Crop Care App

Ali Haider

Iqra Rehman

DEPARTMENT OF COMPUTER SCIENCES

COMSATS UNIVERSITY ISLAMABAD,

ATTOCK CAMPUS – PAKISTAN

SESSION 2019-2023

Crop Care App

Undertaken By:

Ali Haider

CUI/FA19-BSE-002/ATK

Iqra Rehman

CUI/FA19-BSE-006/ATK

Supervised By:

Dr. Muhammad Shehzad Faisal

A DISSERTATION SUBMITTED AS A PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCES

COMSATS UNIVERSITY ISLAMABAD,

ATTOCK CAMPUS – PAKISTAN

SESSION 2019-2023

UNDERTAKEN

We hereby declare that this software, neither whole nor as a part has been copied out from any source. We further declared that we have developed this software and accompanied the report entirely based on our efforts. If any part of this project is proved to be copied from any source. We will stand by the consequences. No portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Ali Haider

FA19-BSE-002

Iqra Rehman

FA19-BSE-006

FINAL APPROVAL

It is to certify that the final year project of BS(SE) "Crop Care Application" is developed by Ali Haider CUI/FA19-BSE-002/ATK and Iqra Rehman CUI/FA19-BSE-006/ATK under the supervision of "Dr. Muhammad Shehzad Faisal" and "Dr. Muhammad Najam Dar". It is fully adequate, in scope and quality for the degree of Bachelor of Science in Software Engineering.

Committee:

1. External Examiner

_____

Dr. Ali Javed
Associate Professor
UET Taxila

2. Supervisor _____

Dr. Muhammad Shehzad Faisal
Assistant Professor
COMSATS University Islamabad
Attock Campus

3. Head of Department _____

Dr. Khalid Iqbal
Associate Professor HOD (CS)
COMSATS University Islamabad
Attock Campus

DEDICATION

We dedicate this piece of work to our parents who brought us up in an appropriate educational atmosphere, and our worthy teachers who had been a source of encouragement, guidance, and enlightenment at each step of our life. Our parents helped us since the beginning of our higher education to cope up with all challenges that we face.

ACKNOWLEDGEMENT

Praise be to Allah, the most Beneficent and the most Merciful, the lord of the world, who guides us in the darkness and help us in difficulties. All powers are due to His Almighty favors.

We express our deep gratitude to our research supervisor, Dr. Muhammad Shehzad Faisal and Mr. Muhammad Najam Dar, whose valuable guidance and supervision make this work more colorful and educative. We believe this study would not have been completed without his valuable suggestions. We are deeply indebted to him for his encouragement and continual help during this work.

Ali Haider

FA19-BSE-002

Iqra Rehman

FA19-BSE-006

PROJECT BRIEF

PROJECT NAME	CROP CARE APP
ORGANIZATION NAME	COMSATS UNIVERSITY ISLAMABAD, ATTOCK CAMPUS
OBJECTIVE	PLANT DISEASE DETECTION AND EDUCATING FARMER
UNDERTAKEN BY	ALI HAIDER CUI/FA19-BSE-002/ATK IQRA REHMAN CUI/FA19-BSE-006/ATK
SUPERVISED BY	DR. MUHAMMAD SHEHZAD FAISAL ASSISTANT PROFESSOR COMPUTER SCIENCE CUI, ATTOCK CAMPUS
STARTED ON	OCTOBER 2022
COMPLETED ON	EXPECTED JUNE 2023
COMPUTER USED	HP LAPTOP, CORES i5

	3.02 GHZ PROCESSOR 24GB RAM
SOURCE LANGUAGE	PYTHON 3.10, DART
OPERATING SYSTEM	WINDOWS 10
TOOLS USED	STARUML, MS WORD PYTHON 3.10 IDE, ANDRIOD, FLUTTER

ABSTRACT

Agriculture is not only the main support for Pakistan's economy but also an important source of income for Pakistani people. As the population of Pakistan is increasing rapidly, the demand for food is also growing. To fulfill the need for food, we need to get the maximum yield from crops and vegetables. Many factors are affecting the maximum production rate such as pests, crop diseases, climate changes, and much more. Vegetable plant diseases are one of the major causes of loss in plant production. Therefore, the farmers must promptly diagnose different types of crop and vegetable plant diseases to stop their spread within their fields. The diagnosis of plant diseases is a very critical process and any misdiagnosis of diseases will lead to the use of wrong fertilizers that will also affect the production and growth of crops and vegetables.

To overcome this problem, we are proposing a system that will detect the diseases of vegetables from leaf images taken in real-time and then recommend a proper handling procedure for detected diseases.

The "Crop Care App" is a mobile application developed to help farmers. The application will detect the disease of vegetable plants from its leaf image taken in real-time or select from a gallery using a Convolutional Neural Network. For this purpose, we are using a PlantVillage dataset that consists of more than 23000 images for the most common 12 plan diseases categories in 3 vegetable species, including potato, pepper, and tomato. These plant diseases include Pepper Bell Bacterial spot, Potato Early blight, Potato Late blight, Tomato Bacterial spot, Tomato Early blight, Tomato Late blight, Tomato Leaf Mold, Tomato Septoria leaf spot,

Tomato Spider Mites Two-spotted spider mite, Tomato Target Spot, Tomato Yellow Leaf Curl Virus, Tomato mosaic virus.

After detecting a disease, the system then recommends a proper handling procedure for that disease. The daily work planner feature helps farmers to plan their everyday activities. It provides information about all the plant's pests and diseases and gives a solution to handle these. The weather condition is important in sowing, watering, spraying fertilizers and chemicals, and harvesting operations. Therefore, the weather forecasting feature helps farmers to know about everyday weather. The system will also provide proper cultivation tips for vegetables for farmers that are new to farming.

Table of Contents

Table of Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
GUI	Graphical User Interface
UI	User Interface
ERD	Entity Relationship Diagram
DFD	Dataflow Diagram

Chapter 1
INTRODUCTION

1. Introduction

The “Crop Care App” is a mobile application developed to help farmers. The application will detect the disease of vegetable plants from its leaf image taken in real-time using a Convolutional Neural Network. For this purpose, we are using a PlantVillage dataset that consists of more than 23000 images for the most common 12 plant disease categories in 3 vegetable species, including potato, pepper, and tomato. These plant diseases include Pepper Bell Bacterial spot, Potato Early blight, Potato Late blight, Tomato Bacterial spot, Tomato Early blight, Tomato Late blight, Tomato Leaf Mold, Tomato Septoria leaf spot, Tomato Spider Mites Two-spotted spider mite, Tomato Target Spot, Tomato Yellow Leaf Curl Virus, Tomato mosaic virus. After detecting a disease, the system will also recommend a treatment procedure for that disease. The daily work planner feature helps farmers to plan their everyday activities. It provides information about all the plant's pests and diseases, and a solution to treat them. The weather condition is important in sowing, watering, spraying fertilizers and chemicals, and harvesting operations. Therefore, the weather forecasting feature helps farmers to know about everyday weather. The system will also provide proper vegetable cultivation tips to farmers new to farming.

1.1 Brief

Crop Care app is a disease detection app for helping farmers in detecting vegetable plant diseases. This will reduce disease detection time and improve the traditional method of disease detection. For detection purposes, the system will use a Convolutional neural network. We aim to help farmers not only in disease detection but also educate them about farming. For this purpose, we also provide farming education by providing proper cultivation procedures, pests, insects, and disease information and provide recovery procedures about how to recover plants from diseases, pests, and insect attacks. Fertilizers and chemicals information will be provided to help farmers in growing healthy vegetables and crops.

As farming requires proper planning like on which time to provide water to plants, when to spray chemicals and fertilizers, etc. so to help farmers in planning their schedule, the system will be providing a daily planner module that will help them in organizing their everyday tasks.

1.2 Relevance to course module

1.2.1 Machine Learning

In Machine learning, we studied different algorithms that are used to train models to perform tasks automatically. Therefore, this course helps us in model implementation, training, and testing.

1.2.2 Computer Vision

This course helps in image dataset preprocessing and arranging it in the same format for better model training.

1.2.3 Human-Computer Interaction

This course helps us to design an interactive system that is easy and comfortable for the user to understand and use.

1.2.4 Database Management (DBMS)

We learn how to maintain data in a database that helps us during maintaining a database of our application i.e., login, registration, etc.

1.3 Project Background

Agriculture is not only the main support for Pakistan's economy but also an important source of income for Pakistani people. As the population of Pakistan is increasing rapidly, the demand for food is also growing. To fulfill the need for food, we need to get the maximum yield from crops and vegetables. Many factors affect the maximum production rate, such as pests, crop diseases, climate changes, etc. Crop and vegetable plant diseases are one of the major causes of loss in yield. Therefore, the farmers must promptly diagnose different types of crop and vegetable plant diseases to stop their spread within their fields. Diagnosing diseases is a very critical process, any misdiagnosis of crop and vegetable diseases will lead to the use of the wrong fertilizers, which will also affect plant yield and growth.

To overcome this problem, the Crop Care App will detect the diseases of vegetable plants from leaf images taken in real-time or from the gallery and recommend a proper treatment procedure for detected diseases.

1.4 Literature Review

Different agricultural apps already exist in the play store, but all these apps have different levels of skills and features. Some of them are discussed below:

- **Leaf Doctor**

A leaf detector is a plant disease detection system that only detects the disease of plants from leaf images. But it does not provide any other functionality like farming guidelines for farmers, crop pest information, proper use of fertilizers, and much more.

- **Crop Diagnosis**

This application has the functionality to provide cultivation information for farmers. It also provides a matching chemical (fertilizers) that can mitigate the problem. In this application, the user needs to answer a smart questionnaire to diagnose a crop disease. Then, the app suggests the most likely diagnosis.

- **Smart Kassan**

Smart kassan is an android application that provides information about all the crops. Information about crop diseases and their treatment solution is also provided. It recommends crop sowing according to the areas. All the farming-related shop information is also provided according to the area. But it does not have a disease-detection feature and fertilizer recommendations are also not provided.

