

Tomato Care



By:

Muhammad Salman Afaq

24775

Usman Afaq

24779

Rafaqat Ahmad

24784

Supervised by:

Mr. Muhammad Usman Karim

Faculty of Computing

Riphaah 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: [2nd May 2024]

Final Approval

This is to certify that we have read the report submitted by *Muhammad Salman Afaq (24775), Usman Afaq (24779), Rafiqat 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

| | |
|---|----|
| Abstract | 1 |
| Chapter 1 | 2 |
| Introduction..... | 2 |
| 1.1 Goals And Objectives..... | 2 |
| 1.1.1 Goals | 2 |
| 1.2.1 Objectives | 3 |
| 1.2 Scope of the Project | 3 |
| Chapter 2 | 4 |
| Literature Review | 4 |
| 2.1 Introduction..... | 4 |
| 2.2 Background and Problem Elaboration..... | 4 |
| 2.3 Detailed Literature Review | 5 |
| 2.4 Literature Review Summary Table | 6 |
| 2.4.1 Market Survey | 7 |
| 2.5 Research gap | 7 |
| 2.6 Problem Statement | 8 |
| Chapter 3 | 9 |
| Requirements and Design | 9 |
| 3.1 Requirements | 9 |
| 3.1.1 Functional Requirements | 9 |
| 3.1.2 Non-Functional Requirements | 10 |
| 3.1.3 Hardware and Software Requirements..... | 10 |
| 3.2 Proposed Methodology..... | 11 |
| 3.3 System Architecture | 12 |
| 3.4 Use Cases..... | 13 |
| 3.4.1 Farmer | 13 |
| 3.4.2 Admin | 14 |
| 3.5 Fully Dressed Use Cases | 15 |
| 3.5.1 Sign Up..... | 15 |
| 3.5.2 Login | 15 |
| 3.5.3 View Profile..... | 16 |
| 3.5.4 Edit Profile | 17 |

| | |
|--|----|
| 3.5.5 Upload Images | 17 |
| 3.5.6 Access Calendar | 18 |
| 3.5.7 Provide Feedback | 19 |
| 3.5.8 FAQs..... | 20 |
| 3.5.9 View Reports..... | 20 |
| 3.5.10 Login | 21 |
| 3.5.11 Manage Diseases | 22 |
| 3.5.12 Manage Treatments | 23 |
| 3.5.13 Manage Users | 24 |
| 3.5.14 Manage Localized Alerts..... | 25 |
| 3.5.15 Review Reports..... | 26 |
| 3.5.16 Manage FAQs..... | 27 |
| 3.6 Database Design | 28 |
| 3.7 Methodology Diagram..... | 29 |
| Chapter 4 | 30 |
| Implementation and Test Cases | 30 |
| 4.1 Test Cases and Descriptions | 30 |
| 4.1.1 Test Cases | 30 |
| Chapter 5 | 34 |
| Experimental Results and Analysis | 34 |
| 5.1 Model Accuracy | 34 |
| 5.2 Limitations of our model | 35 |
| Chapter 6 | 36 |
| Conclusion and Future Directions | 36 |
| 6.1 Conclusion | 36 |
| 6.2 Future Directions..... | 36 |
| References..... | 37 |
| Appendix..... | 38 |
| Appendix A: Guidelines | 38 |
| Appendix B: Heading of Sample Appendix B | 38 |
| Formatting Guidelines..... | 38 |
| Heading 1..... | 39 |
| Heading 2..... | 39 |

| | |
|----------------------------------|----|
| Heading 3..... | 39 |
| Tables and Figures..... | 39 |
| Equations..... | 40 |
| Header/Footer..... | 40 |
| Other Formatting Guidelines..... | 41 |
| References..... | 41 |
| Sample List..... | 41 |

List of Tables

| | |
|---|----|
| Table 1: Literature Review | 6 |
| Table 2: Market Survey | 7 |
| Table 3: This is Sample table caption | 40 |
| Table 4: This is Sample table caption | 40 |

List of figures

| | |
|--------------------------------------|----|
| Figure 1: Architecture Diagram | 12 |
| Figure 2: Farmer Use Case | 13 |
| Figure 3: Admin Use Case | 14 |
| Figure 4: Database Design | 28 |
| Figure 5: Methodology Diagram | 29 |
| Figure 6: List of Styles | 40 |
| Figure 7: IEEE Reference style | 41 |

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 85%. 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.

1.2.1 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.

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.

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, Requirement Engineering serves as the foundation for developing strong and user-friendly Tomato Disease Detection systems.

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

| 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 access a personalized calendar for plant care. |
| 1.7 | User can give feedback. |
| 1.8 | User can also interact with the FAQs. It will provide relevant answers. |
| 1.9 | User can also view the reports. |

3.1.1.2 Admin

| 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 or near-real-time processing for disease detection to ensure prompt results for farmers.

Response time for uploading images and receiving results should be within an acceptable range.

