Tomato Care

By:

Muhammad Salman Afaq
24775
Usman Afaq
24779
Rafaqat Ahmad
24784

Supervised by:
Mr. Muhammad Usman Karim

Faculty of Computing
Riphah International University, Islamabad

Spring 2024

A Dissertation Submitted To

Faculty of Computing,

Riphah International University, Islamabad

As a Partial Fulfillment of the Requirement for the Award of

the Degree of

Bachelors of Science in Computer Science

Faculty of Computing
Riphah International University, Islamabad

Date: [02-05-2024]

Final Approval

This is to certify that we have read the report submitted by ***Muhammad Salman Afaq (24775),Usman Afaq (24779), Rafaqat Ahmad (24784)*** for the partial fulfillment of the requirements for the degree of the Bachelors of Science in Computer Science (BSCS). It is our judgment that this report is of sufficient standard to warrant its acceptance by Riphah International University, Islamabad for the degree of Bachelors of Science in Computer Science (BSCS).

Committee:

1 _____

Mr. Muhammad Usman Karim
(Supervisor)

2 _____

Dr. Muhammad Musharraf
(Head of Department)

Declaration

We hereby declare that this document "**Tomato Care**" neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this project with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of our teachers especially our supervisor **Mr. Muhammad Usman Karim**. If any part of the system is proved to be copied out from any source or found to be reproduction of any project from anywhere else, we shall stand by the consequences.

Muhammad Salman Afaq
24775

Usman Afaq
24779

Rafaqat Ahmad
24784

Dedication

Our final year project is dedicated to our parents, friends and teachers, whose love and support have been our pillars of strength. To our professors and especially supervisor "**Mr. Muhammad Usman Karim**", your guidance has shaped our academic journey.

Acknowledgement

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

We extend our heartfelt gratitude to our project supervisor "**Mr. Muhammad Usman Karim"**, whose unwavering support, invaluable guidance, and continuous mentorship were indispensable to the successful completion of this project. Their dedication and commitment have been a driving force behind our work.

Furthermore, we want to say a big thank you to our family and friends. They have been our constant source of support and motivation, always encouraging us to do our best and be honest and hardworking.

Muhammad Salman Afaq
24775

Usman Afaq
24779

Rafaqat Ahmad
24784

Table of Contents

List of Tables

List of Figures

Abstract

The project, titled **"Tomato Care"** addresses the critical challenge of safeguarding tomato crops against diseases that threaten global agricultural sustainability. Focused on Early Blight, Late Blight, and Leaf Mold and more, our objective is to develop an efficient automated system for early disease detection, providing farmers with a tool for timely intervention.

Leveraging Convolutional Neural Networks (CNNs) and transfer learning on a diverse dataset, our model achieved a validation accuracy of 90%. Real-world testing demonstrated its robustness, showcasing high sensitivity and specificity.

This Tomato Disease Detection System stands as a practical solution, emphasizing the potential of machine learning in precision agriculture for sustainable food production.

Chapter 1

Introduction

Tomato Care is a complete agricultural solution which is designed to solve the problems of farmers. This advanced platform, which is available for Android, leverages cutting-edge technology to revolutionize the way farmers detect and manage diseases in their tomato crops. With current methods falling short in addressing the complexities of tomato crop health, **Tomato Care** aims to provide an efficient and innovative solution to empower farmers. By seamlessly integrating image-based disease detection, treatment recommendations, and real-time updates, Tomato Care ensures a holistic approach to crop health management.

Detecting diseases in tomato plants is crucial for ensuring healthy yields and preventing crop loss. There are various diseases that can affect tomato plants, such as early blight, late blight, bacterial spot, and more. Traditional methods of disease detection often rely on visual inspection by experienced people, which can be time-consuming and subjective. However, advancements in technology, particularly in the field of machine learning have made way for more efficient and accurate methods of disease detection in tomatoes.

This user-friendly platform not only enhances the accuracy of disease identification but also encourages a collaborative environment where farmers can make informed decisions for optimal yields. The old ways of finding diseases in tomato plants are not always good enough, and it is hard to find and fix problems quickly. But Tomato Care is here to help. It uses advanced technology to make sure we find and understand the diseases in tomato crops better and faster.

1.1 Goals And Objectives

1.1.1 Goals

To utilize artificial intelligence and image analysis to process images of tomato plants, training models to recognize visual cues associated with various diseases.

To make the efforts of farmer less and make the system to detect the disease and suggest the treatment.

1.1.2 Objectives

To develop a system using smart technology to analyze pictures of tomato plants and automatically spot signs of diseases.

Create a monitoring setup that can quickly identify early signs of diseases in tomato crops, enabling timely and targeted interventions to prevent the spread of infections and minimize crop damage.

1.2 Scope of the Project

The project aims to create a user-friendly mobile application tailored for farmers, facilitating the identification of tomato plant diseases through image capture. By leveraging advanced technologies like machine learning, the app will enable precise disease recognition. Beyond mere identification, it will offer valuable recommendations for treatment, empowering farmers to take proactive measures against potential threats to their crops. Ensuring the app remains updated with the latest information on plant diseases is integral to its functionality. Ultimately, the project seeks to support farmers in safeguarding their crops, ultimately contributing to enhanced harvests and bolstered food production on a larger scale.

1.3 Summary

Detecting diseases in tomato plants is crucial for preventing crop loss and ensuring healthy yields. Traditional methods rely on visual inspections by experienced individuals, which can be time-consuming and subjective. However, advancements in machine learning have enabled more efficient and accurate disease detection. Technologies like image-based detection systems can identify diseases such as early blight, late blight, and bacterial spot quickly and precisely, enhancing the ability to manage and treat these diseases effectively.

Chapter 2

Literature Review

2.1 Introduction

Tomato Care is a complete agricultural solution which is designed to solve the problems of farmers. This advanced platform, which is available for Android, leverages cutting-edge technology to revolutionize the way farmers detect and manage diseases in their tomato crops. With current methods falling short in addressing the complexities of tomato crop health, Tomato Care aims to provide an efficient and innovative solution to empower farmers. By seamlessly integrating image-based disease detection, treatment recommendations, and real-time updates, Tomato Care ensures a holistic approach to crop health management. This user-friendly platform not only enhances the accuracy of disease identification but also encourages a collaborative environment where farmers can make informed decisions for optimal yields.

The Tomato Disease Detection market survey involves a thorough analysis of technology trends, and customer needs in the agricultural technology sector. By assessing market size, growth potential, and competitive dynamics, businesses can gain valuable insights for informed decision-making.

2.2 Background and Problem Elaboration

Farmers are grappling with a significant challenge posed by diseases impacting their tomato crops. Recognizing the urgency for a more efficient and accessible disease detection solution, extensive research was conducted to assess existing methods and identify farmers' needs. This exploration revealed a gap in the current solutions available. Drawing inspiration from innovative projects tackling similar issues in plant disease detection, we embarked on a journey to develop a solution tailored to address the specific needs of farmers and revolutionize disease detection in tomato crops.

Farmers face the critical challenge of timely and accurate detection of diseases in their tomato plants, which directly affects crop yield and livelihoods. Traditional methods of disease identification rely on visual inspection, which can be time-consuming. Farmers

need an accessible and reliable solution that leverages technology to identify diseases quickly and effectively in tomato plants, enabling them to take proactive measures for disease management and protect their agricultural investments.

2.3 Detailed Literature Review

The first paper explores the application of Convolutional Neural Networks (CNNs) for detecting diseases in tomato leaves. It likely discusses the architecture of the CNN model, including the number of layers, types of layers and activation functions used. The dataset used for training and testing the CNN model is likely described, including the number of images, classes of diseases, and any preprocessing steps applied to the images. The evaluation methodology is probably detailed, including the metrics used to assess the performance of the model (e.g., accuracy, precision, recall, F1-score). Additionally, the paper may discuss the significance of the results obtained, potential applications of the proposed approach in agriculture, and future research directions.