1.5 Analysis of Literature Review

Table 1: Analysis with other's App

Functionalities	Proposed	Smart Kassan	Plantix	Crop Diagnosis	Leaf Detector
User Verification	✓	✘	✘	✘	✘
Disease Detection from image	✓	✘	✓	✘	✓
Curing Procedure Recommendation	✓	✘	✓	✓	✘

Cultivation Tips	✓	✓	✓	✓	✘
Farmers Daily Planner	✓	✓	✘	✘	✘
Crops Pests and Diseases	✓	✘	✓	✘	✘
Weather Forecasting	✓	✓	✘	✘	✘

1.6 Methodology and System Lifecycle

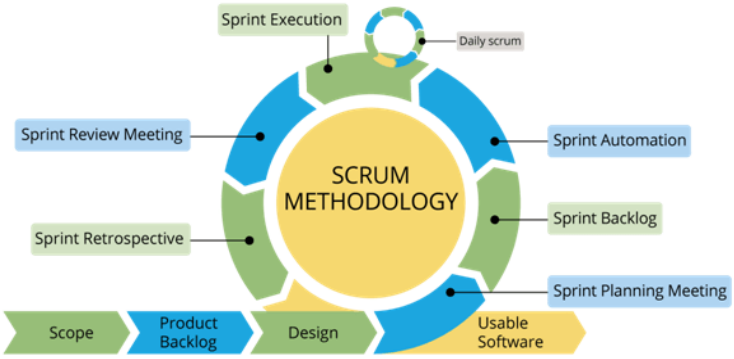

Figure 1: Methodology

As we are developing Crop Care App in incremental form and changes may be required after showing the developed part of a project to the supervisor. For that purpose, we are using the scrum methodology.

Chapter 2

PROBLEM DEFINITION

2 Problem Statement

Pakistan is an agricultural country and a large part of its economy depends on agriculture. However, there is a large difference between the acquired and the actual output of crop and vegetables yield. The main factor affecting crop and vegetables yield is their diseases, which are very difficult to detect by inexperienced farmers.

Therefore, the idea of the Plant Disease Detection system is to overcome this problem. It will help in automatic disease detection and make farming easy for new framers as it helps them choose land, preparation, and all cultivation steps.

2.1 Development Requirements

Here we discuss the tools and technology that we are using in our project.

Table 2: Tools

Tools	Version	Purpose
Python IDLE	2022	IDE
Firebase	Current	DBMS
Flutter	3.3	IDE
Adobe XD	2020	UI UX Design Work

Table 3: Technology

Technology	Version	Purpose
Dart	2.17	Front End Development
Dart	2.17	Backend Development
Python	3.10	Programming language

Chapter 3

REQUIREMENTS ANALYSIS

3 Requirement Analysis

Software Requirement Specification (SRS) provides a basic understanding of functions as well as non-functional requirements. We can consider it as a starting point for a project because it serves as a written contract between the client and the organization about the features and functionalities of the project. With SRS's help, clients and the organization will be clear about the deliverable project.

3.1 Use Case Diagram

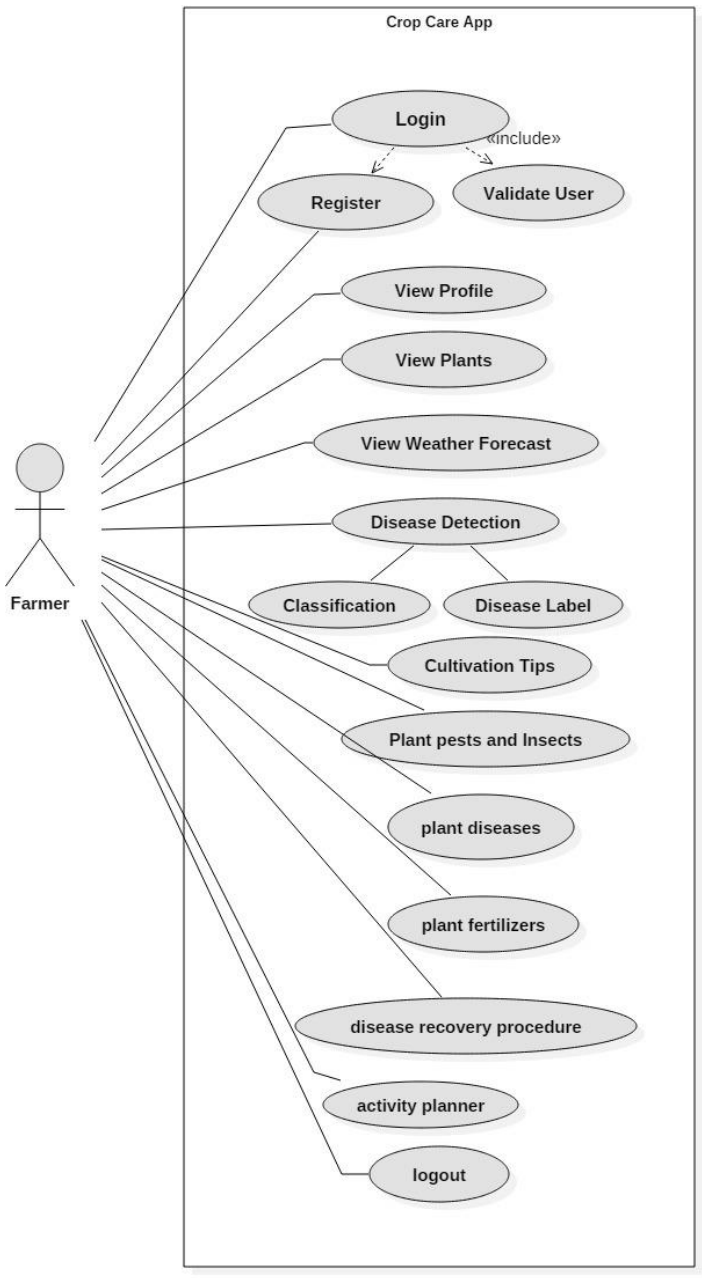

Figure 2: Use case Diagram

3.2 Use Case Detail

Table 4: Registration Use case

Use case name	**Registration/Signup**
Actor	Farmer
Summary	Users sign up for the app.
Pre-condition	Users should download the app on his/her device The user should be on the sign-up page. The user must have entered valid details
Post-condition	The user will redirect to the login page.
Extend	Nil
Normal course of events	1. Open App. 2. Click on sign up. 3. Enter require data Click on the sign-up button to proceed.

Table 5: Login Use case

Use case name	**Login**
Actor	Farmer
Summary	Register users will login to the app.
Pre-condition	Users should download the app on his/her device The user should be on the sign-in page. The user must have entered valid details The user should be registered in the app database
Post-condition	The user is redirected to the dashboard/Home page.
Extend	Registration

Normal course of events	1. Open the application. 2. Click on login. 3. Enter require data 4. Click on the login button to proceed.

Table 6: Edit Profile

Use case name	**Edit Profile**
Actor	Farmer
Summary	Registered users can edit their profiles.
Pre-condition	Users should download the app on his/her device Users should be on the Profile page. The user should be registered in the app database
Post-condition	The change will be applied to the user profile.
Extend	Nil
Normal course of events	1. Open the application. 2. Login and move to the home page 3. Click on Profile Icon. 4. Click edit profile 5. Change the profile info like profile dp, user name, etc. 6. Click on save changes.

Table 7: View Plants Use case

Use case name	**View Plants**
Actor	Farmer
Summary	Register users can view information about different crops plants and vegetables like their growing season, suitable soil, etc.
Pre-condition	Users should download the app on his/her device The user should be registered in the app database

Post-condition	The user is redirected to the Framing page where they can view information about different crops plants, and vegetables like their growing season, suitable soil, etc.
Extend	Nil
Normal course of events	1. Open the application. 2. Login and move to the home page 3. Click on Farming and the plant list will appear 4. Click on the plant whose info you want to see

Table 8: View Weather Forecast Use case

Use case name	**View Weather Forecast**
Actor	Farmer
Summary	Register users can view weather-related information of that day.
Pre-condition	Users should download the app on his/her device The user should be registered in the app database
Post-condition	The user is redirected to the weather forecast page where they can view The weather of that day.
Extend	Nil
Normal course of events	1. Open the application. 2. Login and move to the home page 3. Click on weather forecasting where they can view weather details.

Table 9: Disease Detection Use case

Use case name	**Disease Detection**
Actor	Farmer
Summary	Register users will detect the disease of plants by just taking a picture of a diseased leaf.
Pre-condition	Users should download the app on his/her device

	The user should be registered in the app database The user must take a picture of the plant leaf for detection.
Post-condition	The disease label will be displayed.
Extend	Nil
Normal course of events	1. Open the application. 2. Login and move to the home page 3. Click on Disease Detection, and take a picture. 4. The system will display the label of plant disease.

Table 10: Take Image Use case

Use case name	**Take Image**
Actor	Farmer
Summary	Registered users will take images as an initial step for disease detection.
Pre-condition	Users should download the app on his/her device The user should be registered in the app database The user must take a picture of the plant leaf for detection.
Post-condition	The disease label will be displayed.
Extend	Nil
Normal course of events	1. Open the application. 2. Login and move to the home page 3. Click on Disease Detection, and take a picture. 4. The system will display the label of plant disease.

Table 11: Classification Use case

Use case name	**Classification**
Actor	System
Summary	The input image will be classified.
Pre-condition	Users should download the app on his/her device The user should be registered in the app database

	The user must take a picture of the plant leaf for detection.
Post-condition	The disease label will be displayed.
Extend	Nil
Normal course of events	1. Open the application. 2. Login and move to the home page 3. Click on Disease Detection, and take a picture. 4. The system will display the label of plant disease.

Table 12: Disease Label Use case

Use case name	**Disease Label**
Actor	System
Summary	After the classification of each feature, the leaf image is matched with the trained model for accuracy and generating the results.
Pre-condition	Users should download the app on his/her device The user should be registered in the app database The user must take a picture of the plant leaf for detection. Image classification has been done
Post-condition	The disease label will be displayed.
Extend	Nil
Normal course of events	1. Open the application. 2. Login and move to the home page 3. Click on Disease Detection, and take a picture. 4. After classification, the system will display the label of plant disease.

Table 13: Cultivation tips Use case

Use case name	**Cultivation Tips**
Actor	Farmer

Summary	Register users can view information about different crops plants and vegetables like their cultivation procedure.
Pre-condition	Users should download the app on his/her device The user should be registered in the app database
Post-condition	The user is redirected to the Framing page where they can view cultivation information about different crops plants, and vegetables.
Extend	Nil
Normal course of events	1. Open the application. 2. Login and move to the home page 3. Click on Farming and the plant list will appear 4. Click on the plant whose info you want to see and then click cultivation tips. 5. The cultivation information will be displayed on the screen.

Table 14: View plants pests and Insects Use case

Use case name	**View Plants Pests and Insects**
Actor	Farmer
Summary	Register users will view pests and insect information about different crops and vegetable plants.
Pre-condition	Users should download the app on his/her device The user should be registered in the app database
Post-condition	The user is redirected to the Framing page where they can view information about pests and insects of different crops plants and vegetables.
Extend	Nil
Normal course of events	1. Open the application. 2. Login and move to the home page 3. Click on Farming and the plant list will appear

	4. Click on the plant whose info you want to see and then click pests and insects. 5. The pests and insects information tab will be opened.