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

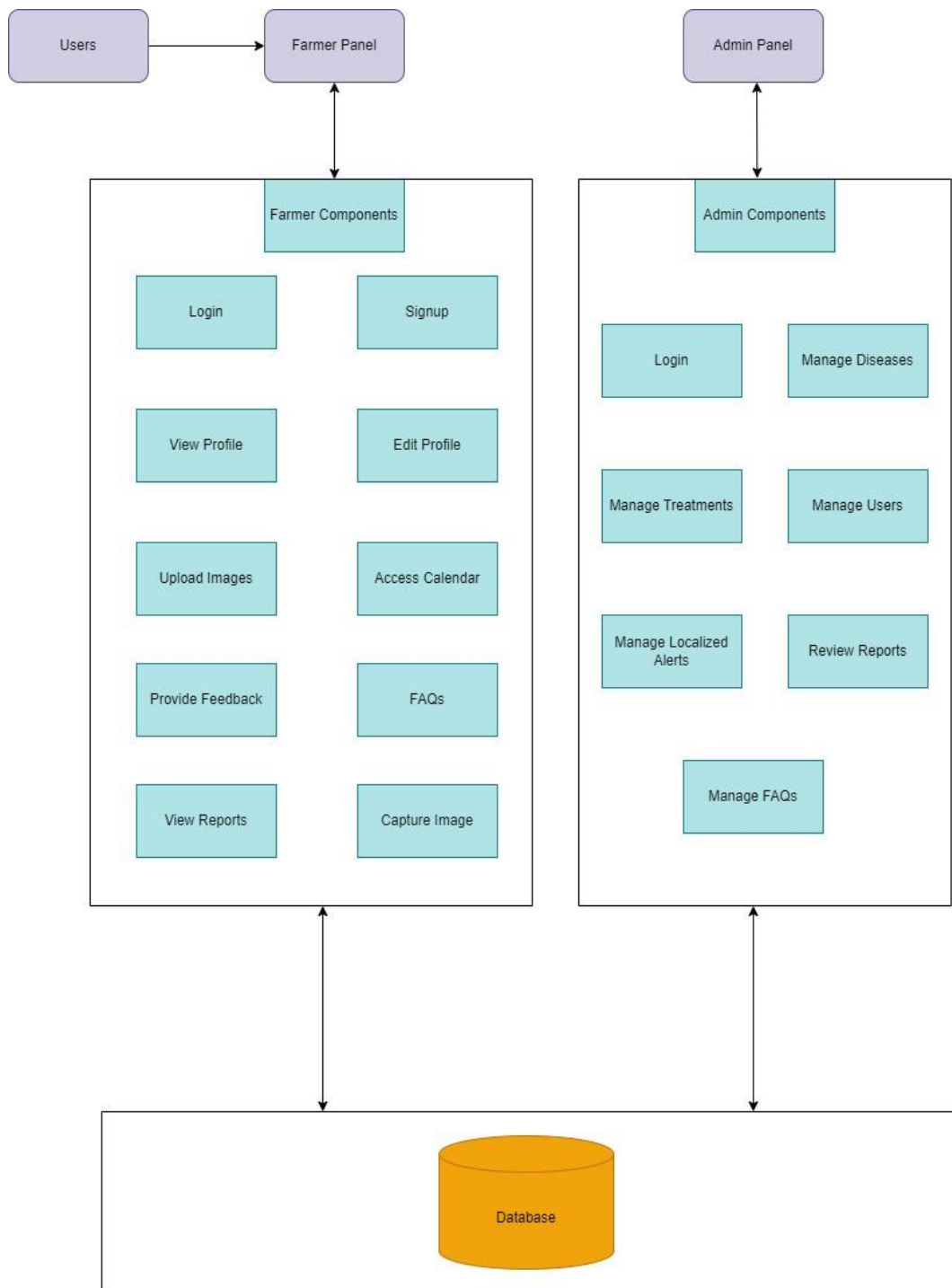


Figure 1: Architecture Diagram

3.4 Use Cases

3.4.1 Farmer

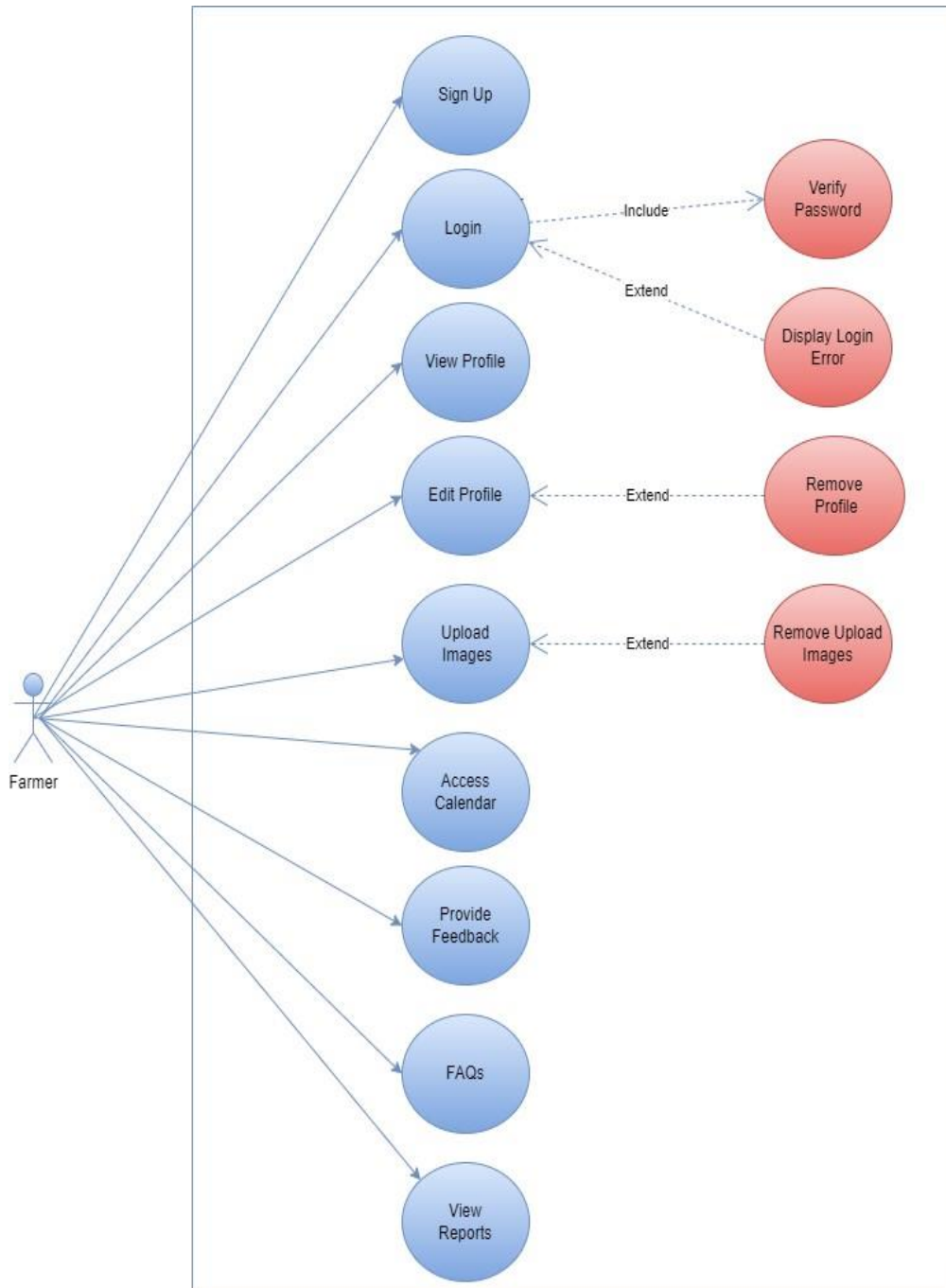


Figure 2: Farmer Use Case

3.4.2 Admin

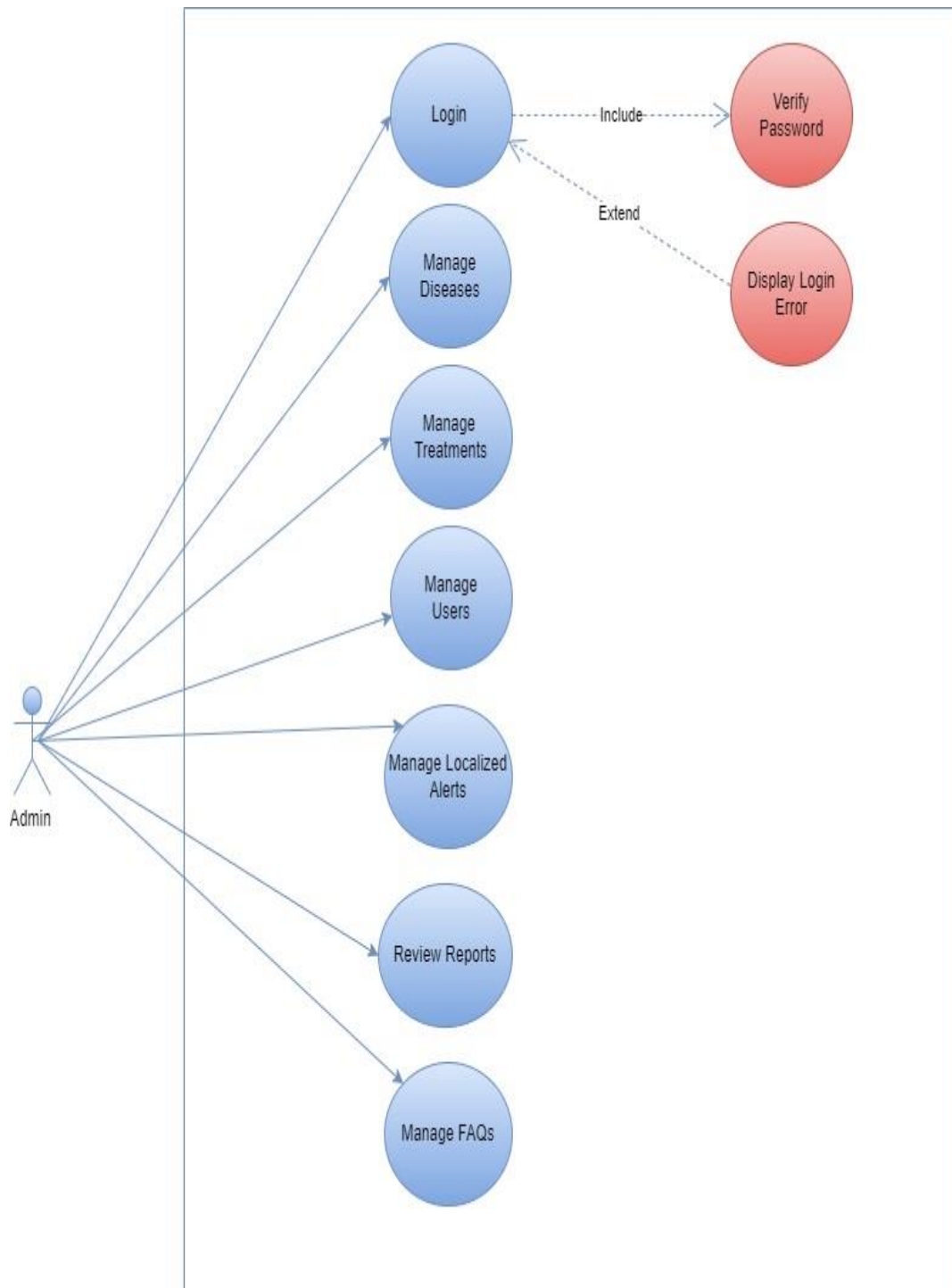


Figure 3: Admin Use Case

3.5 Fully Dressed Use Cases

3.5.1 Sign Up

| | | | |
|----------------------|---|--|--|
| 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 platform. | |
| 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 | | | |
| 1 | If the entered email is already registered, the system displays an error message and prompts the farmer to use a different email. | 2 | If the password doesn't meet the complexity requirements, the system prompts the farmer to choose a stronger password. |

3.5.2 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 | | | |
| 1 | If the entered credentials are incorrect, the system displays an error message and allows the farmer to attempt login again. | 2 | If the farmer forgets the password, there is a "Forgot Password" option that allows them to reset the password |

3.5.3 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. |
| Alternative Flow | | | |
| 1 | None | | |

3.5.4 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. |
| Alternative Flow | | | |
| 1 | If the farmer attempts to submit invalid information, the system displays an error message and prompts the farmer to correct the information. | | |

3.5.5 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 access camera. | |
| 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 | | | |