This second paper investigates the early detection and classification of tomato leaf diseases using a high-performance deep neural network. The architecture of the deep neural network, specifically the Inception V3 model, is likely described in detail, including its inception modules and pre-trained weights. The paper may explain the Rainbow concatenation technique used for feature extraction and classification. Experimental results, including accuracy, precision, recall, and F1-score, are likely presented and compared with existing methods. It also discusses the potential advantages of early disease detection in agriculture, such as improving crop yield and reducing pesticide use.

This third paper likely focuses on tomato leaf disease detection using Convolutional Neural Networks (CNNs) and transfer learning with the Inception V3 model. It may provide insights into the transfer learning process, including how pre-trained weights from the Inception V3 model are fine-tuned for the task of tomato leaf disease detection. Experimental results, including accuracy, may be presented and compared with other methods. The paper may discuss the practical implications of the proposed approach,

such as its potential use in precision agriculture for early disease detection and management.

The fourth paper likely discusses the application of deep learning, specifically Convolutional Neural Networks (CNNs), for disease detection on tomato plant leaves. The architecture of the CNN model and any modifications made to adapt it to the task of disease detection are likely described. Experimental results, including accuracy and possibly confusion matrices, are likely presented and analyzed. The paper may discuss the limitations of the proposed approach and suggest areas for future research, such as improving model robustness and scalability for large-scale deployment in agricultural settings.

2.4 Literature Review Summary Table

The Table below shows some researched papers

Table 1: Literature Review

Paper ID	Domain	Document	Algorithm	Accuracy	Reference
1	Machine Learning	Tomato Leaf Disease Detection Using Convolutional Neural Networks	Convolutional Neural Networks (CNNs)	91.52%	https://ieeexplore.ieee.org/document/9988540/
2	Deep Learning	Early Detection and Classification of Tomato Leaf Disease Using High-Performance Deep Neural Network	Inception V3, Rainbow concatenation	92.52%	https://www.mdpi.com/1424-8220/21/23/7987
3	Deep Learning	Tomato Leaf Disease Detection using Convolution Neural Network	Convolutional Neural Networks (CNNs), Transfer learning with Inception V3	91.2%	https://www.sciencedirect.com/science/article/pii/S1877050920306906

4	Deep Learning	Disease detection on the leaves of the tomato plants by using deep learning	Convolutional Neural Networks (CNNs)	91%	https://ieeexplore.ieee.org/document/9397001

2.4.1 Market Survey

Table 2: Market Survey

Ref	Applications	Tomato-Disease Detection	Treatment Suggestions	Voice and Text	Localized Alerts
1	Detailed tomato cultivation	✓	✕	✕	✕
2	Tomato Cultivation Tips	✕	✓	✕	✕
3	Diagno Plant Tomato	✕	✓	✕	✕

2.5 Research gap

Between 2018 and 2023, research on tomato disease detection has advanced significantly with the integration of machine learning and computer vision techniques. However, several research gaps continue within this period. Firstly, there remains a significant lack of diversity in datasets utilized for training and testing machine learning models. Many studies have focused narrowly on specific diseases or confined geographic regions, resulting in datasets that do not adequately represent the full spectrum of tomato diseases or the diversity of growing conditions across different

regions. Secondly, while numerous studies have showcased the effectiveness of machine learning models in controlled environments or small-scale experiments, their practical deployment in real-world agricultural settings faces challenges. These challenges include ensuring scalability, robustness, and adaptability to varying conditions encountered in field settings, such as different lighting conditions, plant growth stages, and environmental variability. Finally, as machine learning models become increasingly complex, there is a growing demand for interpretability and explainability. Understanding how these models make decisions is crucial for gaining trust and acceptance among end-users, such as farmers. Addressing these research gaps will be essential for advancing the field of tomato disease detection and facilitating its practical application in agriculture.

2.6 Problem Statement

Farmers face the critical challenge of timely and accurate detection of diseases in their tomato plants, which directly affects crop yield and livelihoods. Traditional methods of disease identification rely on visual inspection, which can be error-prone and time-consuming. Farmers need an accessible and reliable solution that leverages technology to quickly and effectively identify diseases in tomato plants, enabling them to take proactive measures for disease management and protect their agricultural investments.

2.7Summary

In response to these challenges, our research highlighted the deficiencies of existing methods and the urgent need for a more effective solution. Inspired by innovative approaches in plant disease detection, we developed a cutting-edge platform tailored to the specific needs of tomato farmers. This solution leverages advanced technology to provide timely, accurate, and accessible disease detection, empowering farmers to proactively manage crop health and protect their livelihoods. Through this innovative approach, we aim to revolutionize disease detection in tomato crops, ensuring better yields and sustainability in agricultural practices.

Chapter 3

Requirements and Design

3.1 Requirements

In the area of agricultural technology, the search for efficient and accurate disease detection in tomato crops has led to the development of advanced solutions through Requirement Engineering. This process involves systematically capturing, analyzing, and defining the needs and specifications for a technology or system customize to detect diseases in tomatoes. Understanding the specific requirements is crucial for designing effective solutions that address the challenges faced by farmers in managing crop health.

From the types of diseases common in tomato plants to the technological capabilities required for timely detection. This introduction sets the stage for exploring the detailed requirements essential for creating effective and impactful solutions in the domain of tomato crop health management.

3.1.1 Functional Requirements

3.1.1.1 User / Farmer

Table 3: User FRs

ID	Requirements
1.1	User shall be able to Sign Up.
1.2	User shall be able to login to their account.
1.3	User shall be able to view his profile.
1.4	User shall be able to edit his profile.
1.5	User can upload images for disease detection.
1.6	User can take photo from the camera.
1.7	User can access a personalized calendar for plant care.
1.8	User can give feedback.
1.9	User can also interact with the FAQs. It will provide relevant answers.
1.10	User can also view the reports.

3.1.1.2 Admin

Table 4: Admin FRs

ID	Requirements
2.1	Admin shall be able to login into the system.
2.2	Admin shall be able to manage diseases.
2.3	Admin shall be able to manage treatments.
2.4	Admin shall be able manage user's profiles.
2.5	Admin can also manage localized alerts.
2.6	Admin can review the reports.
2.7	Admin shall be able to manage the FAQs.

3.1.2 Non-Functional Requirements

3.1.2.1 Performance

The system should provide real-time processing for disease detection to ensure prompt results for farmers.

3.1.3 Hardware and Software Requirements

3.1.3.1 Hardware Requirements

- Phone
- Server
- Storage

3.1.3.2 Software Requirements

- Database
- Programming Languages
- Framework
- Development Tools
- Version Control

3.2 Proposed Methodology

Tomato Care is a complete agricultural solution which is designed to solve the problems of farmers. This advanced platform, which is available for Android, leverages cutting-edge technology to revolutionize the way farmers detect and manage diseases in their tomato crops. With current methods falling short in addressing the complexities of tomato crop health, Tomato Care aims to provide an efficient and innovative solution to empower farmers. By seamlessly integrating image-based disease detection, treatment recommendations, and real-time updates, Tomato Care ensures a holistic approach to crop health management.

The methodology for simplifying disease detection for tomato plants and enhancing crop yields involves utilizing a user-friendly mobile app interface. Farmers capture photos of their tomato plants using smartphones, and the app employs image processing and machine learning algorithms to identify diseases and assess their severity. Based on the analysis, the app suggests appropriate treatment strategies tailored to the specific disease detected. It features a user-friendly interface with FAQs function.

Additionally, the app utilizes localized alerts and notifications about potential disease risks, enabling farmers to take proactive measures. By empowering farmers with accessible tools and information, the methodology aims to streamline disease management, ultimately contributing to improved crop yields and agricultural sustainability.