Table 15: View plants Diseases Use case

Use case name	**View Plants Diseases**
Actor	Farmer
Summary	Register users will view information about diseases of different crops and vegetable plants.
Pre-condition	Users should download the app on his/her device The user should be registered in the app database
Post-condition	The user is redirected to the Framing page where they can view information about diseases of different crops and vegetable plants.
Extend	Nil
Normal course of events	1. Open the application. 2. Login and move to the home page 3. Click on Farming and the plant list will appear 4. Click on the plant whose info you want to see and then click diseases. 5. The diseases information tab will be opened.

Table 16: View plants Fertilizers Use case

Use case name	**View Plants Fertilizers**
Actor	Farmer
Summary	Register users will view information about Fertilizers for different crops and vegetable plants.
Pre-condition	Users should download the app on his/her device The user should be registered in the app database

Post-condition	The user is redirected to the Framing page where they can view information about fertilizers for different crops and vegetable plants.
Extend	Nil
Normal course of events	1. Open the application. 2. Login and move to the home page 3. Click on Farming and the plant list will appear 4. Click on the plant whose info you want to see and then click fertilizers. 5. The fertilizers information tab will be opened.

Table 17: Plan activities Daily Planner Use case

Use case name	**Plan Activities Daily Planner**
Actor	Farmer
Summary	Registered users will plan their everyday farming activities.
Pre-condition	Users should download the app on his/her device The user should be registered in the app database
Post-condition	The user activities will be added to the daily planner.
Extend	Nil
Normal course of events	1. Open the application. 2. Login and move to the home page 3. Click the Daily Planner icon. 4. To add a new activity, click the add tab and write the name of and activity also the time when the activity will be performed. 5. Then click ok, and a new activity will be added to the planner. 6. To delete the activity, click on the delete button. 7. To edit the existing activity timing and other details, click on the edit and change the time and other details. 8. Then click ok, and changes will be applied to the activity in the planner.

Table 18: Logout Use case

Use case name	**Logout**
Actor	Farmer
Summary	Register users will logout from the app.
Pre-condition	Users should download the app on his/her device The user should be registered in the app database
Post-condition	The user is redirected to the login page.
Extend	Nil
Normal course of events	1. Open the application. 2. Click on the User Profile icon. 3. Click on the logout button. 4. The user will be logout and directed to the login page.

3.3 Functional Requirements

Functional requirements involve processes, methods, and functionalities that a system is supposed to perform. The following are the functional requirements of our system:

3.3.1 Login

The login page will allow a user to gain access to an application by entering their username and password. The user is needed to register in an application before login to the application.

3.3.2 Registration

Registration only happens the first time you access the system. The need to be a registered user before using the system. The user provides their email/username and password for registering to the system.

3.3.3 Weather Forecast

This functionality will allow farmers to know about the weather condition so that they can plant their activities according to that.

3.3.4 Disease Detection

The system will use the CNN algorithm to develop a model for detecting plant diseases. The model will detect the disease of the plant from its image taken by the farmer.

3.3.5 Farmer Daily Planner

To help farmers in planning their schedules, the system will be providing a daily planner module which will help them in organizing their everyday tasks.

3.3.6 Farming

In this farming module, we are educating farmers by providing proper cultivation procedures, pests, insects, and disease information and also provide recovery procedures about how to recover plants from diseases, pests, and insect attacks. Fertilizers and chemicals information will be provided to help farmers in growing healthy vegetables and crops.

3.3.6.1 Cultivation Tips

As Farmers require proper knowledge of crops and vegetable cultivation, so to help them, we will be providing details cultivation information. This will help the farmer throughout the cultivation process.

3.3.6.2 Diseases

As diseases are the main factor that affects the growth and productivity of crops and vegetables, before treating them ones needs to know plant diseases and their names. In this section, all the plants along with their possible diseases will be mentioned.

3.3.6.3 Fertilizers

As for healthy growth and productivity of plants, using suitable fertilizers and chemicals is a very important factor. So this section will provide fertilizer information and their use according to the vegetables and crops.

3.3.6.4 Pests and the Insects

Pests and insects can also affect the growth and productivity of crops. So this section will provide pests and insects information and which types of crops and vegetables can have these insects.

3.3.6.5 Disease recovery procedures

As diseases and insects are harmful to crops and vegetables, they need to be treated immediately. So disease recovery procedures will provide a proper method to treat these diseases and insects.

3.4 Non- Functional Requirements

Non-Functional requirements are those requirements that specify the quality of the system. Non-functional requirements of our system are:

3.4.1 Performance

Whenever the user login retrieving all the data must be fast. Retrieving and showing data in visualized forms does not take too long. By using a better algorithm, we improve detection time and accuracy.

3.4.2 Availability

The Crop Care app is available for its user 24/7 unless some maintenance or upgradation will not occur.

3.4.3 Reliability

Our system is a reliable source of farming applications that have feature to detect plant diseases from its leaf images.

3.4.4 Efficiency

The Crop Care App is using the best algorithm to make the app faster and more efficient.

3.4.5 Flexibility

The Crop Care provides provide farming information like plant cultivation procedures, related diseases, fertilizers, pests, insects, etc. which will be very helpful for people who are new to farming. Daily planners make it easy for farmers to plan their everyday farming-related activities

3.4.6 Usability

Crop Care App GUI is user-friendly and easy to understand. Mostly its use is similar to the existing system so the user does not need to learn about our app from a basic level. The new feature is also understandable.

Chapter 4

DESIGN AND ARCHITECTURE

4 Design and Architecture

In this chapter, we describe the design that is used in our system implementation. The design of the system is developed according to the requirements collected in the previous section. After gathering all requirements, the next step is to start planning, how we are going to develop our project, and how many resources, costs, time, benefits, and other items are required.

The result of this activity ends in systems architecture showing components and internal structure of the system.