| 1 | If the uploaded images are of an unsupported format or size, the system prompts the farmer to upload valid images. | 2 | If there is an issue with the image processing, the system notifies the farmer and recommends re-uploading the images. |

3.5.6 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. |
| Alternative Flow | | | |
| 1 | If there are no personalized recommendations for the farmer, the system displays a message suggesting that the farmer has no specific activities scheduled and encourages them to check back later. | | |

3.5.7 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. |
| Alternative Flow | | | |
| 1 | If the feedback form submission fails due to technical issues, the system displays a message informing the farmer about the problem | | |

| | |
|--|--------------------------------------|
| | and advises them to try again later. |
|--|--------------------------------------|

3.5.8 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. |
| Alternative Flow | | | |
| 1 | User cannot find the answer to their question in the FAQs. | | |

3.5.9 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 | | | |
| 1 | The system prompts the user to provide additional information or filter parameters to refine the report. | 2 | The user provides the requested information or modifies filter settings. |
| 3 | The system updates the report with the refined data. | | |

3.5.10 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 | | | |
| 1 | The system displays an error message and prompts the user to re-enter their credentials. | 2 | The system informs the user that their account is locked and provides instructions for unlocking it. |
| 3 | The system sends a verification code to the administrator's registered phone number or email address for additional security. | | |

3.5.11 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 | | | |
| 1 | The system prompts the administrator to complete all required fields before submitting the new disease information. | 2 | 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. |
| 3 | The system validates the format and accuracy of the imported data before adding it to the platform. | | |

3.5.12 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 | | | |
| 1 | The system requires the administrator to provide clear warnings and disclaimers about potential risks associated with the treatment. | 2 | The system allows the administrator to edit and update the treatment information to ensure accuracy and currency. |
| 3 | The system prompts the administrator to provide more reliable references and evidence to support the treatment information. | | |

3.5.13 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 | | | |
| 1 | The system prompts the user to complete all required | 2 | The system may request additional documentation or |

| | | | |
|---|---|--|---|
| | information before submitting the registration request. | | proof of identity to verify the user's information. |
| 3 | The system ensures that user privacy is protected and sensitive information is not publicly accessible. | | |

3.5.14 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 | | | |
| 1 | The system informs the administrator that they need more data to define the alert criteria effectively. | 2 | The system allows users to control their notification preferences and manage the types of alerts they receive. |
| 3 | The system alerts the administrator of any technical issues affecting alert delivery and provides troubleshooting steps. | | |

