Hamdard University

Department of Computing

Final Year Project

Plant Disease Detection Using Mobile Application

FYP-028/FL24

Software Design Specifications

Syed Hasnain Imran
(BSAI/1653-2021)
Khadija Rashid Mughal
(BSSE/1814-2021)
Zainab Jamil
(BSAI/2179-2021)

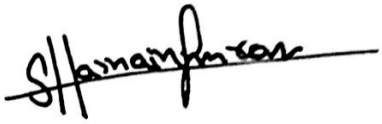
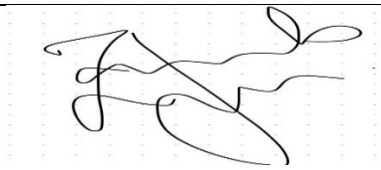
Supervisor
Sir Waqas Pasha

2025

Document Sign off Sheet

Document Information

Project Title	<Plant Disease Detection Using Mobile Application>
Project Code	<FYP-028/FL24>
Document Name	Software Design Specifications
Document Version	<1.0>
Document Identifier	<Project Code>-SRS
Document Status	Draft/ Final
Author(s)	Syed Hasnain Imran Khadija Rashid Mughal Zainab Jamil
Approver	Sir Waqas Pasha
Issue Date	30-10-2024

Name	Role	Signature	Date
Syed Hasnain Imran	Team Lead		17-jan-2025
Khadija Rashid Mughal	Team Member 2		17-jan-2025
Zainab Jamil	Team Member 3		17-jan-2025
Sir Waqas Pasha	Supervisor		17-jan-2025
	Project Coordinator		

Revision History

Date	Version	Description	Author
30-10-2024	0.1	First Draft	Khadija Rashid
26-11-2024	0.2	Revised Draft	Zainab Jamil
10-12-2024	1.0	Final Draft	Khadija Rashid,

Definition of Terms, Acronyms, and Abbreviations

Term	Description
SDS	Software Design Specification
AI	Artificial Intelligence
ANN	Artificial Neural Networks
DL	Deep Learning
CNN	Convolutional Neural Networks
CV	Computer Vision
UI	User Interface
MobileNetV2	Mobile Net Version 2

Table of Contents

1 Introduction

Detection of plant diseases is crucial for maintaining healthy plants and improving nursery productivity. This project aims to develop a mobile application specifically designed for nursery workers to efficiently detect plant diseases. By utilizing advancements in deep learning and image processing, the application will analyze captured or uploaded images of plant leaves to identify potential diseases. The application will offer a user-friendly interface, making it accessible even for individuals with minimal technical expertise. By focusing on reliable disease detection, this tool aims to streamline plant health assessment in nurseries and reduce manual effort.

In recent years, deep learning methods have greatly improved plant disease detection capabilities. One study highlights the effectiveness of convolutional neural networks (CNNs) in identifying plant diseases through image analysis. The study demonstrates how CNN models can accurately classify diseases by analyzing leaf images, facilitating early detection and timely intervention in agricultural practices. This approach surpasses traditional manual methods, as it can quickly process large volumes of plant images with enhanced accuracy and efficiency. By efficiently analyzing numerous images in a short period, CNN-based systems reduce the need for expert involvement and manual inspections, positioning them as a vital tool for modern agricultural practices.

1.1 Purpose of Document

This document presents the design framework for the plant disease detection mobile app. It is designed for developers, testers, supervisors, and future contributors. Adopting an Object-Oriented Design (OOD) approach, the document offers instructions for constructing, managing, and enhancing the app. It outlines the system architecture, database schema, and user interaction flow, promoting clear communication among teams. Furthermore, it acts as a reference for quality assurance and future revisions, ensuring alignment and supporting smooth development.

1.2 Intended Audience

The intended audience for this project includes nursery workers, who will directly use the plant disease detection application for improved plant health management. It also target developers, academic supervisors involved in guiding and building the system and also researchers studying plant disease who can use the application. Additionally, this document serves as a guide for project

team members, including designers, developers, and researchers, to understand the system's objectives, requirements, and deliverables.

1.3 Document Convention

This document is prepared in Times New Roman 12pt font for the text and 14pt bold for section headers and 16 for main heading.

1.4 Project Overview

The Mobile Application for Plant Disease Detection is developed to help nursery workers and gardeners identify plant diseases through image analysis. It employs deep learning, specifically Convolutional Neural Networks (CNNs), to analyze plant images and detect potential diseases. By integrating TensorFlow Lite for on-device processing, the app ensures efficient performance without requiring continuous internet access.