3.3 System Architecture

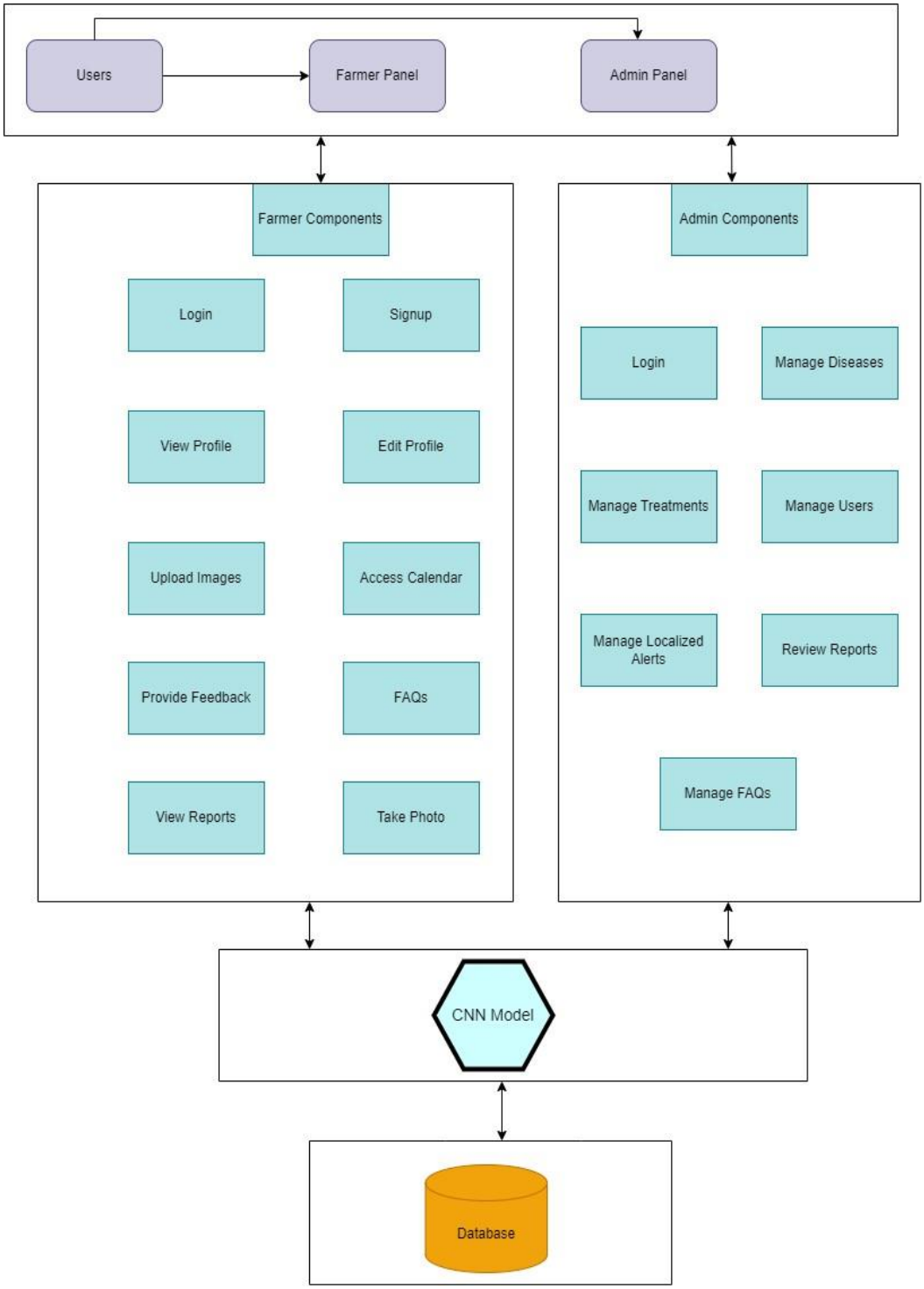


Figure 1: Architecture Diagram

3.4 Use Cases

3.4.1 Farmer

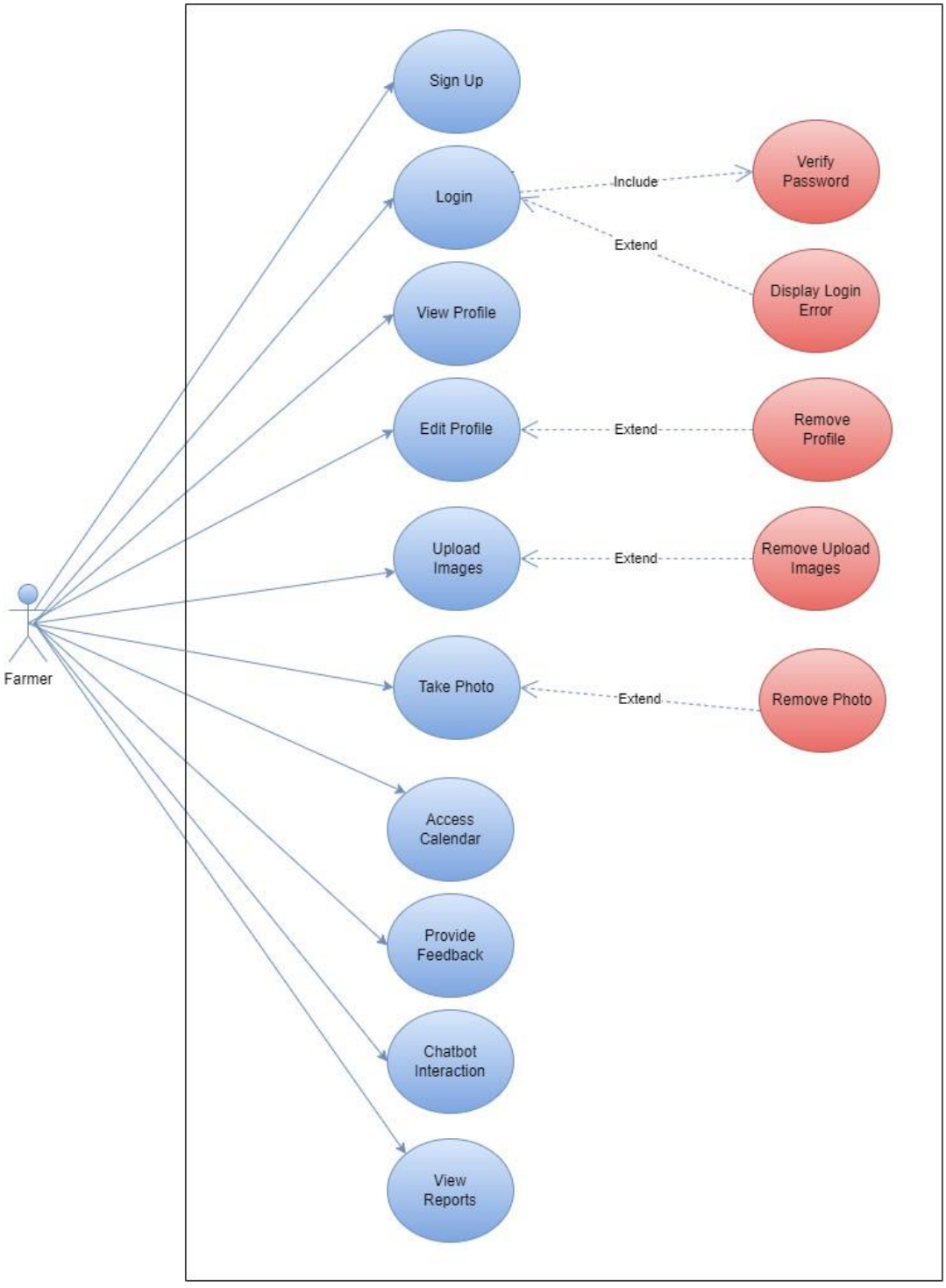


Figure 2: Farmer Use Case

3.4.2 Admin

Figure 3: Admin Use Case

3.5 Fully Dressed Use Cases

3.5.1 Sign Up

Table 5: Sign Up

Name	**Sign Up**		
Actors	Farmer		
Summary	The farmer registers a new account on the platform.		
Pre-Conditions	The farmer is not logged in to the application.		
Post-Conditions	The farmer creates an account with basic information like name, email, phone number, and password.		
Special Requirements	The email address should be unique and valid.		
Basic Flow			
Actor Action		**System Response**	
1	The farmer opens the Sign Up page.	2	The system displays the Sign Up form.
3	The farmer fills out the form with required information.	4	The system validates the information and creates an account upon successful validation.
Alternative Flow			
a	If the entered email is already registered, the system displays an error message and prompts the farmer to use a different email.	b	If the password doesn't meet the complexity requirements, the system prompts the farmer to choose a stronger password.