3.5.15 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, Chatbot interactions. |
| 3 | The administrator analyzes the data to identify trends, patterns, and areas for improvement. | 4 | The system logs the administrator's actions and updates the platform accordingly. |
| Alternative Flow | | | |
| 1 | The administrator investigates further to identify the underlying causes and potential implications. | 2 | The administrator prioritizes addressing these issues and implementing improvements. |
| 3 | The administrator takes immediate action to address the identified risks and protect user data. | | |

3.5.16 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 | | | |
| 1 | None | | |

3.6 Database Design

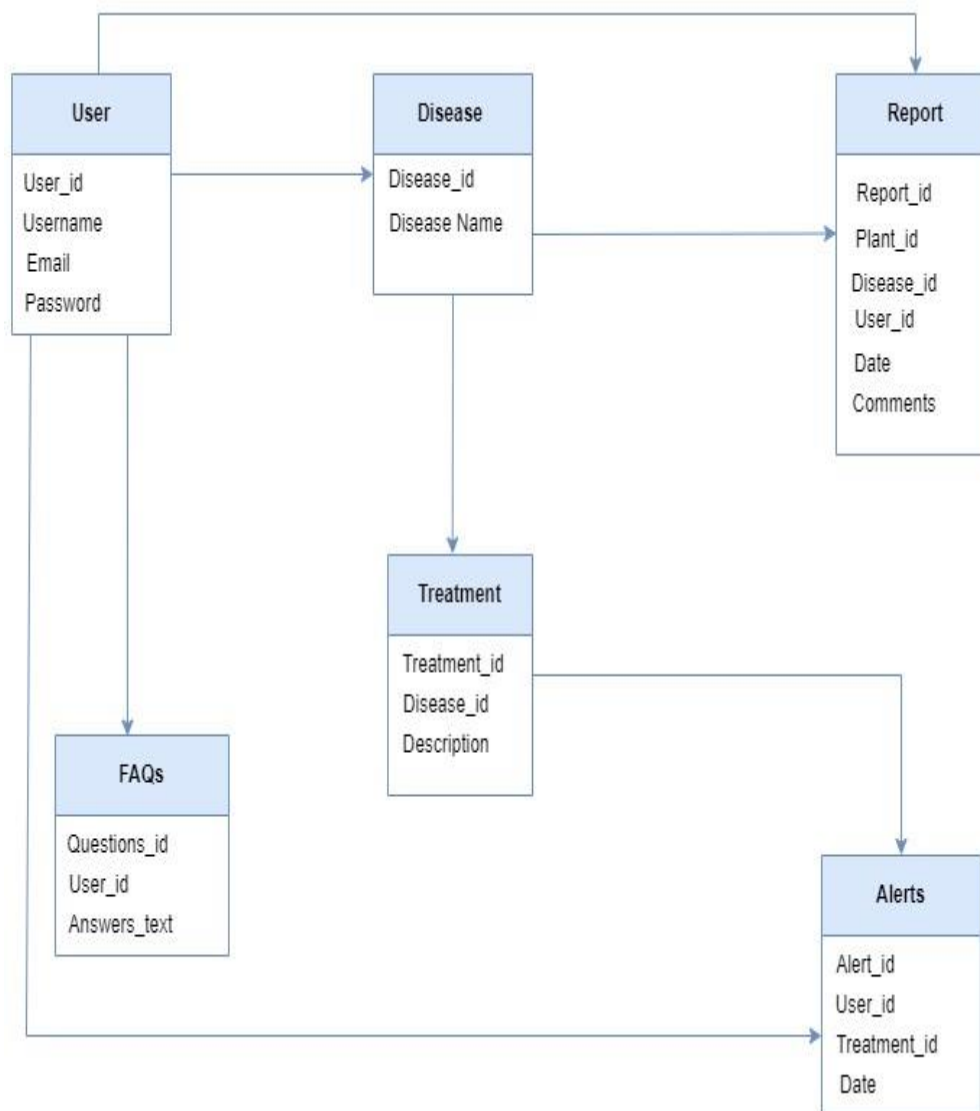


Figure 4: Database Design

3.7 Methodology Diagram

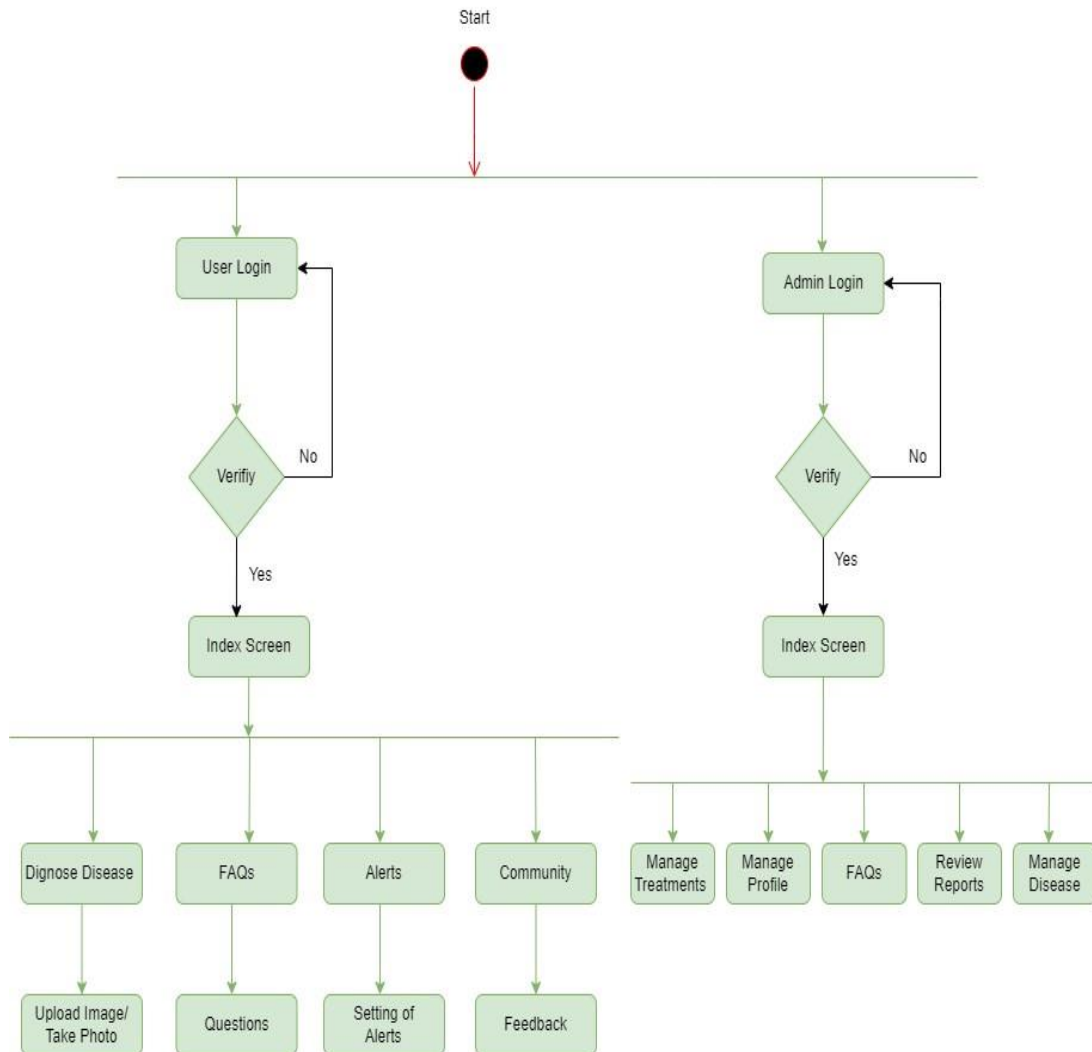


Figure 5: Methodology Diagram

Chapter 4

Implementation and Test Cases

4.1 Test Cases and Descriptions

4.1.1 Test Cases

| | |
|--------------------|--|
| Test Case | TC-1 |
| Form | Login |
| Stakeholder | User |
| Field | Email |
| Technique | Equivalence Partitioning |
| Valid | <ul style="list-style-type: none">• Correct length (user@example.com)• Includes numeric characters (user123@example.com).• Includes country code (user@example.co.uk) |
| Invalid | <ul style="list-style-type: none">• 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). |

| | |
|--------------------|---|
| Test Case | TC-2 |
| Form | Login |
| Stakeholder | User |
| Field | Password |
| Technique | Equivalence Partitioning |
| Valid | <ul style="list-style-type: none">• Password length should be ≥ 6 |

| | |
|----------------|---|