The main goal of the project is to offer an intuitive tool that allows users to upload plant images, receive diagnostic results, and generate downloadable reports. The system utilizes MobileNetV2, a lightweight CNN model optimized for mobile devices, for disease classification. This approach ensures quick and accurate plant disease detection on smartphones, enabling users to take prompt actions to safeguard their plants.

1.5 Scope

- Focuses on detecting and classifying plant diseases from images captured through mobile devices.
- Utilizes Convolutional Neural Networks (CNNs), specifically optimized models like MobileNetV2.
- The project will integrate lightweight models like MobileNetV2 for efficient and accurate processing on mobile devices.
- The app will feature an easy-to-use interface for nursery owners, Includes features like image upload, real-time detection, and logout functionality.

1.6 Not In Scope

- Doesn't include external farming tools or equipment like drones and sensors.
- No advanced editing features for uploaded images like manual adjustments or filters.

2 Design Considerations

The Design Considerations section discusses the key decisions shaping the system's structure and function. It ensures the architecture aligns with project objectives, prioritizes usability, and remains adaptable to future enhancements. This section also highlights how the design addresses constraints, scalability, error handling, and the efficient integration of external systems like TensorFlow.

2.1 Assumptions and Dependencies

System relies on the availability and utilization of the PlantVillage dataset as a fundamental resource for the development and evaluation of the deep learning model. This open-access repository of plant health images offers a vast and diverse range of categorized images essential for effectively training the CNN model. The system assumes that users will utilize mobile devices to capture and upload images of plant leaves, with the real-time detection and disease classification process designed to operate smoothly on these mobile platforms.

2.2 Risks and Volatile Areas

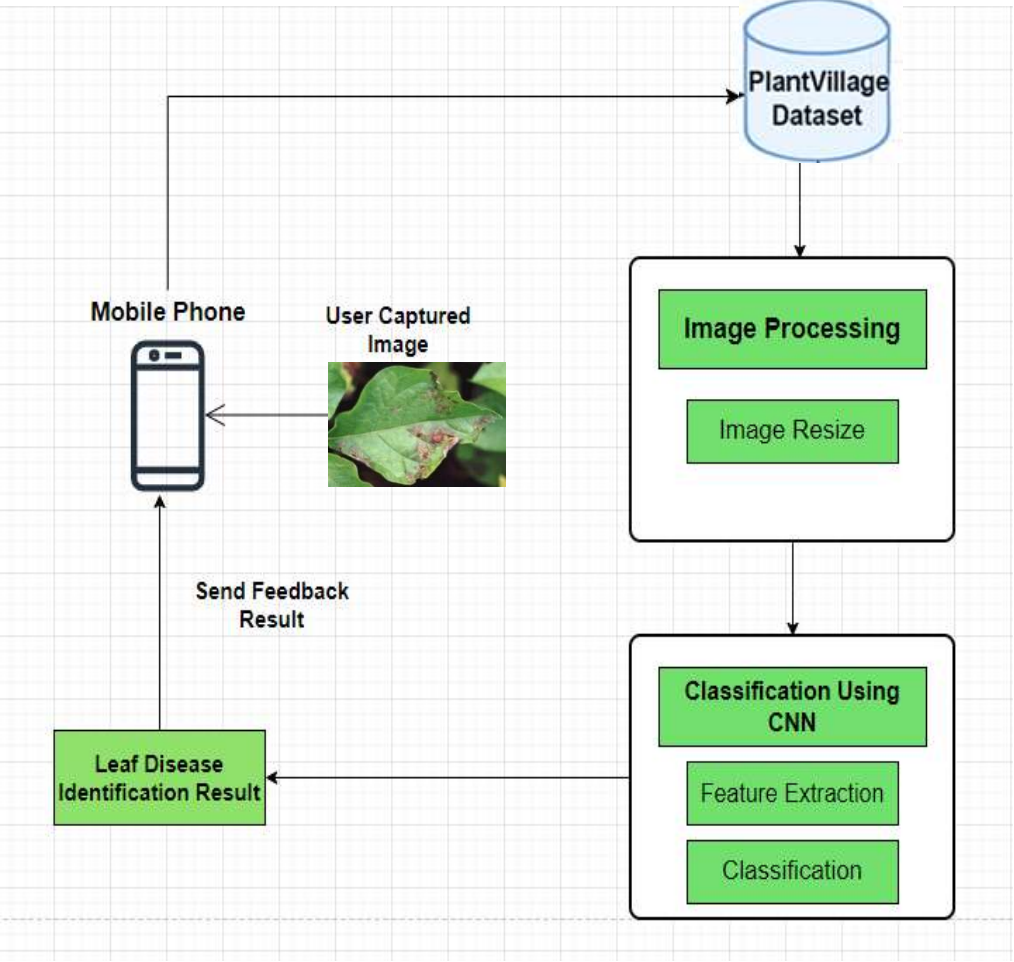
- **Data Quality:** The effectiveness of the disease detection model is closely tied to the quality and variety of the data used for training. Inaccurate or limited data could lead to incorrect diagnoses, reducing the app's overall reliability.
- **Algorithm Performance**: The performance of the deep learning model could be affected by slow processing speeds or its inability to accurately recognize new diseases or plant species, which may impact the app's efficiency and accuracy.
- **User Adoption and Satisfaction**: For the app to succeed, it is crucial that users find it easy to use and reliable. If the app is complicated, prone to errors, or fails to deliver consistent results, it could lead to reduced user engagement and abandonment.
- **Regulatory Compliance**: The app must meet all necessary data protection and privacy standard. Failing to ensure proper data security could lead to legal complications and erode user trust.
- **Resource Constraints**: Mobile devices have limited processing power and storage capacity, which may affect the app's performance. These limitations could cause slow processing or even crashes, particularly when handling large images or datasets.