3.5.2 Login

Table 6: Login

Name	**Login**
Actors	Farmer, Admin
Summary	The farmer or admin accesses their account by providing login credentials.

Pre-Conditions	The user has a registered account.		
Post-Conditions	The user is logged into the platform and has access to their account features.		
Special Requirements	The password should be entered correctly.		
Basic Flow			
Actor Action		**System Response**	
1	The user opens the Login page.	2	The system displays the Login form.
3	The user enters their email address and password.	4	The system validates the credentials.
Alternative Flow			
a	If the entered credentials are incorrect, the system displays an error message and allows the farmer to attempt login again.	b	If the farmer forgets the password, there is a "Forgot Password" option that allows them to reset the password

3.5.3 View Profile

Table 7: View Profile

Name	**View Profile**		
Actors	Farmer, Admin		
Summary	The user views their profile information.		
Pre-Conditions	The user having securely authenticated their identity.		
Post-Conditions	The user can see their profile details like name, email address, phone number, and other relevant information.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user clicks on the "Profile" link or icon.	2	The system displays the user's profile page with their information.

3.5.4 Edit Profile

Table 8: Edit Profile

Name	**Edit Profile**		
Actors	Farmer, Admin		
Summary	The user updates their profile information.		
Pre-Conditions	The user has an access of his/her profile.		
Post-Conditions	The user's profile information is updated with the changes.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user clicks on the "Edit Profile" button.	2	The system displays the editable profile form with the user's current information.
3	The user modifies the desired information on the form.	4	The system validates the input and updates the user's profile with the changes.

3.5.5 Upload Images

Table 9: Upload Images

Name	**Upload Images**	
Actors	Farmer	
Summary	The farmer uploads images of their plants to identify potential diseases.	
Pre-Conditions	The farmer accepts the permission to use the album.	
Post-Conditions	The system analyzes the images and provides the farmer with potential disease diagnoses and information.	
Special Requirements	The image format should be supported by the system.	
Basic Flow		
Actor Action		**System Response**

1	The farmer clicks on the "Disease Detection" feature.	2	The system displays the image upload interface.
3	\The farmer selects and uploads the images of their plants.	4	The system analyzes the images using image recognition technology.
Alternative Flow			
a	If the uploaded images are of an unsupported format or size, the system prompts the farmer to upload valid images.	b	If there is an issue with the image processing, the system notifies the farmer and recommends re-uploading the images.

3.5.6 Take Photo

Table 10: Take Photo

Name	**Take Photo**		
Actors	Farmer		
Summary	The farmer uses the phone camera for disease detection.		
Pre-Conditions	The farmer can take photo from the phone camera.		
Post-Conditions	The system analyzes the photo and then detects the disease of the leaf.		
Special Requirements	The image format should be supported by the system.		
Basic Flow			
Actor Action		**System Response**	
1	The farmer clicks on the phone camera option.	2	The system displays the camera of the phone.
3	The farmer takes the photo from the camera.	4	The system analyzes the images and then give the disease of the leaf.
Alternative Flow			
a	If the photo is clear and the system understands the image, then the disease will be easily detected and the accuracy will be good.	b	If there is an issue with the photo and if the photo is not clear then the accuracy will not be that good.

3.5.7 Access Calendar

Table 11: Access Calendar

Name	**Access Calendar**		
Actors	Farmer		
Summary	The farmer receives a personalized calendar with recommended tasks and schedules for their specific crops based on their location and climate.		
Pre-Conditions	The farmer is ready to access the calendar for planning crops and tasks.		
Post-Conditions	The farmer has a customized calendar with reminders for essential plant care activities like: Watering, Fertilization, Pest and disease control, Harvesting		
Special Requirements	Plant care calendar section.		
Basic Flow			
Actor Action		**System Response**	
1	The system automatically generates a personalized calendar after the farmer provides required information.	2	The farmer can filter and prioritize tasks based on urgency or category.
3	The calendar displays tasks with dates, deadlines, and instructions for each activity.	4	The farmer can set reminders for upcoming tasks.

3.5.8 Provide Feedback

Table 12: Provide Feedback

Name	**Provide Feedback**
Actors	Farmer, Admin
Summary	The user provides feedback on the platform's features and functionalities.
Pre-Conditions	The application stands ready to receive and respond to the user's feedback.

Post-Conditions	The feedback is communicated to the admin for review and consideration.		
Special Requirements	Feedback form.		
Basic Flow			
Actor Action		**System Response**	
1	The user opens the "Feedback" section.	2	Describe the issue or suggestion in detail. Attach screenshots or relevant media for clarity. The system sends the feedback to the admin for review.
3	Submits the feedback.	4	System acknowledges the feedback.

3.5.9 FAQs

Table 13: FAQs

Name	**FAQs**		
Actors	Farmer, Admin		
Summary	The FAQs functionality will allow the user to ask the questions and will get relevant answers about the disease detection.		
Pre-Conditions	The User can interact with the FAQs		
Post-Conditions	The user receives answers about the asked questions		
Special Requirements	Security measures to protect sensitive information. Also Search functionality for users to quickly find relevant FAQs.		
Basic Flow			
Actor Action		**System Response**	
1	User accesses the FAQs page on the website.	2	System displays a list of frequently asked questions.

3.5.10 View Reports

Table 14: View Reports

Name	**View Reports**
Actors	Farmer, Admin

Summary	The user accesses and reviews reports generated by the platform to gain insights into their farming practices and crop performance.		
Pre-Conditions	Users are all set to open and check out reports in the application.		
Post-Conditions	The user gains valuable insights into their farming practices and crop performance based on the generated reports.		
Special Requirements	Reports should be generated regularly and updated with new data.		
Basic Flow			
Actor Action		**System Response**	
1	The user accesses the reporting section of the platform.	2	The system displays available reports, categorized by type, such as Crop Performance Reports, Resource Consumption Reports, Soil Health Reports, Environmental Impact Reports
3	The user selects a specific report to view.	4	The system displays the report in an interactive format, allowing the user to Filter and analyze data based on specific criteria.
Alternative Flow			
a	The system prompts the user to provide additional information or filter parameters to refine the report.	b	The user provides the requested information or modifies filter settings.
c	The system updates the report with the refined data.		

3.5.11 Login

Table 15: Login

Name	**Login**
Actors	Admin
Summary	The administrator accesses the platform's administration panel to manage various functionalities.

Pre-Conditions	The administrator has valid login credentials.		
Post-Conditions	The administrator has access to all administrative features.		
Special Requirements	Multi-factor authentication and secure session management.		
Basic Flow			
Actor Action		**System Response**	
1	The administrator opens the Login page.	2	The system displays the Login form.
3	The administrator enters their email address and password.	4	The system validates the credentials.
5	If credentials are valid	6	The system logs the administrator into the platform.
7	If credentials are invalid	8	The system displays an error message.
Alternative Flow			
a	The system displays an error message and prompts the user to re-enter their credentials.	b	The system informs the user that their account is locked and provides instructions for unlocking it.
c	The system sends a verification code to the administrator's registered phone number or email address for additional security.		

3.5.12 Manage Diseases

Table 16: Manage Diseases

Name	**Manage Diseases**
Actors	Admin
Summary	The administrator adds, edits, and updates information about various diseases affecting crops within the platform.
Pre-Conditions	The admin gains the ability to manage diseases within the application.