| | <ul style="list-style-type: none"> • Includes two special characters • Includes uppercase and lowercase character • Includes one numeric character |
| Invalid | <ul style="list-style-type: none"> • Password length < 6 • No uppercase and lowercase characters • No special character • No numeric character |

| | |
|--------------------|---|
| Test Case | TC-3 |
| Form | Login |
| Stakeholder | Admin |
| Field | Email |
| Technique | Equivalence Partitioning |
| Valid | <ul style="list-style-type: none"> • Correct length (user@example.com) • Includes numeric characters (user123@example.com). • Includes country code (user@example.co.uk) |
| Invalid | <ul style="list-style-type: none"> • 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). |

| | |
|--------------------|--|
| Test Case | TC-4 |
| Form | Login |
| Stakeholder | Admin |
| Field | Password |
| Technique | Equivalence Partitioning |
| Valid | <ul style="list-style-type: none"> • Password length should be >=6 |

| | |
|----------------|---|
| | <ul style="list-style-type: none"> • Includes two special characters • Includes uppercase and lowercase character • Includes one numeric character |
| Invalid | <ul style="list-style-type: none"> • Password length < 6 • No uppercase and lowercase characters • No special character • No numeric character |

| | |
|--------------------|--|
| Test Case | TC-5 |
| Form | Identify Disease |
| Stakeholder | User |
| Field | Capture Photo |
| Technique | Equivalence Partitioning |
| Valid | <ul style="list-style-type: none"> • Requires a mobile device with a camera |
| Invalid | <ul style="list-style-type: none"> • No phone other than smartphone • No smartphone without camera |