- **Integration Challenges**: Integrating the deep learning model with the mobile platform and external services like databases could face technical obstacles. Compatibility issues or integration failures might disrupt the app's functionality and reliability.

2.3 System Architecture

The system architecture of the Plant Disease Detection Mobile Application is designed with a modular structure to ensure smooth functionality. The user interface begins with a homepage, offering options to log in or sign up. After authentication, users access the main page, where they can either capture plant images using the mobile camera or upload images from the gallery. These images are analyzed by a disease detection engine, which utilizes preloaded datasets and algorithms to identify plant diseases, such as bacterial spots. The analysis results, including the plant's health status and detected disease, are displayed to the user. The architecture also incorporates a database layer to manage user details, uploaded images, and detection results. Furthermore, the app includes features for secure authentication and logout, ensuring reliability, security, and effective disease detection for nursery workers.

2.3.1 Software Architecture Diagram



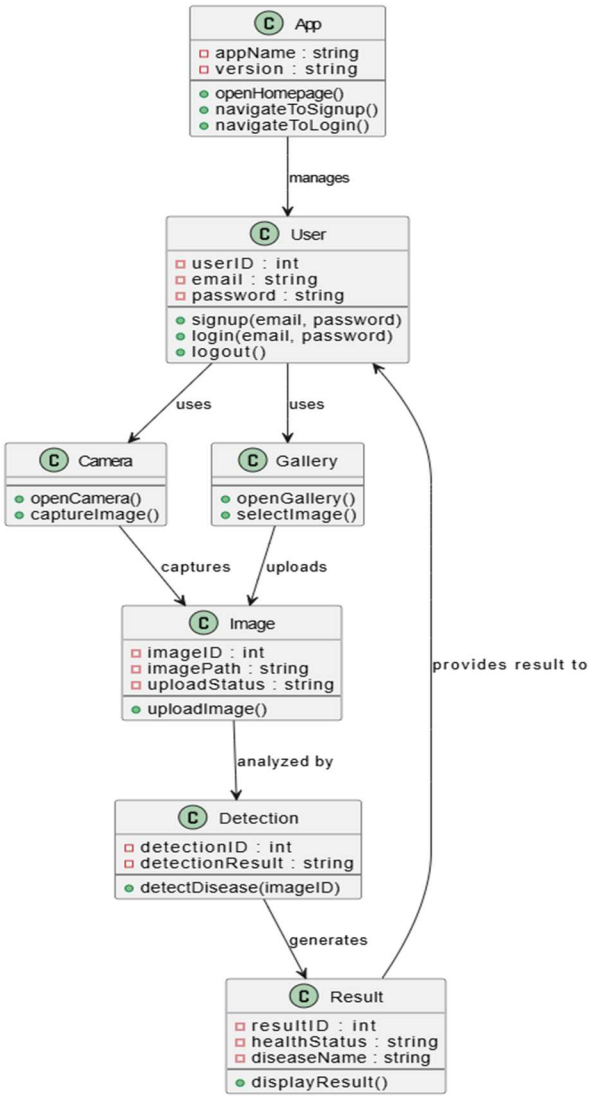
3 Design Strategy

1. **Future System Extension or Enhancement:** To accommodate future system extension or enhancement, the system architecture is designed with modularity and scalability in mind. The use of a layered architecture allows for the independent development and modification of individual components. Each component has well-defined interfaces, facilitating the addition of new features without impacting the entire system. This design decision enables the system to adapt and grow as new requirements emerge.
2. **System Reuse:** The system architecture aims to maximize reusability of components and modules. By separating concerns into distinct layers, components can be reused across different projects or systems.

3. **User Interface Paradigms:** The user interface is designed to be intuitive and user-friendly. The system employs standard user interface paradigms and design principles to ensure familiarity and ease of use for the users. The user interface design follows established best practices to enhance usability and improve the overall user experience.
4. **Concurrency and Synchronization:** The system design considers concurrency and synchronization to handle potential concurrent access to shared resources. For example, if multiple users are accessing the application simultaneously, the system ensures that there are no conflicts in data access.