4.1 Flowchart

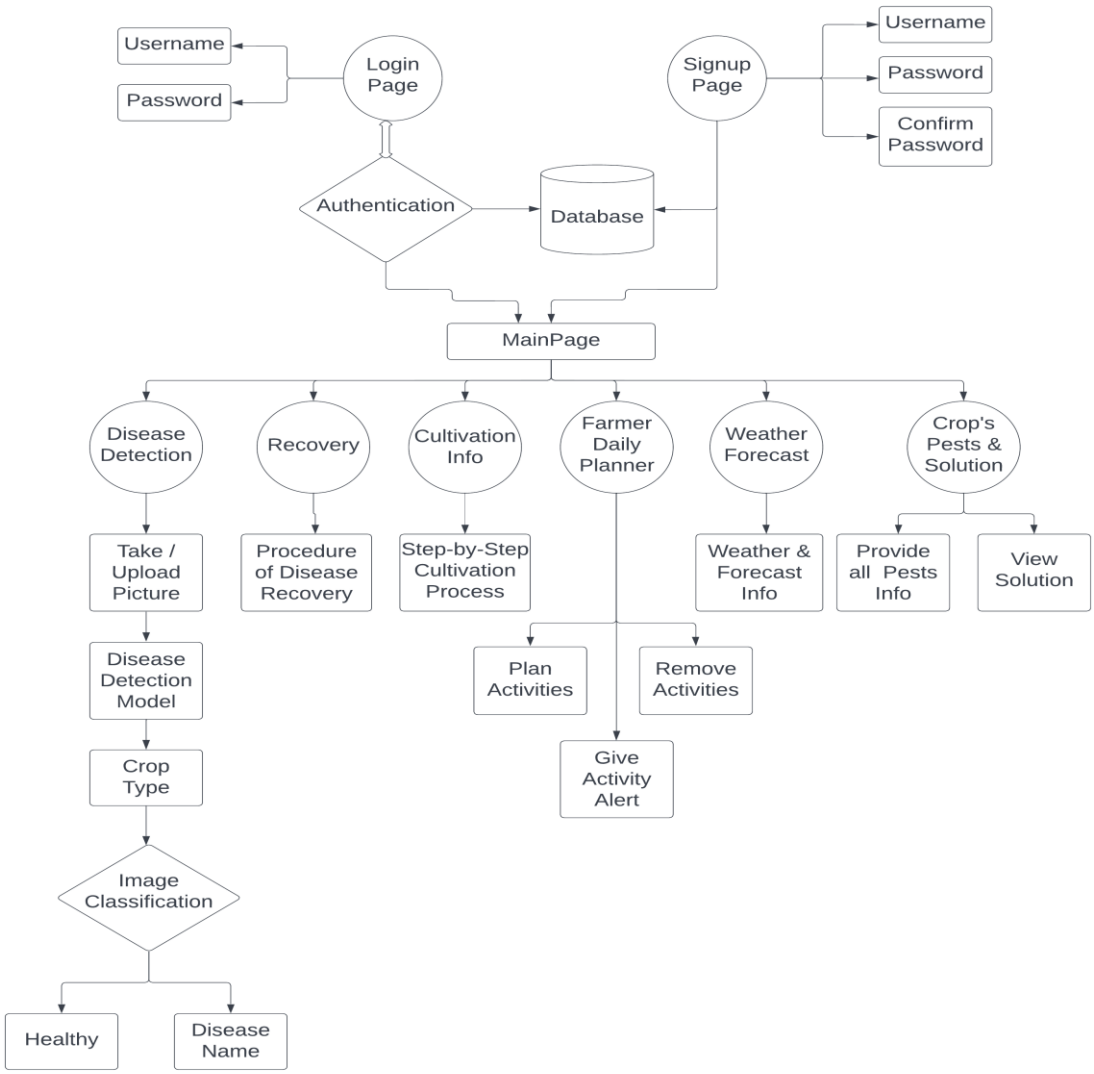

Figure 3: Flowchart

4.2 Activity Diagram

Figure 4: Activity Diagram

4.3 Sequence Diagram

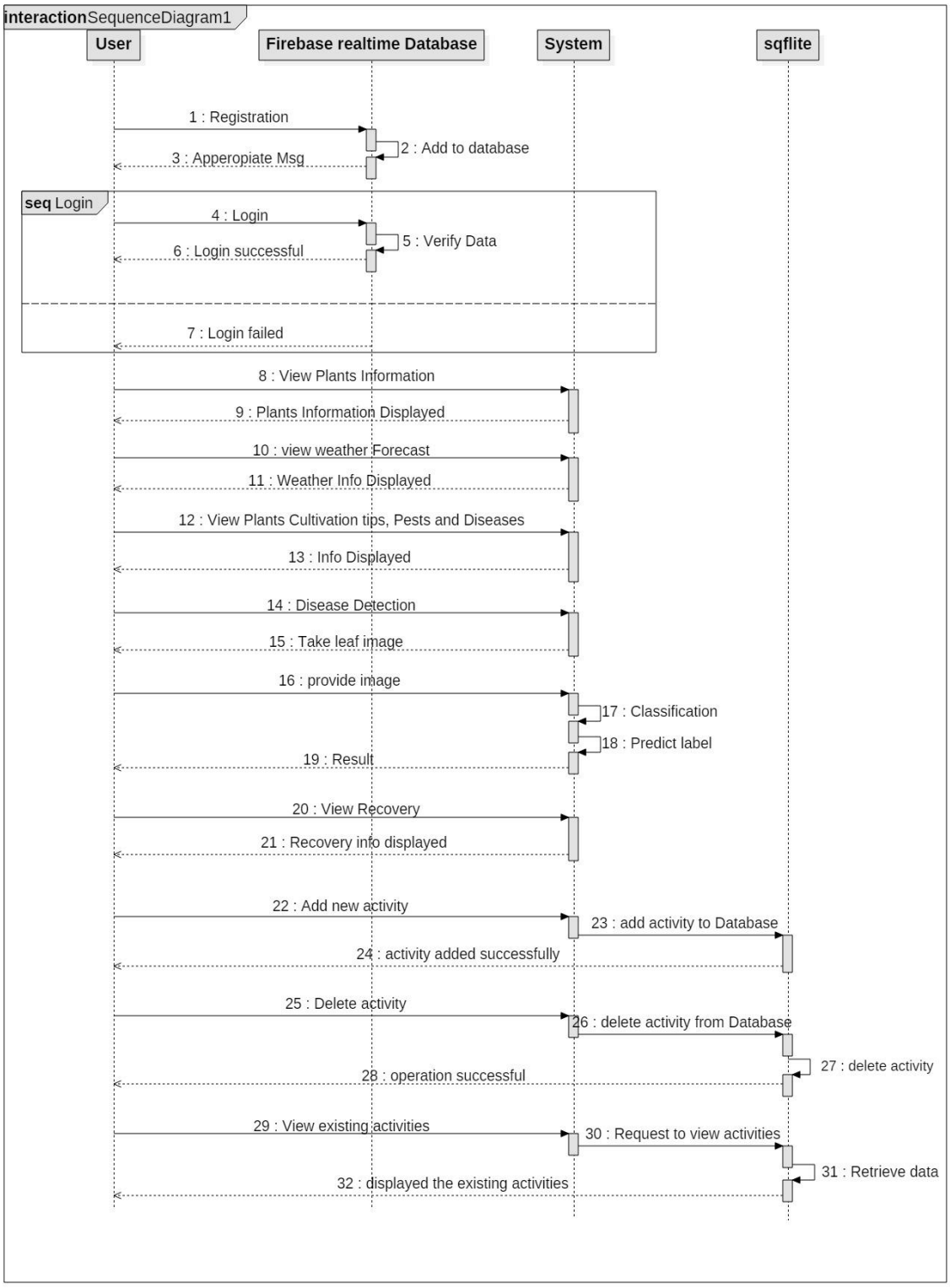

Figure 5: Sequence Diagram

4.4 Component Diagram

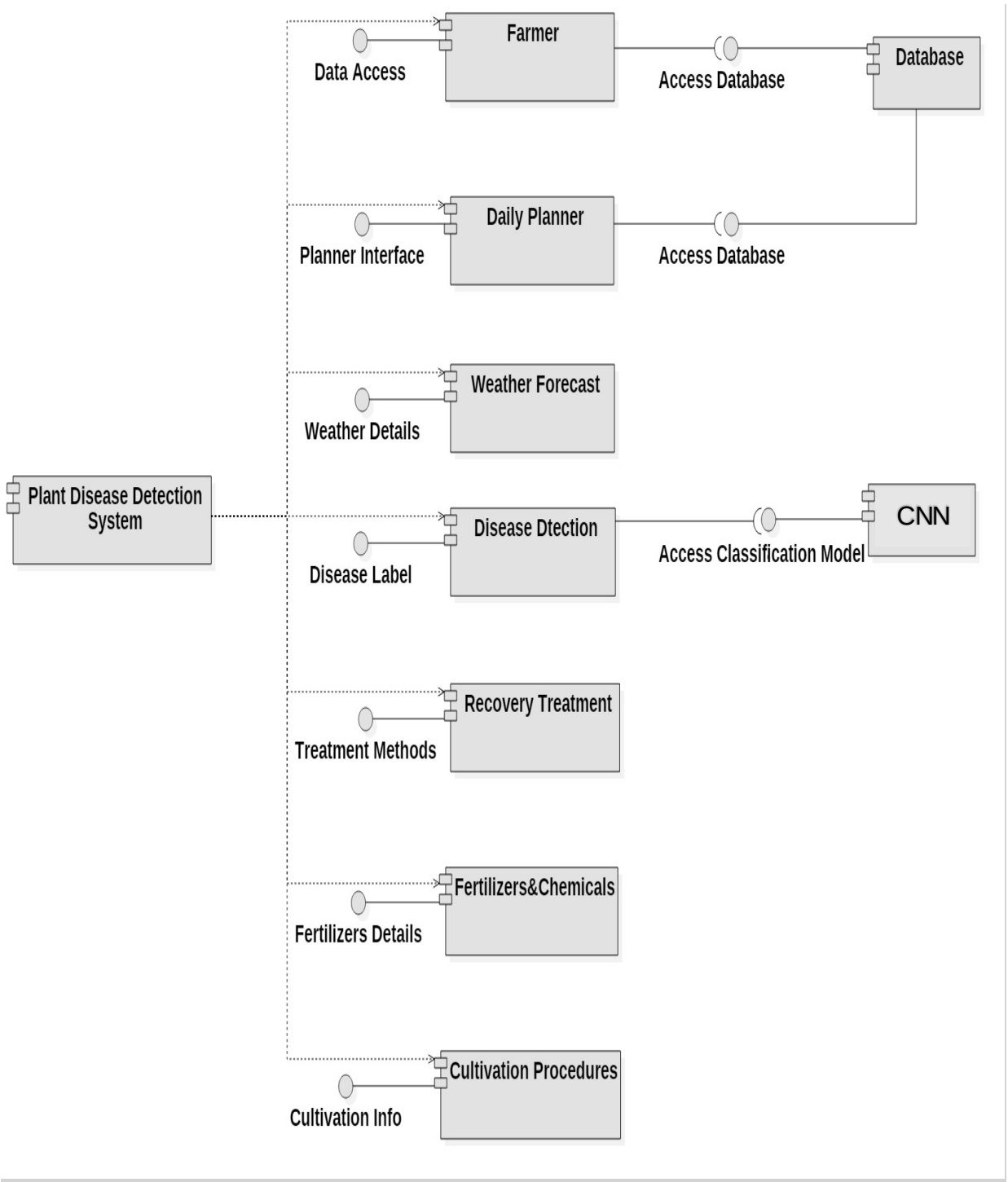

Figure 6: Component Diagram

4.5 Class Diagram

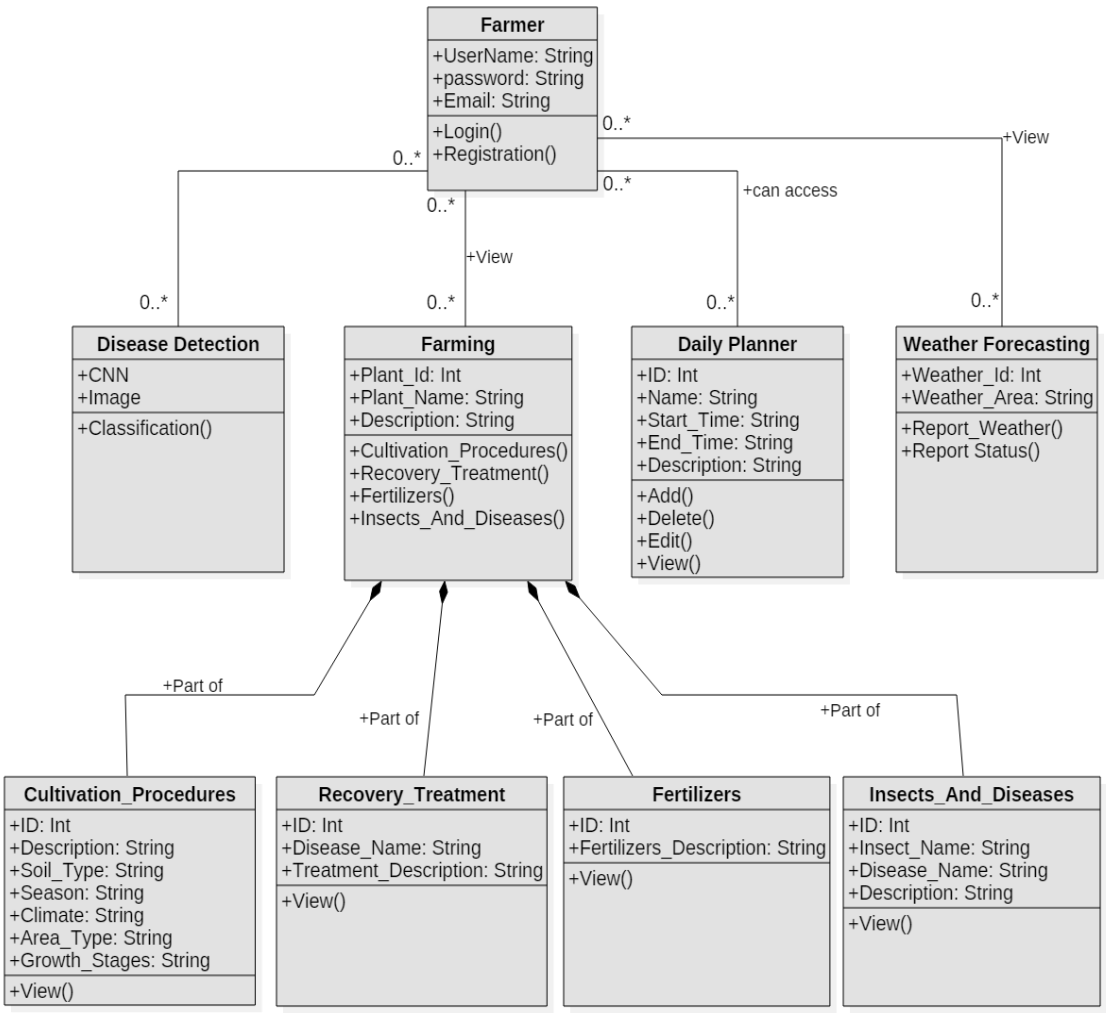

Figure 7: Class Diagram

Chapter 5

Implementation

5. Implementation

This chapter discusses all the Algorithms and user interfaces used to develop the project. Implementation is the most prominent phase of development. At this stage, all the ideas are transforming into a meaningful picture. This step is a significant and challenging step toward developing skills. An application is the result of the successful implementation of a project, various testing approaches, algorithms, and results.

5.1 Tools and Technologies

5.1.1 Flutter

Flutter is an open-source mobile application development framework created by Google. It helps you create high quality, fast, and beautiful iOS, Android, and web apps – all from a single codebase. We are using Flutter for building our mobile application.