Post-Conditions	The administrator manages the disease database, ensuring accurate and up-to-date information for users.		
Special Requirements	Integration with a dynamic disease database, and support for classification.		
Basic Flow			
Actor Action		**System Response**	
1	The administrator opens the Disease Management section.	2	The system displays a list of existing diseases.
3	The administrator can: Add a new disease, Edit an existing disease, Delete a disease:	4	The system updates the disease database accordingly.
Alternative Flow			
a	The system prompts the administrator to complete all required fields before submitting the new disease information.	b	The system warns the administrator that the disease already exists and suggests potential actions, such as merging information or creating a new entry with different details.
c	The system validates the format and accuracy of the imported data before adding it to the platform.		

3.5.13 Manage Treatments

Table 17: Manage Treatments

Name	**Manage Treatments**
Actors	Admin
Summary	The administrator adds, edits, and updates information about various treatment options for different diseases.
Pre-Conditions	The admin is equipped to administer and oversee treatments within the application.

Post-Conditions	The administrator maintains a comprehensive list of treatment options for users to access and utilize.		
Special Requirements	Integration with a treatment database, update mechanism, and metrics for effectiveness.		
Basic Flow			
Actor Action		**System Response**	
1	The administrator opens the Treatment Management section.	2	The system displays a list of existing treatment options.
3	The administrator can: Add a new treatment, Edit an existing treatment, Delete a treatment	4	The system updates the treatment database accordingly.
Alternative Flow			
a	The system requires the administrator to provide clear warnings and disclaimers about potential risks associated with the treatment.	b	The system allows the administrator to edit and update the treatment information to ensure accuracy and currency.
c	The system prompts the administrator to provide more reliable references and evidence to support the treatment information.		

3.5.14 Manage Users

Table 18: Manage Users

Name	**Manage Users**
Actors	Admin
Summary	The administrator reviews, approves, and manages user accounts within the platform.
Pre-Conditions	The admin is empowered to oversee and manage users within the application.
Post-Conditions	The administrator ensures a safe and productive environment for platform users.
Special Requirements	Role-based access control, activity logs, and customizable profiles.
Basic Flow	

Actor Action		System Response	
1	The administrator opens the User Management section.	2	The system displays a list of all registered users.
3	The administrator can: Review user profiles, Approve or reject user registrations, Suspend or ban users, Assign user roles and permissions	4	The system updates user accounts and access accordingly.
Alternative Flow			
a	The system prompts the user to complete all required information before submitting the registration request.	b	The system may request additional documentation or proof of identity to verify the user's information.
c	The system ensures that user privacy is protected and sensitive information is not publicly accessible.		

3.5.15 Manage Localized Alerts

Table 19: Manage Localized Alerts

Name	**Manage Localized Alerts**
Actors	Admin
Summary	The administrator creates and sends localized alerts to users based on specific criteria.
Pre-Conditions	The admin gains direct authority to manage localized alerts in the application, demonstrating a high level of control and responsiveness.
Post-Conditions	Users receive timely and relevant notifications about potential threats or important information related to their location.
Special Requirements	Geolocation services, weather API integration, and user preferences.
Basic Flow	
Actor Action	**System Response**
1 The administrator opens the Localized Alert Management section.	2 The system displays tools for creating and sending alerts.

3	The administrator can: Define alert criteria, Create alert messages, Schedule alert delivery.	4	The system delivers alerts to relevant users via push notifications, email, or other channels.
Alternative Flow			
a	The system informs the administrator that they need more data to define the alert criteria effectively.	b	The system allows users to control their notification preferences and manage the types of alerts they receive.
c	The system alerts the administrator of any technical issues affecting alert delivery and provides troubleshooting steps.		

3.5.16 Review Reports

Table 20: Review Reports

Name	**Review Reports**		
Actors	Admin		
Summary	The administrator analyzes platform usage data and user feedback to gain insights and improve the platform.		
Pre-Conditions	The administrator is logged into the platform and has access to reporting dashboards.		
Post-Conditions	The administrator identifies trends, patterns, and areas for improvement based on data analysis.		
Special Requirements	Comprehensive reporting tools and integration with analytics.		
Basic Flow			
Actor Action		**System Response**	
1	The administrator selects specific reports from a dashboard.	2	The system displays visualizations and data analysis reports on various aspects: User activity, Content engagement, Disease management, FAQs**.**
3	The administrator analyzes the data to identify	4	The system logs the administrator's actions and

	trends, patterns, and areas for improvement.		updates the platform accordingly.
Alternative Flow			
a	The administrator investigates further to identify the underlying causes and potential implications.	b	The administrator prioritizes addressing these issues and implementing improvements.
c	The administrator takes immediate action to address the identified risks and protect user data.		

3.5.17 Manage FAQs

Table 21: Manage FAQs

Name	**Manage FAQs**		
Actors	Admin		
Summary	Admins can log in to the system and manage the FAQs by editing, deleting, or adding new entries.		
Pre-Conditions	Admin will manage the FAQs in the system		
Post-Conditions	FAQs are successfully managed, and the system reflects the changes.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	Admin navigates to the "Manage FAQs" section.	2	System displays a list of existing FAQs with options to edit, delete, or add new entries.
Alternative Flow			
a	None		