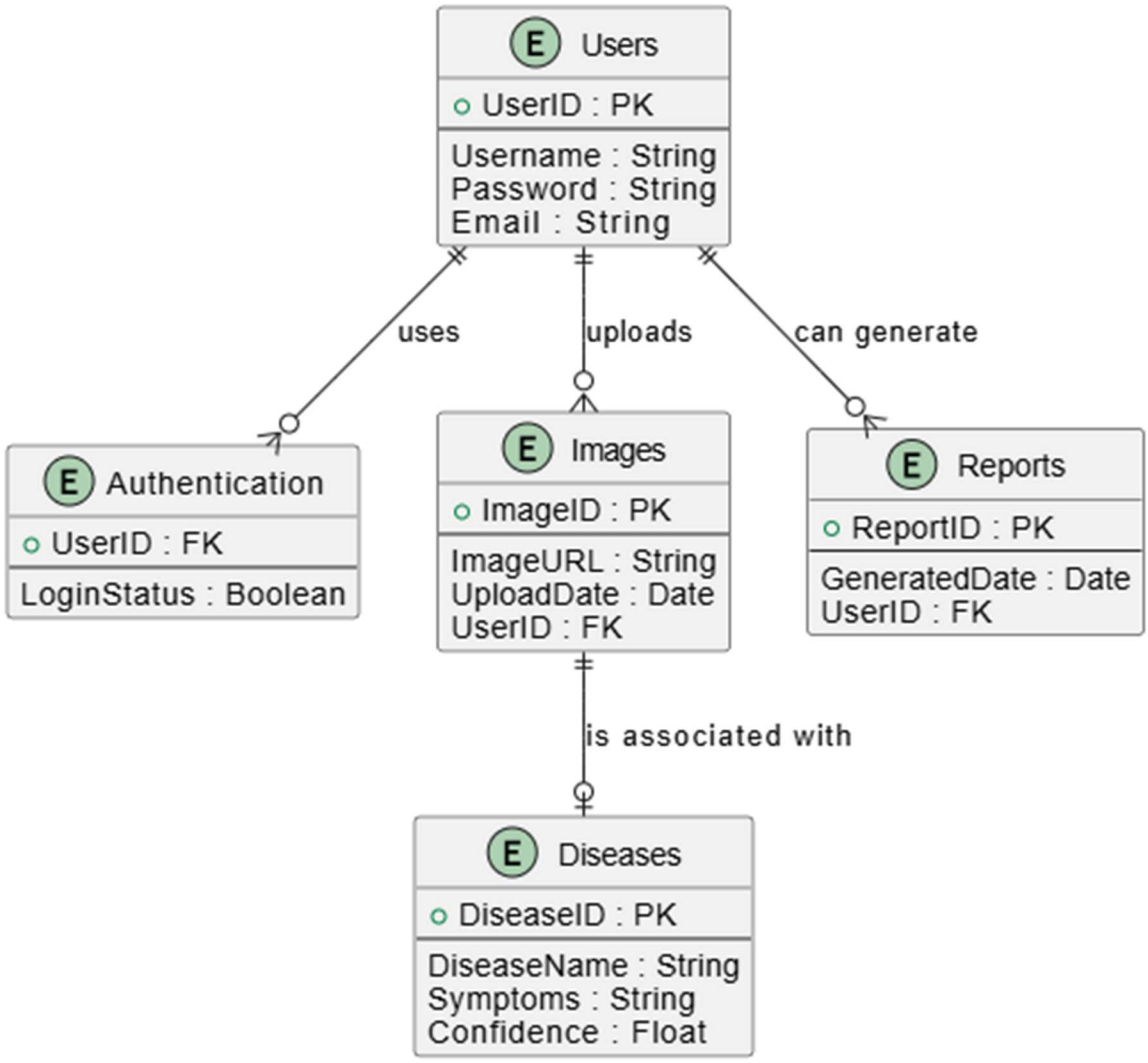
4 Detailed System Design

4.1 Design Class Diagram



4.2 Database Design

4.2.1 ER Diagram

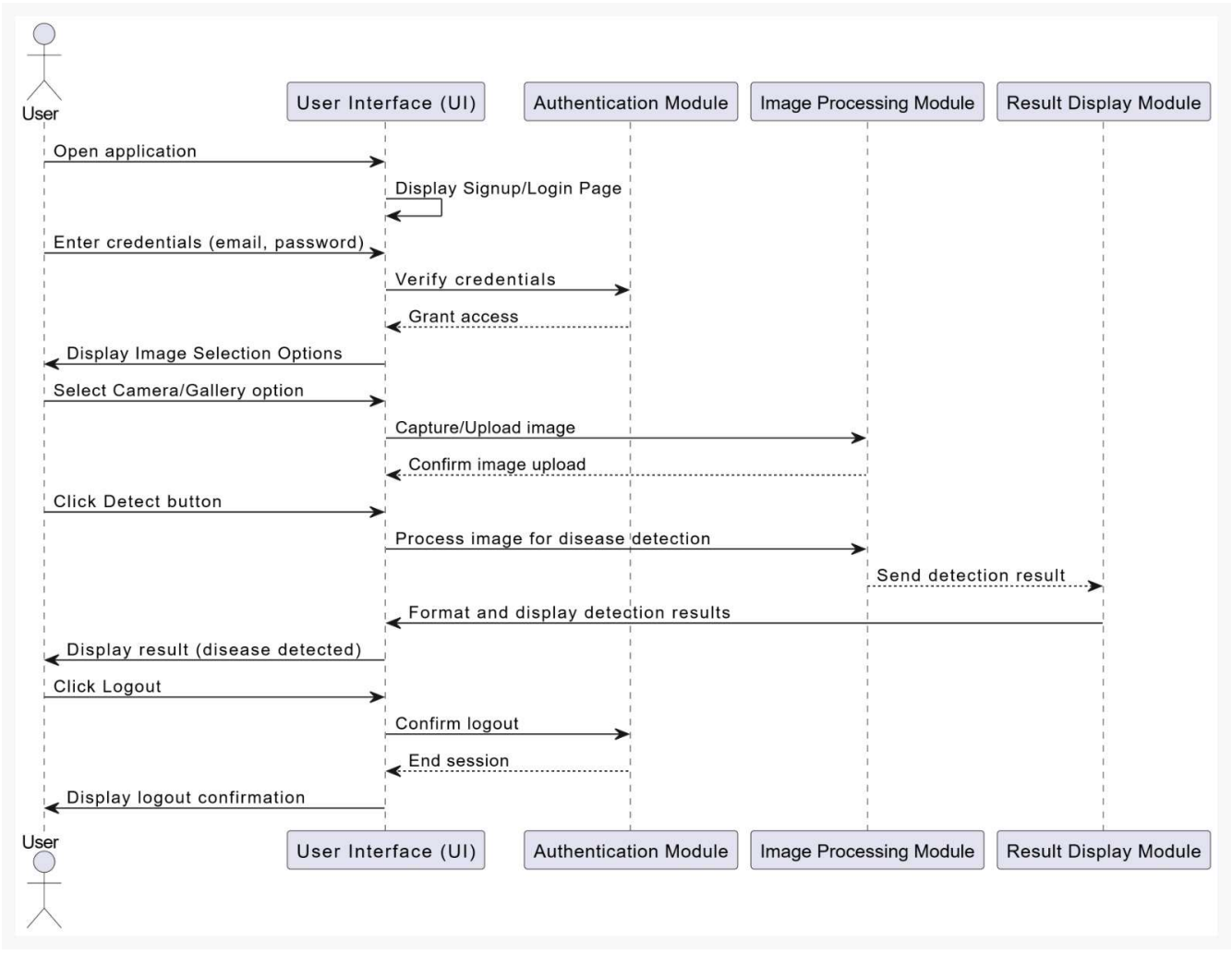


4.2.2 Data Dictionary

Data Name	Type	Description	Possible Values/Format
User Email	String	Email address entered during Signup or Login for user authentication.	Example: user@example.com
User Password	String	Password entered during Signup or Login for user authentication.	Minimum 8 characters, including letters, numbers, and symbols.
Camera Option	Boolean	Represents the user selecting the camera to capture an image.	true (selected), false (not selected).
Gallery Option	Boolean	Represents the user selecting the gallery to upload an image.	true (selected), false (not selected).
Plant Image	File (Image)	The image file selected or captured by the user for disease detection.	JPEG, PNG formats supported.
Upload Status	String	Status of the image upload process.	Pending, Success, Failed.
Detection Status	String	Status of the disease detection process.	Pending, In Progress, Completed.
Detection Result	String	Result of disease detection indicating whether the plant is healthy or a disease is detected.	Healthy, Bacterial Spot, etc.
Logout Action	Boolean	Represents whether the user logs out of the system after viewing the result.	true (user logged out), false (still logged in).