5.1.2 Jupyter Notebook

Jupyter is a free tool that can used with dozens of programming languages. It is an open-source, open-standard, and interactive web tool. We are implementing our Ml algorithm in Jupyter Notebook using Python as a programming language.

5.1.3 Programming language

We are using two different languages for creating our application. For front-end and back-end development, we used Dart. For the implementation of the Ml algorithm, we used Python as a programming language, as it is the most powerful, readable, and scalable language.

5.1.4 Firebase DBMS

We use Firebase Real-time Database for DBMS used in login verification and keeping records of the user accounts. Firebase is Cloud Firestore that enables us to save, sync, and query app records at a global scale.

5.1.5 Star UML

Star UML is a software tool used for modeling software or application. We have used this tool to create ERD, DFD, Activity Diagrams, Use cases, and many more.

5.2 Dataset

We are using a PlantVillage dataset from Kaggle that consists of more than 22000 images for the most common 12 disease categories in 3 vegetable plant species, including potato, pepper, and tomato. We also added one folder of invalid pictures that contains the images of different things like human, furniture electronic object to deal with the other images. All the images used

are RGB images and are in (.JPG) format. All the images are in separate folders corresponding to their disease and plant type. The dataset contains the following diseases:

- Pepper bell Bacterial spot
- Potato Early blight
- Potato Late blight
- Tomato Bacterial spot
- Tomato Early blight
- Tomato Late blight
- Tomato Leaf Mold,
- Tomato Septoria leaf spot
- Tomato Spider mites Two-spotted spider mites
- Tomato Target Spot
- Tomato Yellow Leaf Curl Virus
- Tomato mosaic virus
- Invalid_Picture

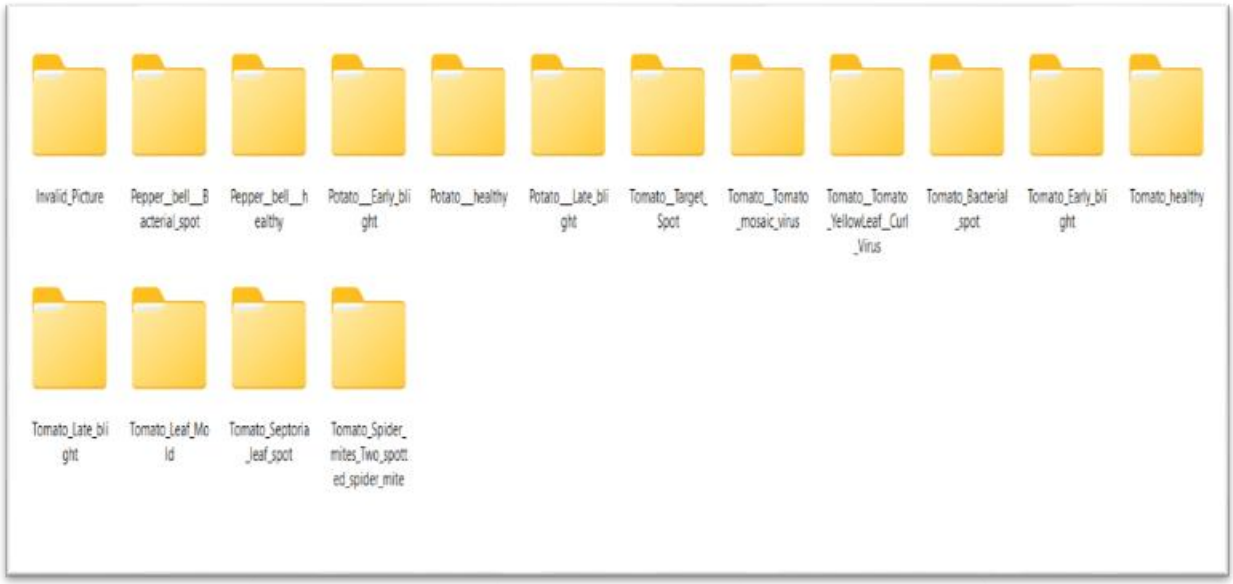

Figure 8: Dataset Folders

Table 19: Dataset Total Images

Disease Name	Number Of images
Pepper_bell_Bacterial_spot	992
Pepper__bell___healthy	1473
Potato_Early_blight	1000
Potato_healthy	152
Potato_Late_blight	1000
Tomato_Target_Spot	1404
Tomato_Tomato_mosaic_virus	337
Tomato_Tomato_YellowLeaf_Curl_Virus	3209
Tomato_Bacterial_spot	2127
Tomato_Early_blight	1000
Tomato_healthy	1591
Tomato_Late_blight	1909
Tomato_Leaf_Mold	952
Tomato_Septoria_leaf_spot	1771
Tomato_Spider_mites_Two_spotted_spider_mite	1676
Invalid_Picture	738

5.3 Algorithm

For the disease detection module, we need an ML model. Multiple algorithms like pre-trained VGG19, VGG16, YOLO, Resnet 50, CNN, and many other models can be used for this purpose. We implemented two models VGG19 and CNN and use the CNN model as it gives the best accuracy.

Table 20: Model accuracy and loss

Model	**Training Accuracy**	**Testing Accuracy**	**Training Loss**	**Testing Loss**
VGG 19	0.8379	0.7959	0.4806	0.6813
CNN	0.9777	0.9599999	0.0717	0.2211

5.3.1 CNN Model

We develop the CNN algorithm for the disease detection of plants. Our model is trained on the PlantVillage dataset. Based on features, the model compares the input image with trained data and displays the result.

5.3.1.1 CNN Model Architecture

The designed CNN model has 15 layers, including 6 convolutional layers, 6 max-pooling layers, 1 flatten layer, and 2 dense layers.

5.3.1.1.1 Input Layer (resize and rescale)

This is not a layer but a preprocessing step that resizes and rescales the input images to a size (256, 256) and range (1.0/255).

5.3.1.1.2 Convolutional Layers

This layer applies a 2D convolution operation to the input. The layer learns a set of filters that are convolved with the input to produce a set of feature maps. The number of filters and the filter size in each layer are mentioned below.

- The first Conv2D layer has 32 filters.
- The second Conv2D layer has 64 filters.
- The third Conv2D layer has 64 filters.
- The fourth Conv2D layer has 64 filters.
- The fifth Conv2D layer has 64 filters.
- The sixth Conv2D layer has 64 filters.

They all use an activation function called "relu" and filter size is (3, 3).

5.3.1.1.3 Max Pooling Layers

To reduce the dimension of the vegetable plant images and preserve their features, we use a Max Pooling layer, which helps with downsampling. We use 6 Max-Pooling layers in the model.

5.3.1.1.4 Flatten Layer

This layer flattens the output of the previous layer into a 1D array. The flattened array is then passed to the dense layers for final classification.

5.3.1.1.5 Dense layers

This layer performs the final classification task. The number of units is 16 as total classes in the dataset and the activation function is softmax

5.3.1.2 Implementation steps

5.3.1.2.1 Dataset

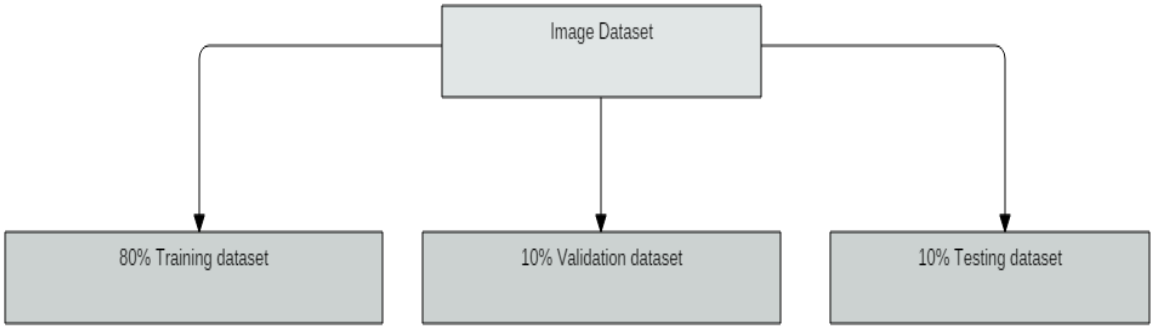

Figure 9: Dataset partitioning

5.3.1.2.2 CNN Model

CNN network uses an image and then the image and filters combined that produce features. Classification of disease labels is done by CNN that is using convolutional layer, Max pooling, ReLU activation function, and flatten and fully connected layers. The number of epochs is 50.

5.3.1.2.3 Training

In the training phase, we used 16000 images. Our CNN model trained over 80% of images in the train folder.

5.3.1.2.4 Epochs

```
400/400 [==============================] - 405s 1s/step - loss: 0.0849 - accuracy: 0.9708 - val_loss: 0.2422 - val_accuracy: 0.9563
Epoch 36/50
400/400 [==============================] - 406s 1s/step - loss: 0.0181 - accuracy: 0.9939 - val_loss: 0.1733 - val_accuracy: 0.9638
Epoch 37/50
400/400 [==============================] - 407s 1s/step - loss: 0.0907 - accuracy: 0.9730 - val_loss: 0.2486 - val_accuracy: 0.9375
Epoch 38/50
400/400 [==============================] - 511s 1s/step - loss: 0.0740 - accuracy: 0.9756 - val_loss: 0.2159 - val_accuracy: 0.9525
Epoch 39/50
400/400 [==============================] - 582s 1s/step - loss: 0.0666 - accuracy: 0.9780 - val_loss: 0.1707 - val_accuracy: 0.9600
Epoch 40/50
400/400 [==============================] - 571s 1s/step - loss: 0.0500 - accuracy: 0.9842 - val_loss: 0.2711 - val_accuracy: 0.9275
Epoch 41/50
400/400 [==============================] - 423s 1s/step - loss: 0.0921 - accuracy: 0.9733 - val_loss: 0.2963 - val_accuracy: 0.9400
Epoch 42/50
400/400 [==============================] - 420s 1s/step - loss: 0.0933 - accuracy: 0.9725 - val_loss: 0.2238 - val_accuracy: 0.9550
Epoch 43/50
400/400 [==============================] - 419s 1s/step - loss: 0.0315 - accuracy: 0.9909 - val_loss: 0.2463 - val_accuracy: 0.9463
Epoch 44/50
400/400 [==============================] - 418s 1s/step - loss: 0.0104 - accuracy: 0.9967 - val_loss: 0.2445 - val_accuracy: 0.9488
Epoch 45/50
400/400 [==============================] - 421s 1s/step - loss: 0.1076 - accuracy: 0.9677 - val_loss: 0.2042 - val_accuracy: 0.9550
Epoch 46/50
400/400 [==============================] - 421s 1s/step - loss: 0.0487 - accuracy: 0.9831 - val_loss: 0.2697 - val_accuracy: 0.9550
Epoch 47/50
400/400 [==============================] - 474s 1s/step - loss: 0.0494 - accuracy: 0.9844 - val_loss: 0.3320 - val_accuracy: 0.9413
Epoch 48/50
400/400 [==============================] - 437s 1s/step - loss: 0.0694 - accuracy: 0.9784 - val_loss: 0.2300 - val_accuracy: 0.9450
Epoch 49/50
400/400 [==============================] - 428s 1s/step - loss: 0.0570 - accuracy: 0.9814 - val_loss: 0.2798 - val_accuracy: 0.9475
Epoch 50/50
400/400 [==============================] - 437s 1s/step - loss: 0.0717 - accuracy: 0.9777 - val_loss: 0.2936 - val_accuracy: 0.9525
```

Figure 10: epochs

5.3.1.2.5 Testing

In the testing phase, we used 10% images of dataset. The testing phase is done when an image is browsed from the testing folder and that image is matched with the pre-trained model and predicts the class of diseases label.