3.6 Database Design

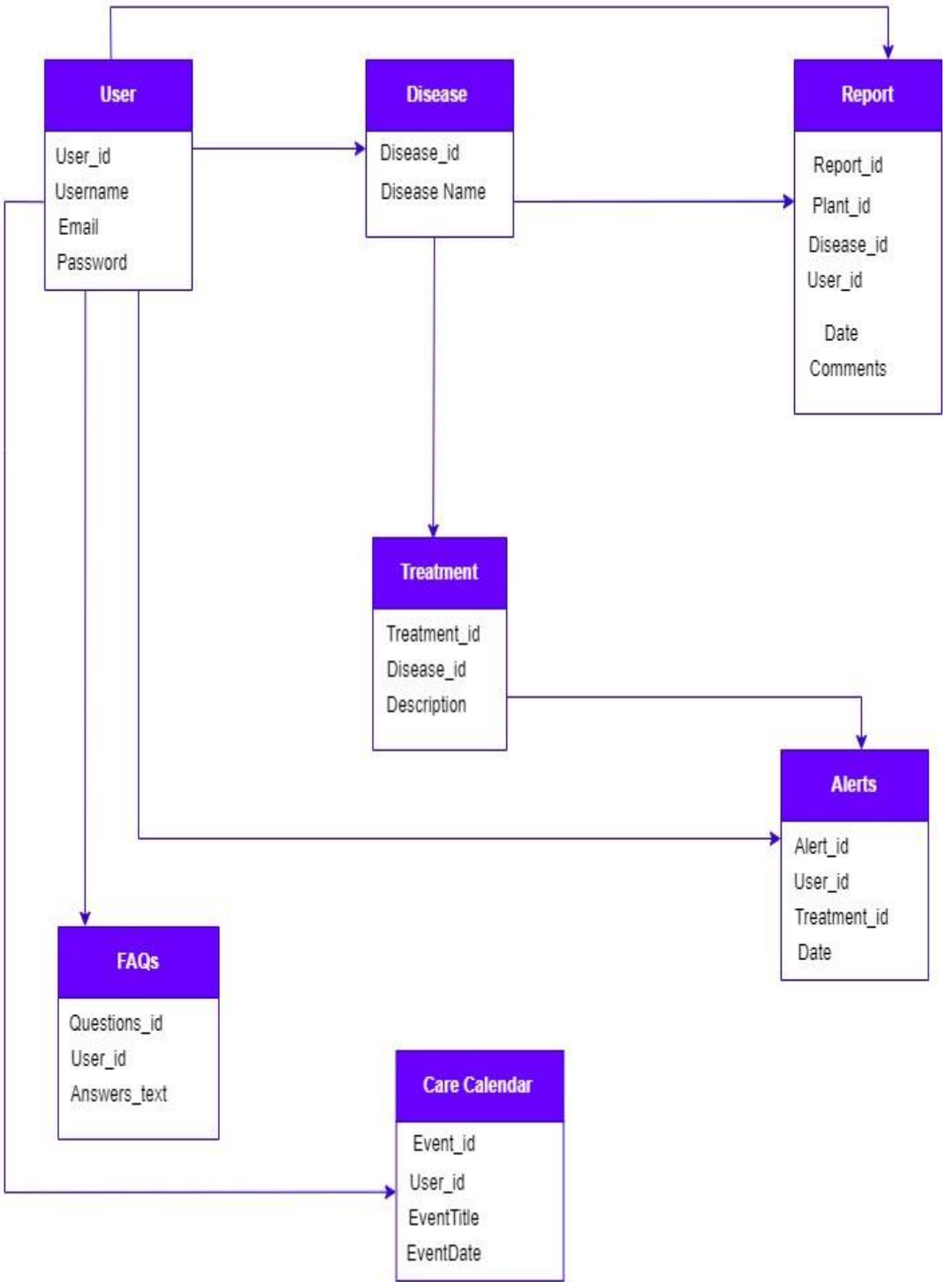


Figure 4: Database Design

3.7 Methodology Diagram

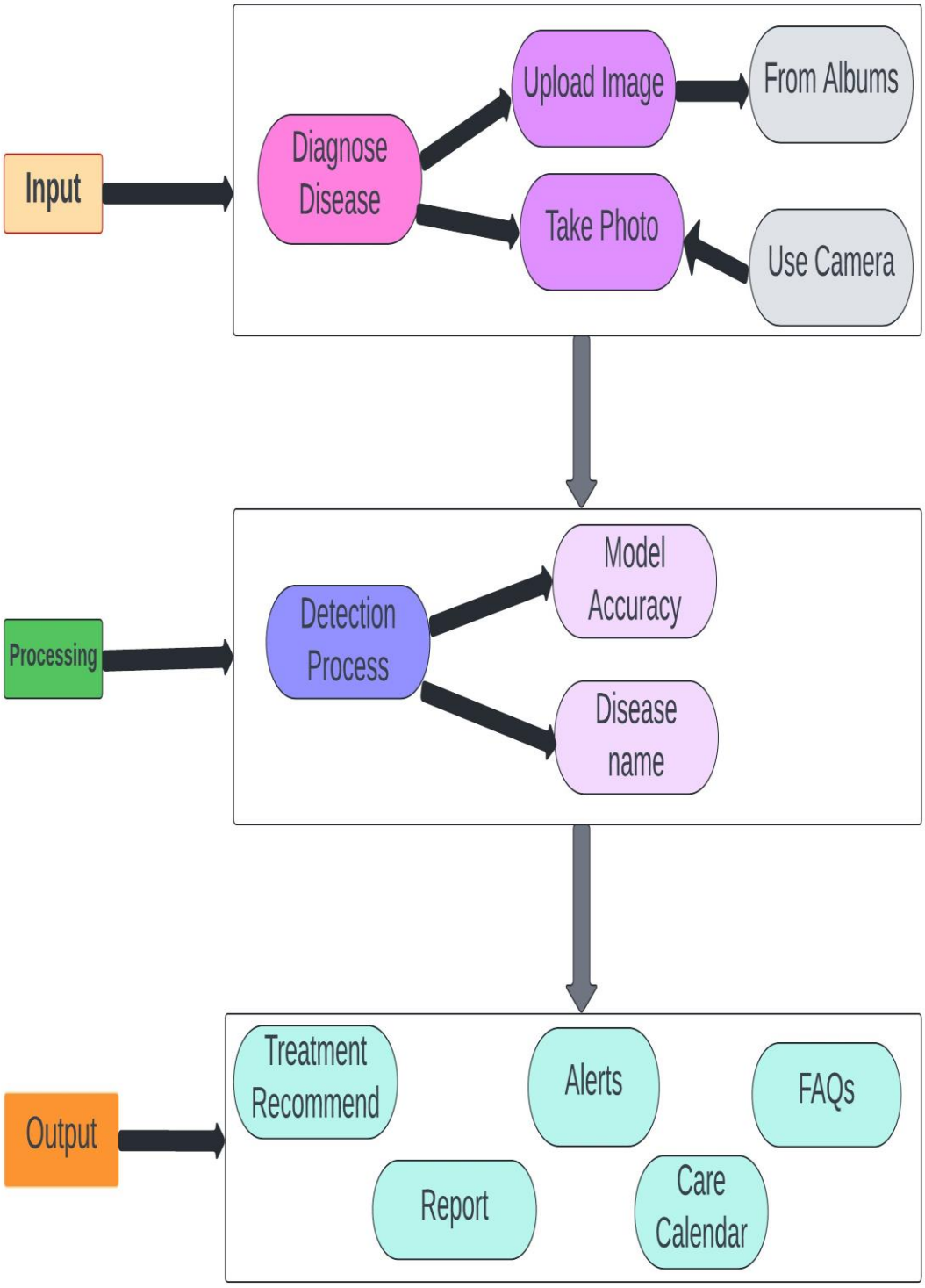


Figure 5: Methodology Diagram

3.8 Summary

In the realm of agricultural technology, the quest for efficient and accurate tomato disease detection has driven the development of sophisticated solutions through Requirement Engineering. This method systematically captures, analyzes, and defines the needs and specifications necessary for a technology tailored to detect diseases in tomatoes. Understanding these specific requirements is crucial for crafting solutions that effectively address the challenges farmers face in managing crop health. This includes identifying common tomato plant diseases and determining the technological capabilities needed for timely detection. This foundation sets the stage for delving into the detailed requirements essential for creating impactful solutions in tomato crop health management.

Chapter 4

Implementation and Test Cases

4.1 Endeavour

4.1.1 Tomato Care

Nurturing Crops with Innovative Farming Solutions

4.1.2 Team Members

Muhammad Salman Afaq (ID: 24775)

Usman Afaq (ID: 24779)

Rafaqat Ahmad (ID: 24784)

4.2 Flow Control

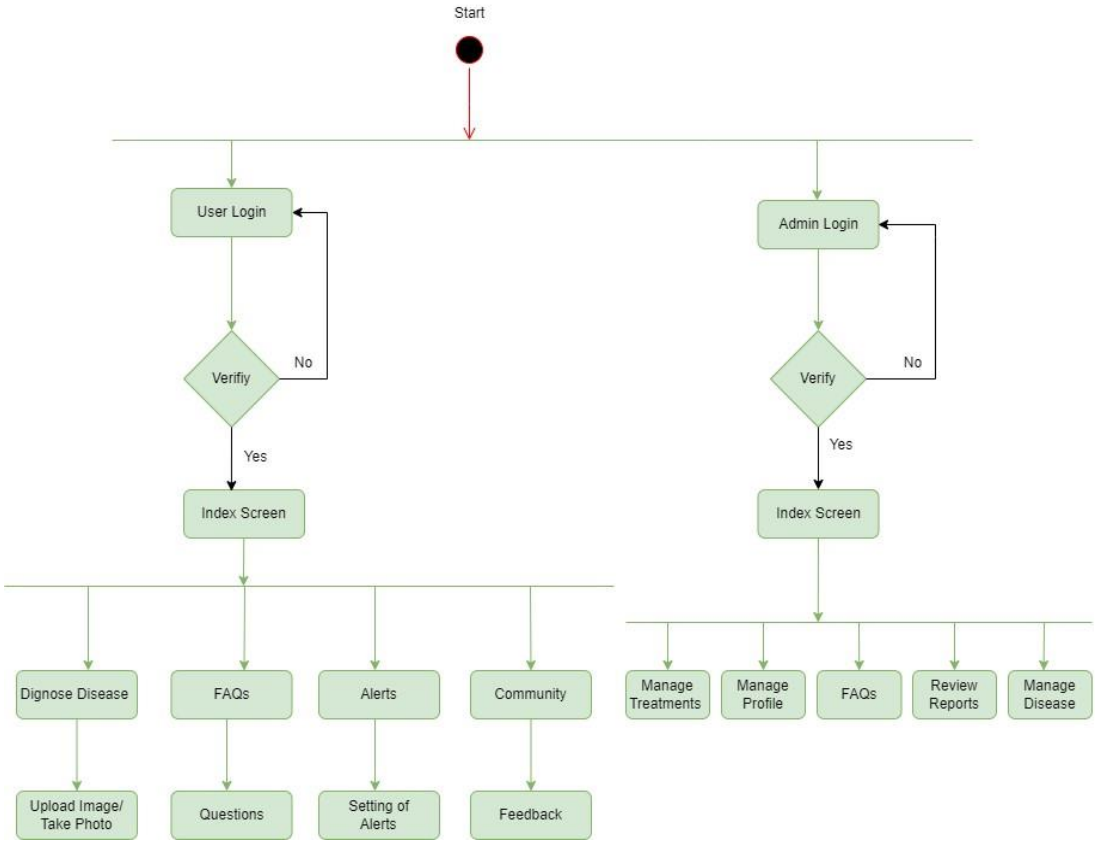


Figure 6: Flow Control

4.3 Components, Libraries, Web Services, and Stubs

4.3.1 Components

4.3.1.1 CNN Model

Description: Core for disease detection.