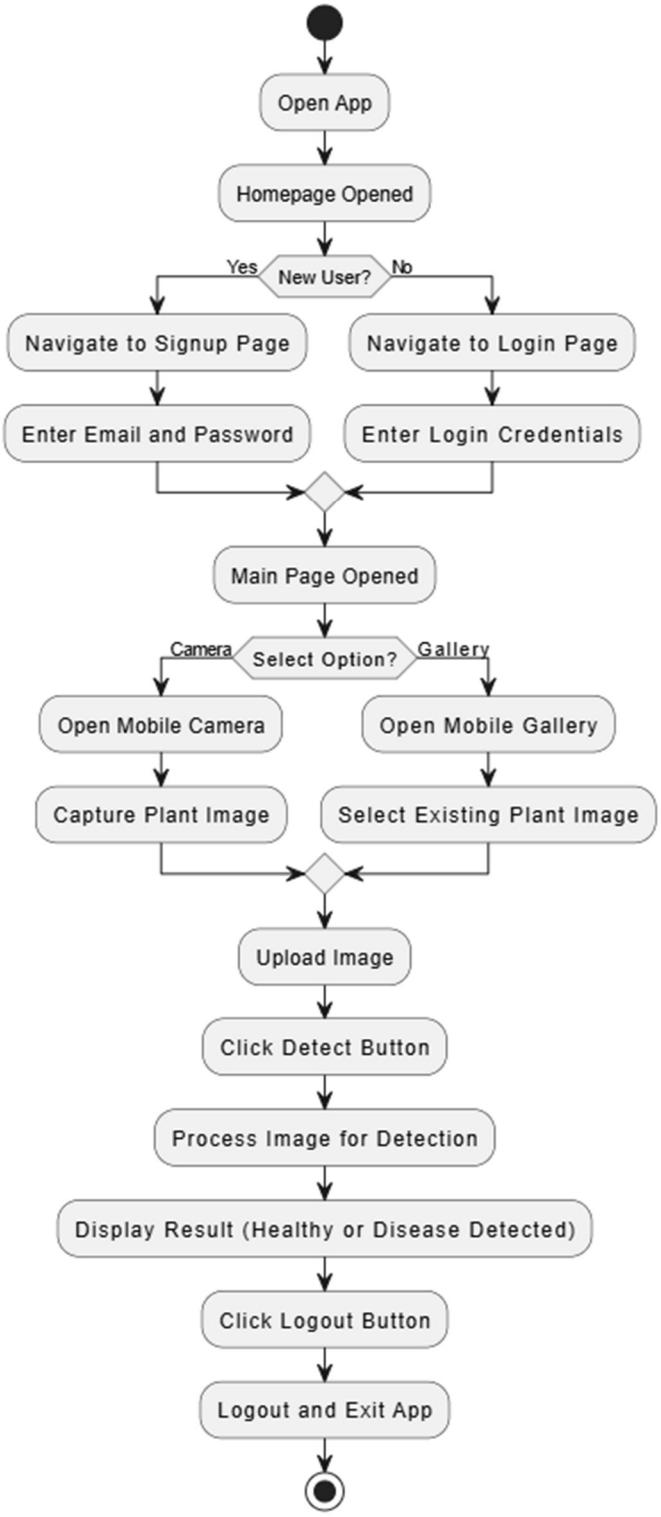
4.3 Application Design

4.3.1 Sequence Diagram



4.3.2 UseCase Diagram

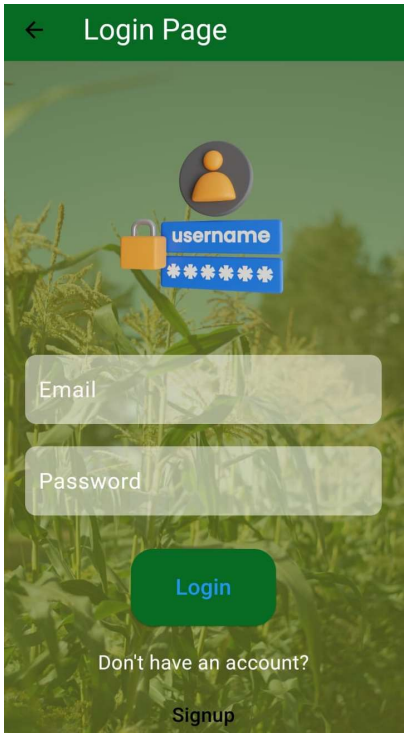
4.3.3 State Diagram



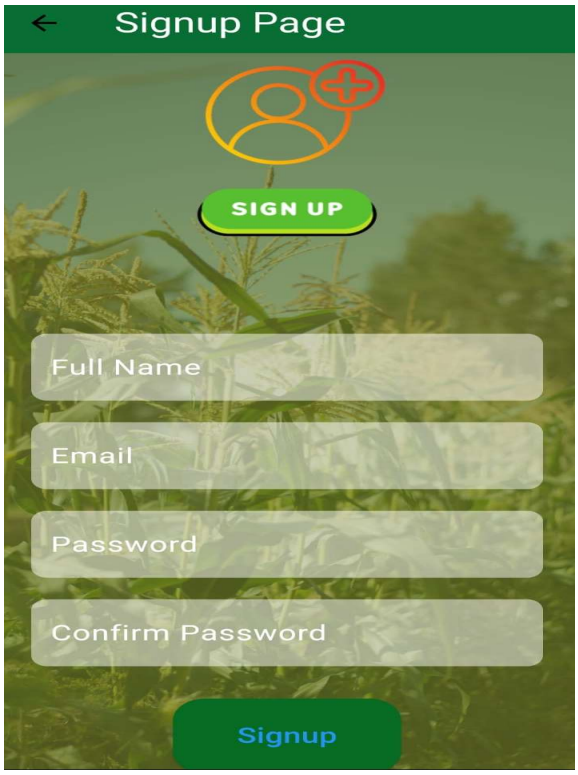
4.4 GUI Design

4.4.1 Mock Screen 1

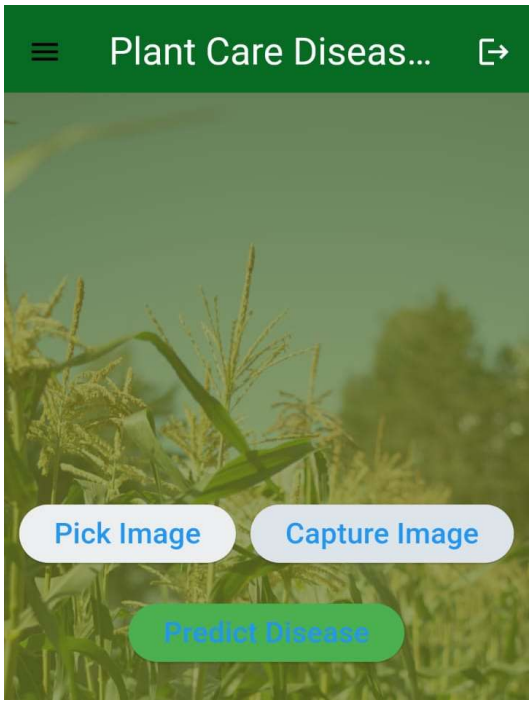
4.4.2 Mock Screen 2



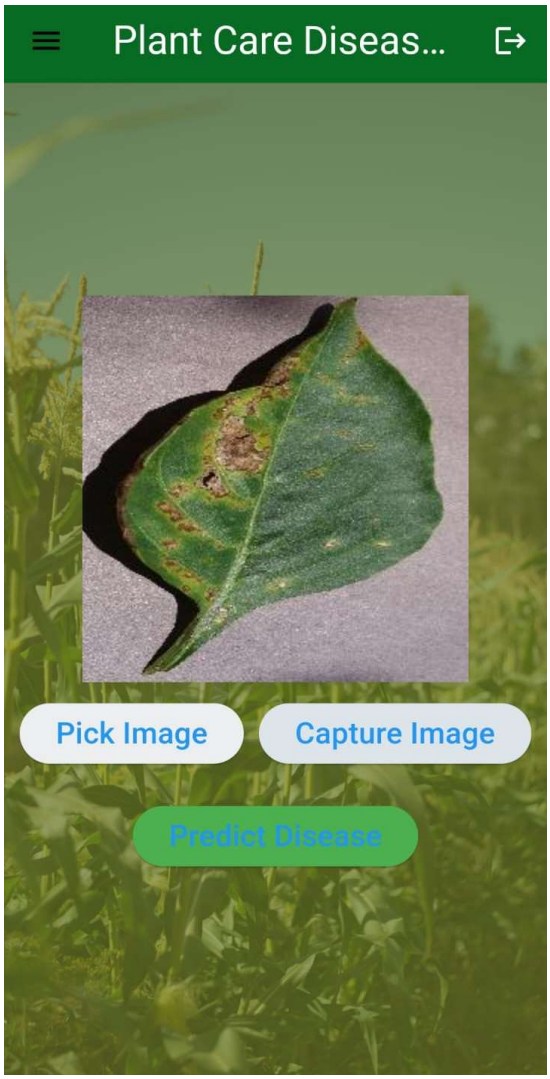
4.4.3 Mock Screen 3



4.4.4 Mock Screen 4



4.4.5 Mock Screen 5



5 References

[1] Deep Learning for Plant Disease Detection, *International Journal of Mathematics, Statistics, and Computer Science, 2024*

- **Authors**: M. M. Khalid and O. Karan
- **Summary**: Explores the application of CNNs for plant disease detection, supporting the use of MobileNetV2 for efficient disease identification.

[2] Using Deep Learning for Image-Based Plant Disease Detection, *Frontiers in Plant Science, 2016*

- **Authors**: S. P. Mohanty, D. P. Hughes, and M. Salathé
