edemekong PF, Annamaraju P, Haydel MJ. Health Insurance Portability and Accountability Act. In: StatPearls. StatPearls Publishing, Treasure Island (FL); 2023. PMID: 29763195.

[4] Tensorflow (2024, May, 27). *Tensorflow Developers* [Online]. Available: https://www.tensorflow.org/

[5] Pytorch (2024, May, 27), *Pytorch Developers* [Online]. Available: https://pytorch.org/

[6] M. Naqvi, S. Q. Gilani, T. Syed, O. Marques, and H.-C. Kim, "Skin Cancer Detection Using Deep Learning—A Review," *Diagnostics,* vol. 13, no. 11, p. 1911, 2023.

[7] N. S. A. ALEnezi, "A method of skin disease detection using image processing and machine learning," *Procedia Computer Science,* vol. 163, pp. 85-92, 2019.

[8] A. Trettel, L. Eissing, and M. Augustin, "Telemedicine in dermatology: findings and experiences worldwide–a systematic literature review," *Journal of the European Academy of Dermatology and Venereology,* vol. 32, no. 2, pp. 215-224, 2018.

[9] W.-Y. Hsu, "A customer-oriented skin detection and care system in telemedicine applications," *The Electronic Library,* vol. 37, no. 6, pp. 1007-1021, 2019.

[10] C. Marasca, A. Ruggiero, G. Fontanella, M. Ferrillo, G. Fabbrocini, and A. Villani, "Telemedicine and support groups could be used to improve adherence to treatment and health - related quality of life in patients affected by inflammatory skin conditions during the COVID - 19 pandemic," *Clinical and experimental dermatology,* vol. 45, no. 6, pp. 749-749, 2020.

[11] M. L. Staicu, A. M. Holly, K. M. Conn, and A. Ramsey, "The use of telemedicine for penicillin allergy skin testing," *The Journal of Allergy and Clinical Immunology: In Practice,* vol. 6, no. 6, pp. 2033-2040, 2018.

[12] M. N. Alam, T. T. K. Munia, K. Tavakolian, F. Vasefi, N. MacKinnon, and R. FazelRezai, "Automatic detection and severity measurement of eczema using image processing," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016: IEEE, pp. 1365-1368.

[13] S. Inthiyaz *et al.*, "Skin disease detection using deep learning," *Advances in Engineering Software,* vol. 175, p. 103361, 2023.

[14] V. B. Kumar, S. S. Kumar, and V. Saboo, "Dermatological disease detection using image processing and machine learning," in *2016 Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR)*, 2016: IEEE, pp. 1-6.

[15] R. P. Braun *et al.*, "Telemedical wound care using a new generation of mobile telephones: a feasibility study," *Archives of dermatology,* vol. 141, no. 2, pp. 254258, 2005.

[16] N. Zhang, Y.-X. Cai, Y.-Y. Wang, Y.-T. Tian, X.-L. Wang, and B. Badami, "Skin cancer diagnosis based on optimized convolutional neural network," *Artificial intelligence in medicine,* vol. 102, p. 101756, 2020.

[17] M. Hasan, S. D. Barman, S. Islam, and A. W. Reza, "Skin cancer detection using convolutional neural network," in *Proceedings of the 2019 5th international conference on computing and artificial intelligence*, 2019, pp. 254-258.

[18] S. A. AlDera and M. T. B. Othman, "A model for classification and diagnosis of skin disease using machine learning and image processing techniques," *International Journal of Advanced Computer Science and Applications,* vol. 13, no.

5, 2022.

[19] A. Nawar, N. K. Sabuz, S. M. T. Siddiquee, M. Rabbani, A. A. Biswas, and A. Majumder, "Skin disease recognition: a machine vision based approach," in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2021, vol. 1: IEEE, pp. 1029-1034.

[20] S. Akyeramfo-Sam, A. A. Philip, D. Yeboah, N. C. Nartey, and I. K. Nti, "A webbased skin disease diagnosis using convolutional neural networks," *International Journal of Information Technology and Computer Science,* vol. 11, no. 11, pp. 5460, 2019.

[21] I. S. S. Al Shabibi and S. Koottala, "Detection of Skin Diseases Using Matlab," *Journal of Student Research*, 2019.

[22] M. N. Bajwa *et al.*, "Computer-aided diagnosis of skin diseases using deep neural networks," *Applied Sciences*, vol. 10, no. 7, p. 2488, 2020.

[23] S. Gomathi, S. Nishanth, and S. Nikhitha, "Skin Disease Classification using Convolutional Neural Network," in *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, 2022: IEEE, pp. 177-182.

[24] T. A. Rimi, N. Sultana, and M. F. A. Foysal, "Derm-NN: skin diseases detection using convolutional neural network," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2020: IEEE, pp. 1205-1209.

[25] J. Deeks, J. Dinnes, and H. Williams, "Sensitivity and specificity of SkinVision are likely to have been overestimated," *J Eur Acad Dermatol Venereol*, vol. 34, no. 10, pp. e582-3, 2020.

[26] I. Malhi and Z. Yiu, "Algorithm - based smartphone apps to assess risk of skin cancer in adults: critical appraisal of a systematic review," *British Journal of Dermatology*, vol. 184, no. 4, pp. 638-639, 2021.

[27] T. M. de Carvalho, E. Noels, M. Wakkee, A. Udrea, and T. Nijsten, "Development of smartphone apps for skin cancer risk assessment: progress and promise," *JMIR Dermatology*, vol. 2, no. 1, p. e13376, 2019.

[28] H. F. Hasya, H. H. Nuha, and M. Abdurohman, "Real Time-based Skin Cancer Detection System using Convolutional Neural Network and YOLO," in *2021 4th International Conference of Computer and Informatics Engineering (IC2IE)*, 2021: IEEE, pp. 152-157.

[29] J. M. G. Aglibut, L. L. Alonzo, M. F. B. Coching, J. L. Torres, and N. B. Linsangan, "Skin disease identification system using gray level co-occurrence matrix," in *Proceedings of the 9th international conference on computer and automation engineering*, 2017, pp. 136-140.

[30] Keras (2024, May, 27). *Keras Developers* [Online]. Available: https://keras.io/

[31] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” Computer Science, 2014, doi: https://doi.org/10.48550/arXiv.1409.1556.

PLAGIARISM REPORT

CHAPTER 1 to 7:

TELECLINIC (SKIN)

ORIGINALITY REPORT

12%	9%	4%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Middlesex University Student Paper	2%
2	archderm.jamanetwork.com Internet Source	1%
3	Submitted to University of Wales Institute, Cardiff Student Paper	1%
4	Submitted to University of Hertfordshire Student Paper	<1%
5	www.coursehero.com Internet Source	<1%
6	Submitted to Berlin School of Business and Innovation	<1%