Functionality: Implements CNN for image classification.

Implementation: TensorFlow and Python.

4.3.1.2 Flask Web Application

Description: Web interface for CNN interaction.

Functionality: Processes image data.

Implementation: Flask in Python.

4.3.1.3 Flutter Application

Description: Mobile app for real-time monitoring.

Functionality: Integrates with Flask for interaction.

Implementation: Flutter SDK.

4.3.2 Libraries

4.3.2.1 TensorFlow

Description: Deep learning library.

Integration: Constructs and trains CNN.

4.3.2.2 OpenCV

Description: Image processing library.

Integration: Used for image pre-processing.

4.3.2.3 Flutter SDK

Description: Cross-platform app development kit.

Integration: Utilized for Flutter app.

4.3.3 Web Services

4.3.3.1 Flask API

Description: Interface for CNN-Flutter communication.

Implementation: Created using Flask.

4.3.3.2 Fire Base Database

Description: Stores disease data.

Integration: Utilizes FireBase Database.

4.3.4 Stubs

4.3.4.1 Testing Stubs

Description: Simulates external components for testing.

Usage: Validates component functionality.

4.4 IDE, Tools and Technologies

4.4.1 Statement

Optimized Technology Stack for Tomato Care Application

4.4.2 Integrated Development Environment (IDE)

Backend: Python (Specifically for Backend Development)

Flutter: Visual Studio Code

Collaborative Model Training: Google Colab

4.4.2.1 Tools

Document Sharing: GitHub

4.4.2.2 Technologies

Backend: Python (Specifically for Backend Development)

Framework: Flutter

Machine Learning: TensorFlow, OpenCV

Database: Firebase

Language: Dart

4.5 Deployment Environment for Tomato Care Application

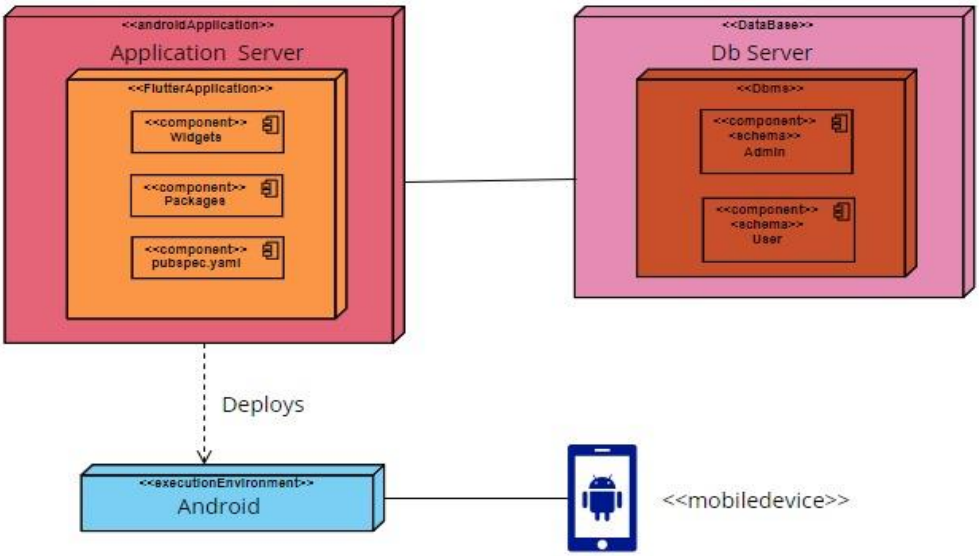


Figure 7: Deployment Environment

4.6 Test Cases and Descriptions

4.6.1 Test Data

Table 22: TD-1

Test Data	TD-1
Form	Login
Stakeholder	User
Field	Email
Technique	Equivalence Partitioning
Valid	• Correct length (user@example.com) • Includes numeric characters (user123@example.com). • Includes country code (user@example.co.uk)
Invalid	• Does not contain '@' (userexample.com).Include characters

	• Includes characters (user@exampl*e.com).Not End with ‘.’ • Includes special characters (user@ex!ample.com). • Does not end with '.' (user@examplecom).

Table 23: TD-2

Test Data	TD-2
Form	Login
Stakeholder	User
Field	Password
Technique	Equivalence Partitioning
Valid	• Password length should be >=6 • Includes two special characters • Includes uppercase and lowercase character • Includes one numeric character
Invalid	• Password length < 6 • No uppercase and lowercase characters • No special character • No numeric character

Table 24: TD-3

Test Data	TD-3
Form	Login
Stakeholder	Admin
Field	Email
Technique	Equivalence Partitioning
Valid	• Correct length (user@example.com) • Includes numeric characters (user123@example.com). • Includes country code (user@example.co.uk)

Invalid	• Does not contain '@' (userexample.com).Include characters • Includes characters (user@exampl*e.com).Not End with '.' • Includes special characters (user@ex!ample.com). • Does not end with '.' (user@examplecom).

Table 25: TD-4

Test Data	TD-4
Form	Login
Stakeholder	Admin
Field	Password
Technique	Equivalence Partitioning
Valid	• Password length should be >=6 • Includes two special characters • Includes uppercase and lowercase character • Includes one numeric character
Invalid	• Password length < 6 • No uppercase and lowercase characters • No special character • No numeric character

Table 26: TD-5

Test Data	TD-5
Form	Identify Disease
Stakeholder	User
Field	Capture Photo
Technique	Equivalence Partitioning
Valid	• Requires a mobile device with a camera

Invalid	• No phone other than smartphone • No smartphone without camera

Table 27: TD-6

Test Data	TD-6
Form	FAQs
Stakeholder	Admin
Field	Manage FAQs
Technique	Equivalence Partitioning
Valid	• Includes User questions • Relevant Questions
Invalid	• Irrelevant Questions • User must have logged in

Table 28: TD-7

Test Data	TD-7
Form	Report
Stakeholder	User
Field	Generate Report
Technique	Equivalence Partitioning
Valid	• Includes the Disease detection process
Invalid	• No direct process

4.6.2 Test Cases

Table 29: TC-1

ID	TC-1
Form	Login
Stakeholder	User
Field	Email
Input	Rafaqat@gmail.com
Partition Tested	• Correct length (user@example.com) • Includes numeric characters (user123@example.com). • Includes country code (user@example.co.uk)
Expected Output	• OK
Observed Output	• No error

Table 30: TC-2

ID	TC-2
Form	Login
Stakeholder	User
Field	Password
Input	admin123
Partition Tested	• Password length should be >=6 • Includes two special characters • Includes uppercase and lowercase character • Includes one numeric character
Expected Output	• Password OK

Observed Output	• No error

Table 31: TC-3

ID	TC-3
Form	Login
Stakeholder	Admin
Field	Email
Input	Equivalence Partitioning
Partition Tested	• Correct length (user@example.com) • Includes numeric characters (user123@example.com). • Includes country code (user@example.co.uk)
Expected Output	• OK
Observed Output	• No Error

Table 32: TC-4

ID	TC-4
Form	Login
Stakeholder	Admin
Field	Password
Input	Equivalence Partitioning
Partition Tested	• Password length should be >=6 • Includes two special characters • Includes uppercase and lowercase character • Includes one numeric character