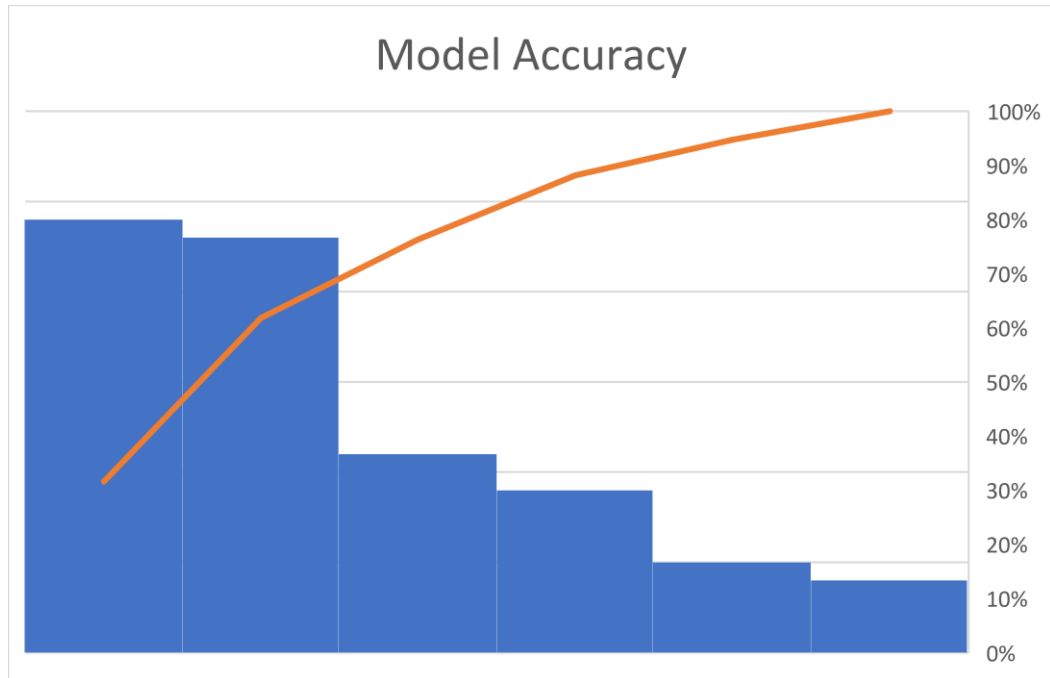
| | |
|--------------------|--|
| Test Case | TC-6 |
| Form | FAQs |
| Stakeholder | Admin |
| Field | Manage FAQs |
| Technique | Equivalence Partitioning |
| Valid | <ul style="list-style-type: none"> • Includes User questions • Relevant Questions |
| Invalid | <ul style="list-style-type: none"> • Irrelevant Questions • User must have logged in |

| | |
|--------------------|--|
| Test Case | TC-7 |
| Form | Report |
| Stakeholder | User |
| Field | Generate Report |
| Technique | Equivalence Partitioning |
| Valid | <ul style="list-style-type: none"> • Includes the Disease detection process |
| Invalid | <ul style="list-style-type: none"> • No direct process |

Chapter 5

Experimental Results and Analysis

5.1 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 85% and still we are aiming to produce more further.

5.2 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

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

- [1] Durmuş, H., Güneş, E. O., & Kırıcı, 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., & Elsis, M. (2022). Detection and classification of tomato crop disease using convolutional neural network. *Electronics*, 11(21), 3618.

Appendix

Appendix A: Guidelines

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

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

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

Appendix B: Heading of Sample Appendix B

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

```
Void SampleFunction(){  
    Print "Hello World.";  
}
```

Formatting Guidelines

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

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

```
int x = 10; // Writing important code
```
- **Table Header Row** – Style for the header row of table
- **Table Grid** – the style for the data rows in the tables
- **Reference** – The style for references
- **Bullets** – The style for the bullet lists

- **Numbered List**– Style for numbered lists

All Heading styles with different level numbers are listed below.