5.3.1.2.6 Output

The predicted disease labels along with the confidence rate will be shown individually.

5.3.1.3 Graphical Representation

5.3.1.3.1 Accuracy Graph

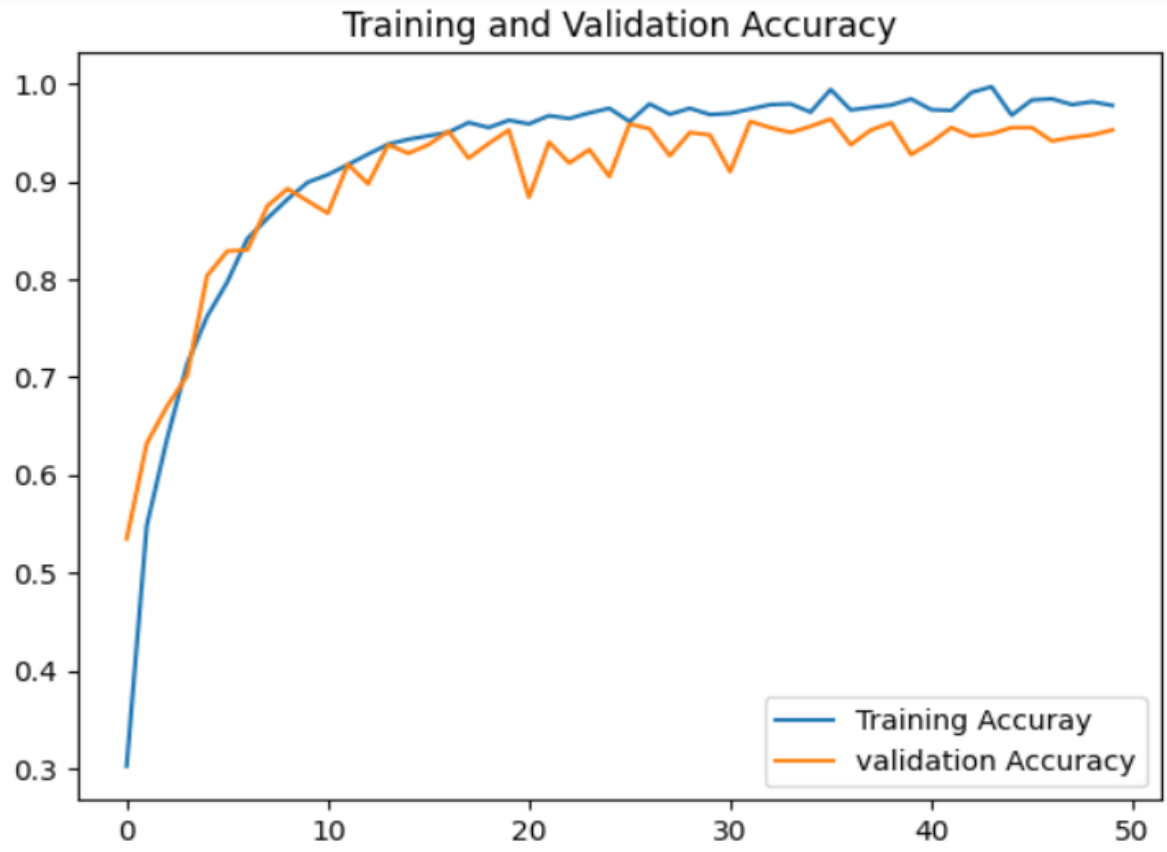

Figure 11: Training and validation Accuracy Graph

5.3.1.3.2 Loss Graph

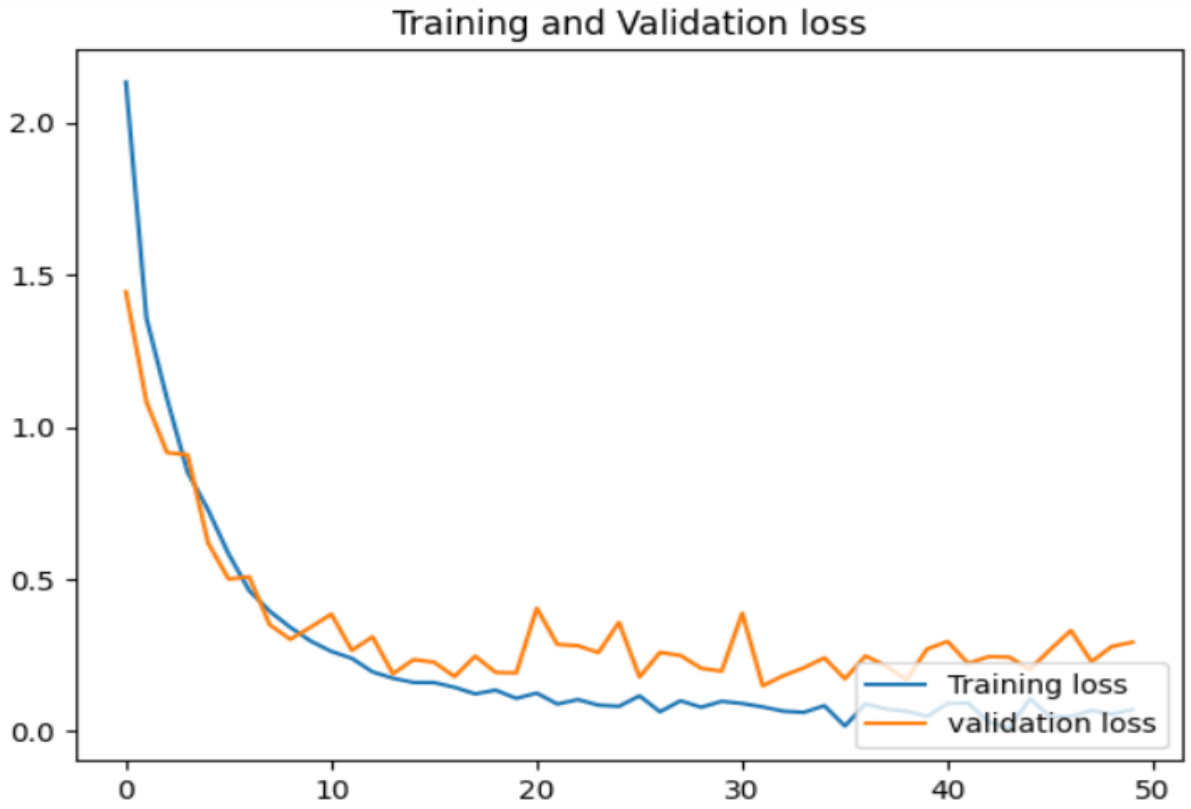

Figure 12: Training and Validation loss Graph

5.4 User Interface

UI is one of the exceptionally essential and commendable parts of the task. UI is the fragment that pulls in or involves the customers. Dynamically the interface would advance; progressively the customer will feel the interface is supportive and pleasing. The interface must be straightforward to use. We have tried to make it as much beneficial as we could. The interface of our application is designed in Flutter using Dart language. The working of each module and interface is shown below:

5.4.1 Dashboard Screen

Figure 13: Dashboard screen

5.4.2 Sign in

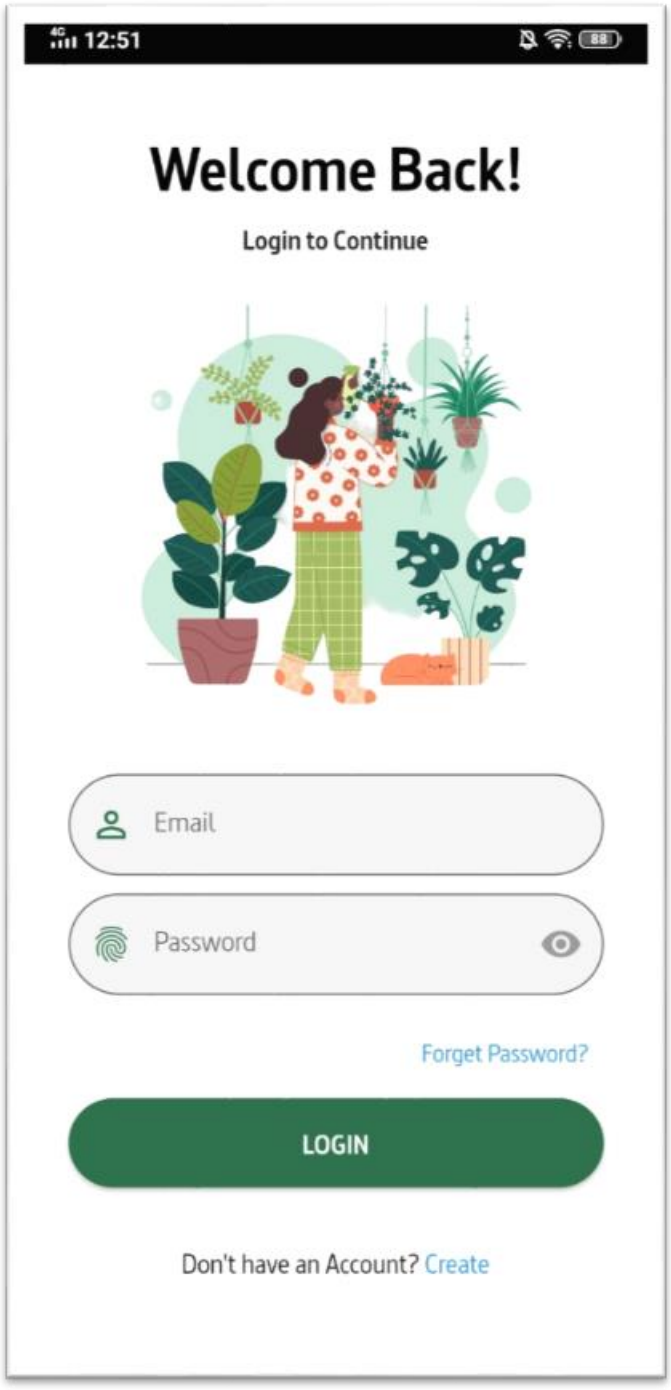

Figure 14: Sign in screen

5.4.3 Sign up

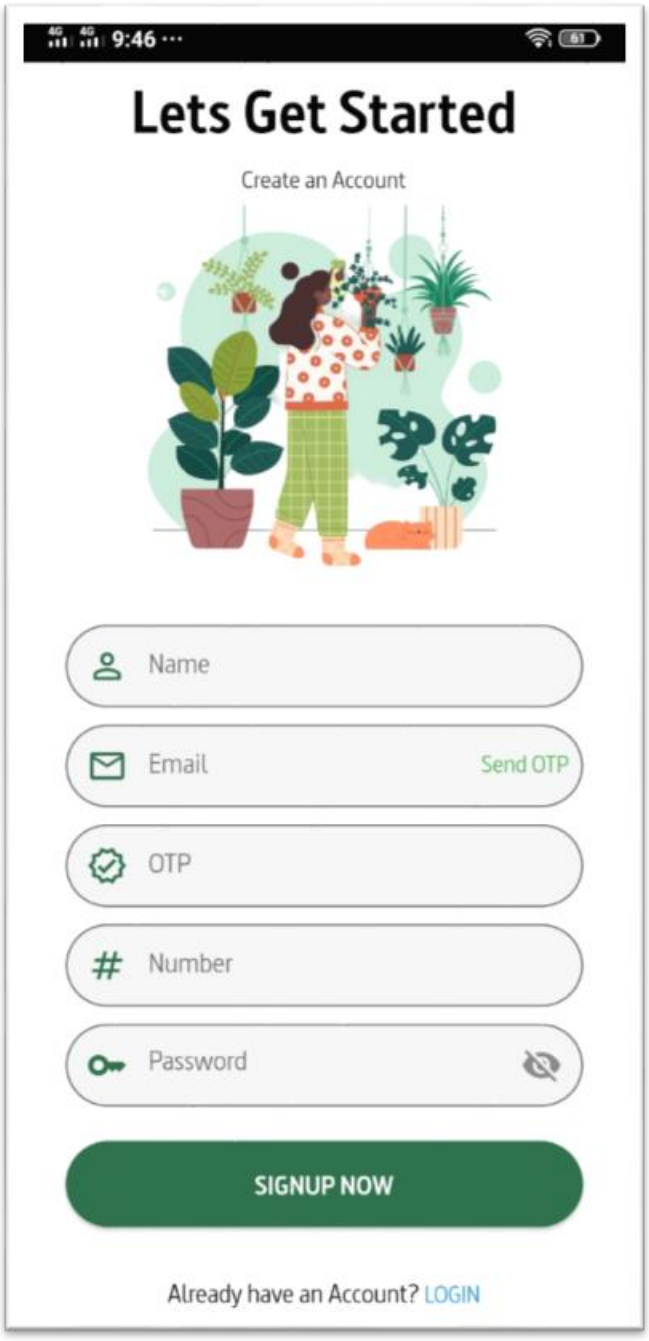

Figure 15: Sign up Screen

5.4.4 Home screen

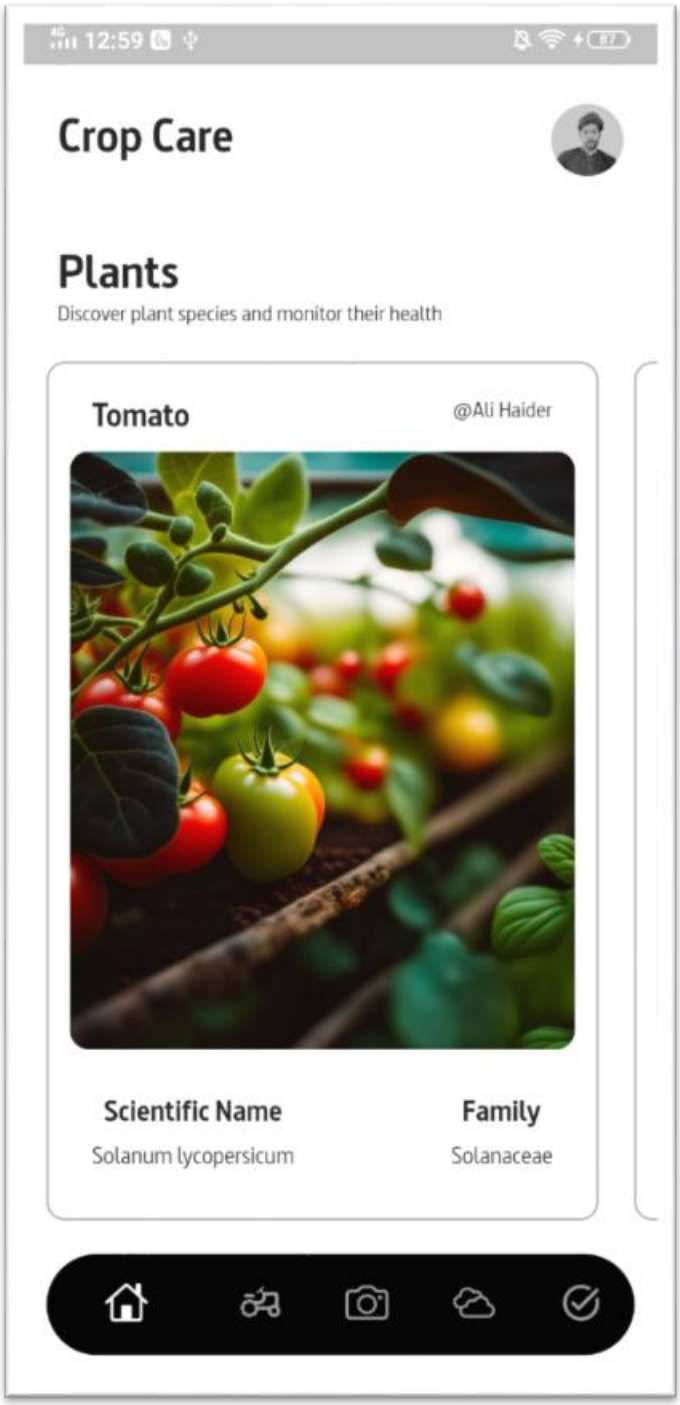

Figure 16: Home Screen

5.4.5 Plant detail screen

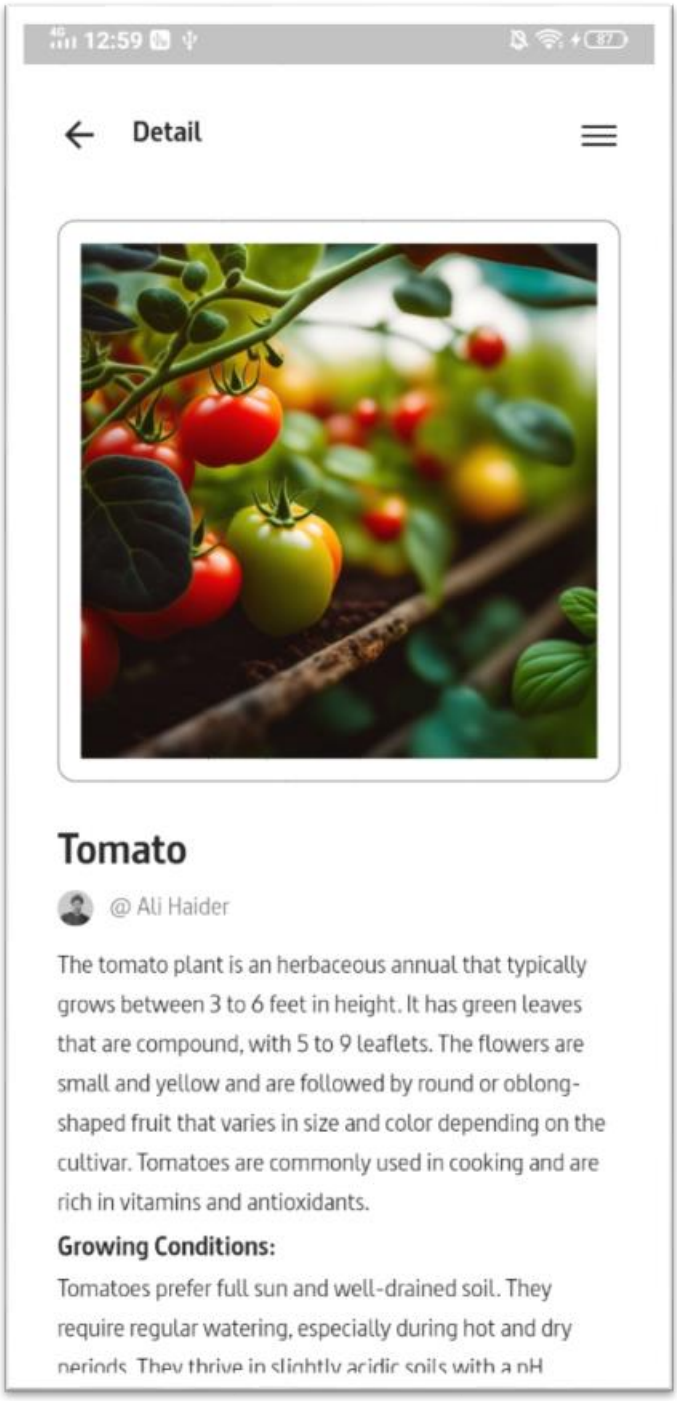

Figure 17: Plant Detail screen

5.4.6 Profile screen