Expected Output	• OK
Observed Output	• No error occured

Table 33: TC-5

ID	TC-5
Form	Identify Disease
Stakeholder	User
Field	Capture Photo
Input	Take Photo
Partition Tested	• Requires a mobile device with a camera
Expected Output	• Disease Detection
Observed Output	• No error

Table 34: TC-6

ID	TC-6
Form	FAQs
Stakeholder	Admin
Field	Manage FAQs
Input	Management
Partition Tested	• Includes User questions • Relevant Questions

Expected Output	• FAQs manage
Observed Output	• All Ok

Table 35: TC-7

ID	TC-7
Form	Report
Stakeholder	User
Field	Generate Report
Input	Check Report
Partition Tested	• Includes the Disease detection process
Expected Output	• Report of Disease
Observed Output	• Accuracy up-down

4.7 Summary

The implementation of tomato disease detection began with gathering a diverse dataset of tomato images, followed by training a CNN model through iterative optimization to improve its accuracy in disease identification. After evaluating its performance on separate test images and undergoing validation and refinement processes to address any inaccuracies, the trained model is ready to give a good accuracy. We gathered approximately 24,000 leaf images to train a model to detect leaf diseases more effectively. After three rounds of training and adjustments, the model reached an accuracy of about 90% in spotting diseases.We used a CNN architecture which is ResNet 18 to make the model good at spotting diseases. This could greatly help farmers keep plants healthy and safe.

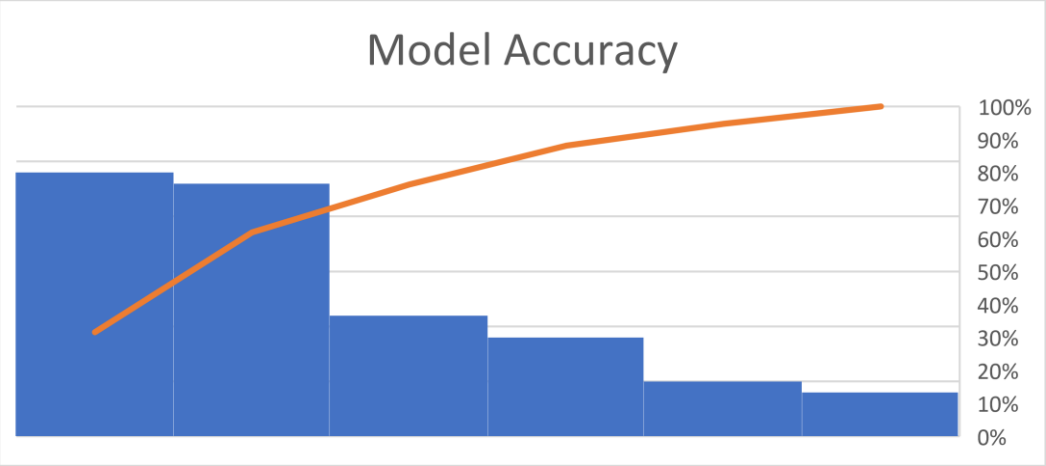
Chapter 5

Experimental Results and Analysis

5.1 Introduction

In our pursuit of advancing agricultural technology, we have assembled a comprehensive dataset comprising 24 thousand meticulously curated leaf images. This dataset serves as the cornerstone for training our model, a process that underwent three rigorous iterations. Our objective was clear: to refine the model's ability to accurately detect diseases affecting tomato leaves. After the fourth attempt, our dedication yielded remarkable results—an impressive accuracy rate of approximately 90%. To further optimize performance, we meticulously fine-tuned the model's parameters through epoch runs, carefully balancing computational efficiency and training effectiveness. Leveraging the ResNet 18 architecture renowned for its depth and efficiency, we equipped our model with the capability to discern and classify leaf diseases with unparalleled precision and reliability. This culmination of meticulous experimentation and optimization has laid the groundwork for a robust and dependable tool poised to revolutionize disease diagnosis and management in agriculture. Our solution promises invaluable support to farmers and researchers, safeguarding crop health and ensuring global food security.

5.2 Model Accuracy



The graph shows that the accuracy of our model is achieving higher with the changing of datasets. It depends on the dataset on which we have trained our model. It achieves a final accuracy of 90% and still we are aiming to produce more further.

5.3 Limitations of our model

Although our CNN model, based on ResNet18, effectively detects diseases in tomato leaves, it has some limitations. It tends to identify any green-colored object, leading to false positives for objects with similar hues. This reduces its ability to specifically identify tomato leaves. Additionally, it may struggle to differentiate between different types of leaves if they appear similar. The model's reliance on color features can result in inaccuracies in varying lighting conditions or with different shades of green in tomato leaves. Moreover, it may not perform well with backgrounds or textures resembling tomato leaves, limiting its generalization capabilities. Despite these challenges, the model achieves good accuracy in detecting tomato leaf diseases.

Chapter 6

Conclusion and Future Directions

6.1 Conclusion

The tomato disease detection project aims to streamline the process of identifying and managing diseases affecting tomato plants to improve crop yields and agricultural sustainability. The proposed solution involves a user-friendly mobile app interface where farmers can capture photos of their tomato plants using smartphones. These images are then analyzed using image processing and machine learning algorithms to identify diseases and assess their severity. Based on the analysis results, the app provides farmers with tailored treatment recommendations for the detected diseases.

6.2 Future Directions

Future directions in tomato disease detection involve integrating advanced technologies like tailored machine learning, remote sensing, and IoT for early detection and comprehensive understanding of disease dynamics. Implementing blockchain for traceability and mobile computing for on-device processing can enhance scalability and real-time performance. Engaging stakeholders through citizen science and interdisciplinary collaboration will be crucial for sustainable agriculture and food security.

References

[1] Durmuş, H., Güneş, E. O., & Kırcı, M. (2017, August). Disease detection on the leaves of the tomato plants by using deep learning. In 2017 6th International conference on agro-geoinformatics (pp. 1-5)

[2] Ashqar, B. A., & Abu-Naser, S. S. (2018). Image-based tomato leaves diseases detection using deep learning.

[3] Tm, P., Pranathi, A., SaiAshritha, K., Chittaragi, N. B., & Koolagudi, S. G. (2018, August). Tomato leaf disease detection using convolutional neural networks. In 2018 eleventh international conference on contemporary computing (IC3) (pp. 1-5)

[4] Chowdhury, M. E., Rahman, T., Khandakar, A., Ibtehaz, N., Khan, A. U., Khan, M. S., ... & Ali, S. H. M. (2021). Tomato leaf diseases detection using deep learning technique. Technology in Agriculture, 453.

[5] Harakannanavar, S. S., Rudagi, J. M., Puranikmath, V. I., Siddiqua, A., & Pramodhini, R. (2022). Plant leaf disease detection using computer vision and machine learning algorithms. Global Transitions Proceedings, 305-310.

[6] Gadade, H. D., & Kirange, D. K. (2021, April). Machine learning based identification of tomato leaf diseases at various stages of development. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 814-819). IEEE.

[7] Ullah, Z., Alsubaie, N., Jamjoom, M., Alajmani, S. H., & Saleem, F. (2023). EffiMob-Net: A deep learning-based hybrid model for detection and identification of tomato diseases using leaf images. *Agriculture*, *13*(3), 737.

[8] Paul, S. G., Biswas, A. A., Saha, A., Zulfiker, M. S., Ritu, N. A., Zahan, I., ... & Islam, M. A. (2023). A real-time application-based convolutional neural network approach for tomato leaf disease classification. *Array*, *19*, 100313.

[9] Parvez, S., Uddin, M. A., Islam, M. M., Bharman, P., & Talukder, M. A. (2023). Tomato leaf disease detection using convolutional neural network.

[10] Sakkarvarthi, G., Sathianesan, G. W., Murugan, V. S., Reddy, A. J., Jayagopal, P., & Elsisi, M. (2022). Detection and classification of tomato crop disease using convolutional neural network. *Electronics*, *11*(21), 3618.