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Heading 7

Heading 8

Heading 9

Tables and Figures

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

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

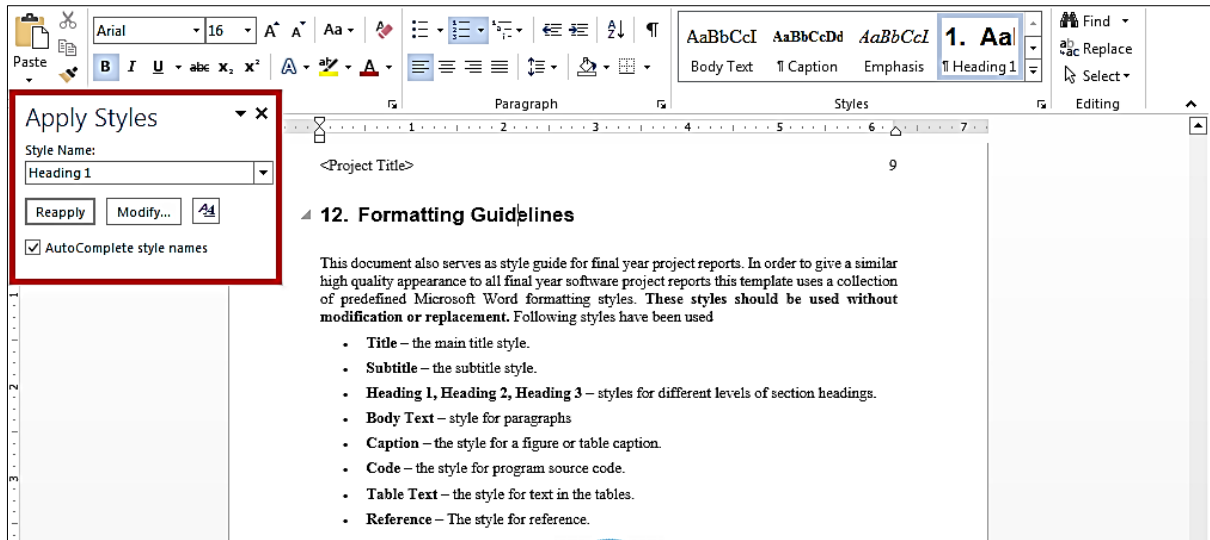


Figure 6: List of Styles

This is brief description of above figure.

Table 3: This is Sample table caption

This is brief description of following Table.

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

Table 4: This is Sample table caption

This is brief description of following Table.

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

Equations

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

$$\sum_{\forall v \in V(G)} \deg(v) = 2|E(G)| \quad (1)$$

Header/Footer

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

Other Formatting Guidelines

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

References

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

- [1] B. Klaus and P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 1986.
 - [2] L. Stein, “Random patterns,” in *Computers and You*, J. S. Brake, Ed. New York: Wiley, 1994, pp. 55-70.
 - [3] R. L. Myer, “Parametric oscillators and nonlinear materials,” in *Nonlinear Optics*, vol. 4, P. G. Harper and B. S. Wherret, Eds. San Francisco, CA: Academic, 1977, pp. 47-160.
 - [4] M. Abramowitz and I. A. Stegun, Eds., *Handbook of Mathematical Functions* (Applied Mathematics Series 55). Washington, DC: NBS, 1964, pp. 32-33.
 - [5] E. F. Moore, “Gedanken-experiments on sequential machines,” in *Automata Studies* (Ann. of Mathematical Studies, no. 1), C. E. Shannon and J. McCarthy, Eds. Princeton, NJ: Princeton Univ. Press, 1965, pp. 129-153.
 - [6] Westinghouse Electric Corporation (Staff of Technology and Science, Aerospace Div.), *Integrated Electronic Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1970.
 - [7] M. Gorkii, “Optimal design,” *Dokl. Akad. Nauk SSSR*, vol. 12, pp. 111-122, 1961 (Transl.: in L. Pontryagin, Ed., *The Mathematical Theory of Optimal Processes*. New York: Interscience, 1962, ch. 2, sec. 3, pp. 127-135).
 - [8] G. O. Young, “Synthetic structure of industrial plastics,” in *Plastics*, vol. 3, *Polymers of Hexadromicon*, J. Peters, Ed., 2nd ed. New York: McGraw-Hill, 1964, pp. 15-64.

Figure 7: IEEE Reference style

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

Sample List

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

- P.M. Morse and H. Feshback, *Methods of Theoretical Physics*. New York: McGraw Hill, 1953. **//Format for Book**
- S.K. Kenue and J.F. Greenleaf, “Limited angle multifrequency diffraction tomography,” *IEEE Trans. Sonics Ultrason.*, vol. SU-29, no. 6, pp. 213-217, July 1982. **//Format for Journal Article**
- B. Tsikos, “Segmentation of 3-D scenes using multi-modal interaction between machine vision and programmable mechanical scene manipulation,” Ph.D. dissertation, Univ. of Pennsylvania, BCE Dept., Philadelphia, 1987. [Add if applicable: University Microfilms, Inc., University of Michigan, Ann Arbor, Michigan.] **//Format for Dissertation or thesis**

- R. Finkel, R. Taylor, R. Bolles, R. Paul, and J. Feldman, "An overview of AL, programming system for automation," in *Proc. Fourth Int. Joint Conf Artif. Intell.*, pp. 758-765, Sept. 3-7, 1975. **//Format for Proceedings paper**
- "Technology threatens to shatter the world of college textbooks, *The Wall Street Journal*, vol 91, pp. A1, A8, June 1, 1993. **//Format for Newspaper article**
- R. Cox and J. S. Turner, "Project Zeus: design of a broadband network and its application on a university campus," Washington Univ., Dept. of Comp. Sci., Technical Report WUCS-91-45, July 30, 1991. **//Format for Technical Report**
- M. Janzen, *Instant Access Accounting*. Computer software. Nexus Software, Inc IBM-PC, 1993. **//Format for Software**
- Fuminao Okumura and Hajime Takagi, "Maglev Guideway On the Yamanashi Test Line," <http://www.rtri.or.jp/rd/maglev2/okumura.html>, October 24, 1998. **//Format for World Wide Web** (give author and title if named)
- "AT&T Supplies First CDMA Cellular System in Indonesia," <http://www.att.com/press/1095/951011.nsa.html>, Feb 5, 1996. **//Format for World Wide Web**