- **Summary**: Highlights deep learning methods for image-based plant disease detection, directly supporting the classification methods.

[3] A Mobile-Based System for Detecting Plant Leaf Diseases Using Deep Learning, *Journal of Agricultural Technology and Innovation, 2023*

- **Authors**: A. A. Ahmed and G. H. R. Reddy
- **Summary**: Discusses mobile-optimized implementations of CNNs for real-time plant disease detection on smartphones.

[4] Deep Learning and Content-Based Filtering Techniques for Improving Plant Disease Identification, *Computers and Electronics in Agriculture, 2020*

- **Authors**: Isinkaye, F. O., Olusanya, M. O., & Singh, P. K.
- **Summary**: Reviews techniques combining deep learning and content-based filtering for plant disease detection and treatment recommendations.

[5] An Open Access Repository of Images on Plant Health to Enable the Development of Mobile Disease Diagnostics

- **Authors:** David P. Hughes, Marcel Salathé
- **Summary:** This paper presents the PlantVillage dataset, an open-access resource of plant health images, aimed at advancing plant disease detection research and enabling mobile diagnostic tool development.

[6] Plant disease detection and classifcation techniques, *A comparative study of the performances,* ***International Journal of Computer Applications***, vol. 182, no. 15, pp. 1-9, 2021.

- **Authors:** Wubetu Barud Demilie
- **Summary:** This paper provides a comprehensive comparative analysis of various techniques for plant disease detection and classification, evaluating the performance of each method and highlighting the strengths and weaknesses of traditional and modern approaches.

[7] C. Adams, "Role of Use Case Diagram in Software Development," 26 May 2022. [Online]. Available: https://www.researchgate.net/publication/322991847_role_of_use_case_diagram_in_software_development