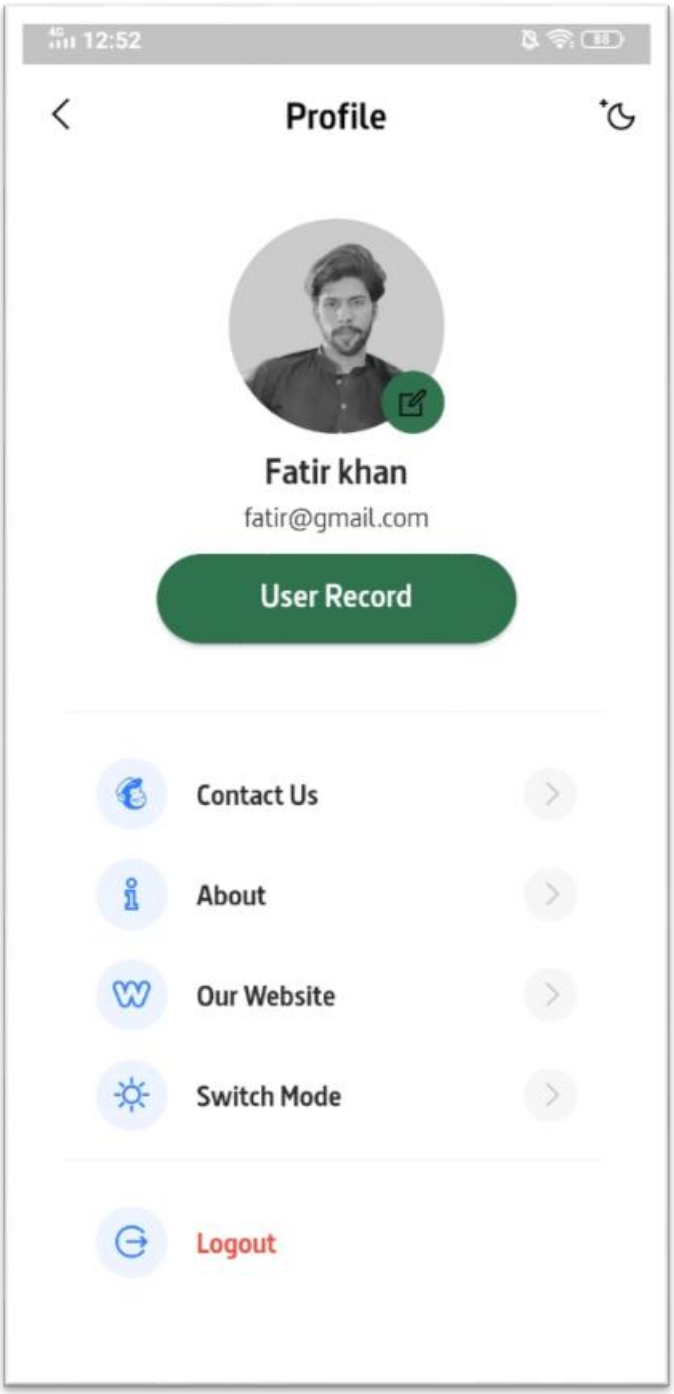

Figure 18: Profile screen

5.4.7 User Record Screen

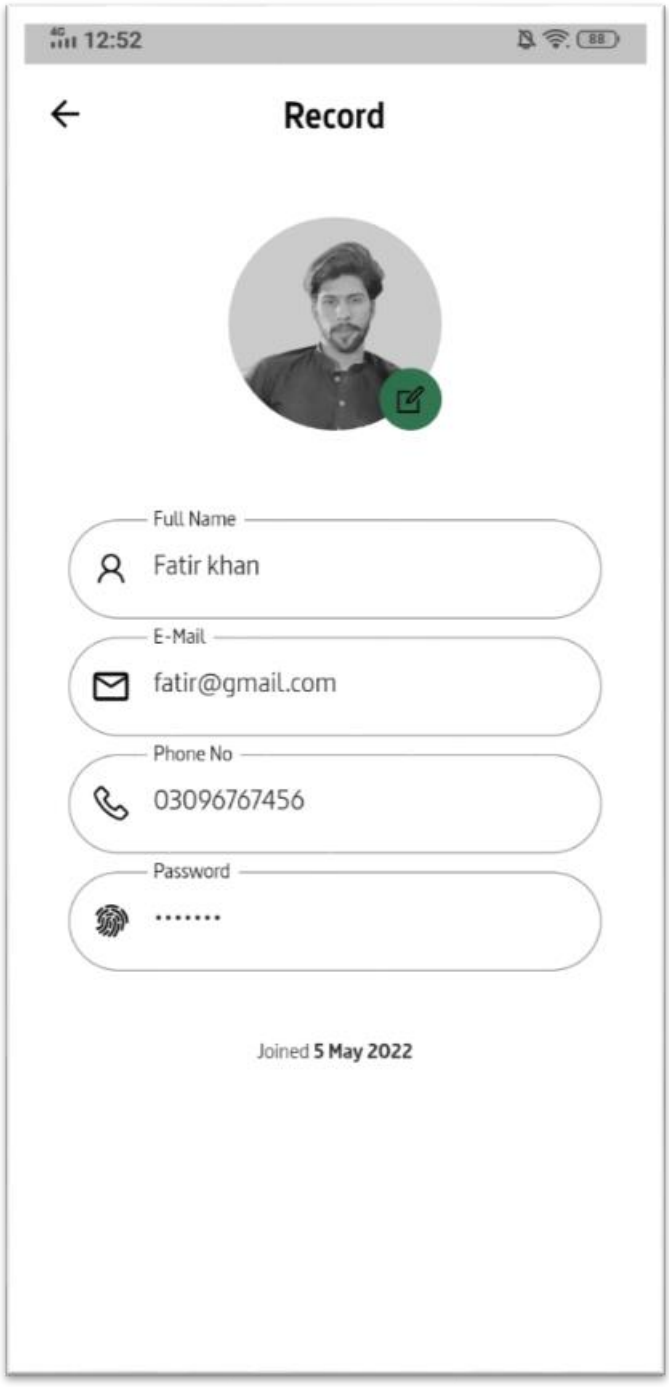

Figure 19: User Record screen

5.4.8 Weather forecast screen

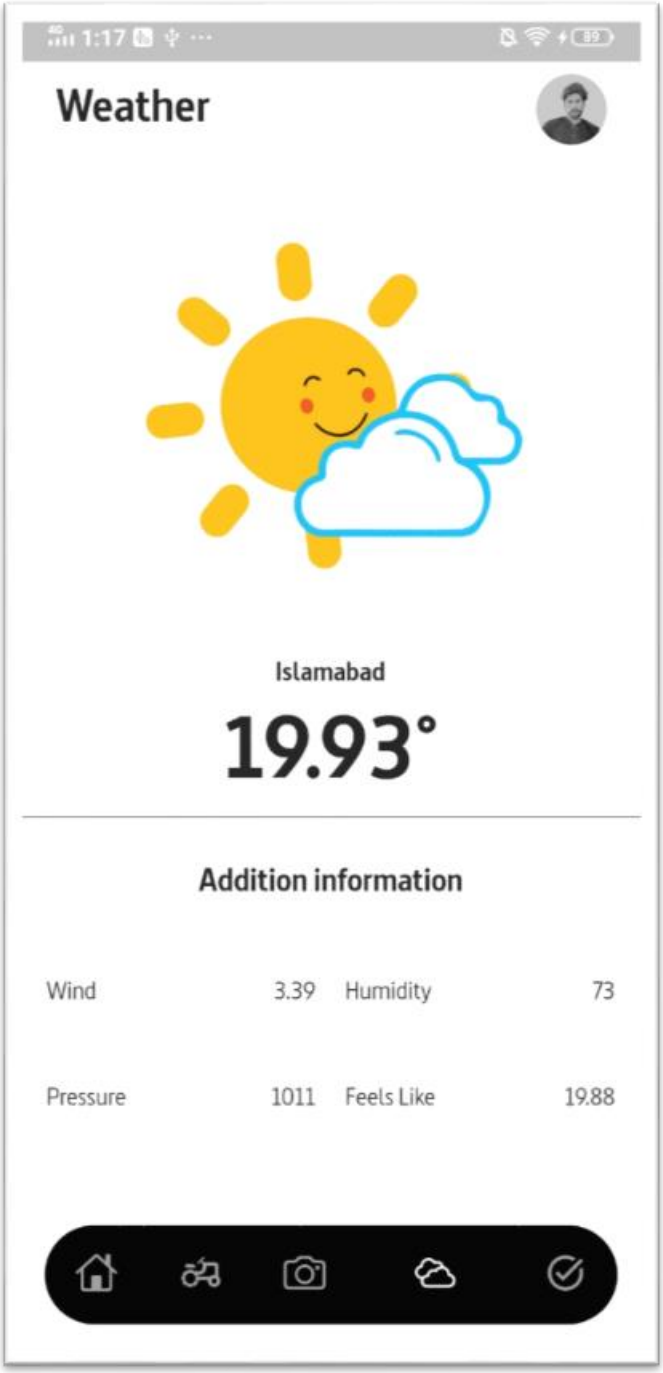

Figure 20: Weather Forecast Screen

5.4.9 Activity Planner screen

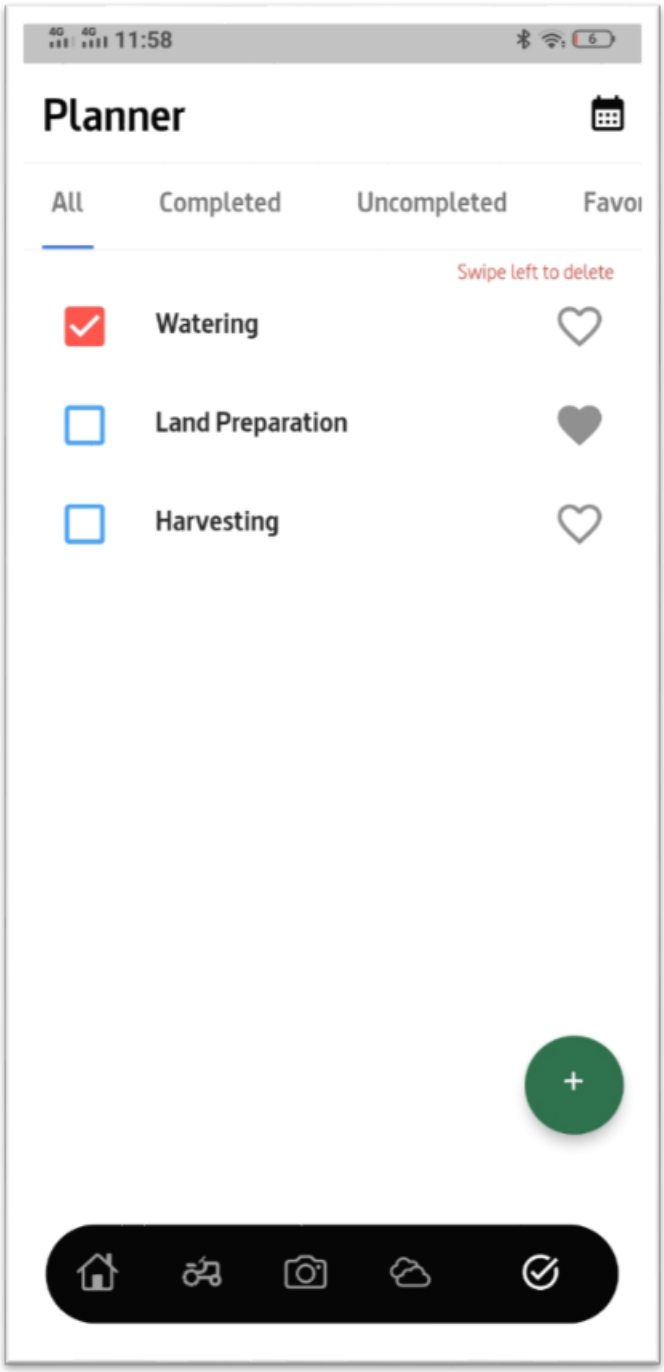

Figure 21: Activity Planner Screen

5.4.10 Add Task Screen

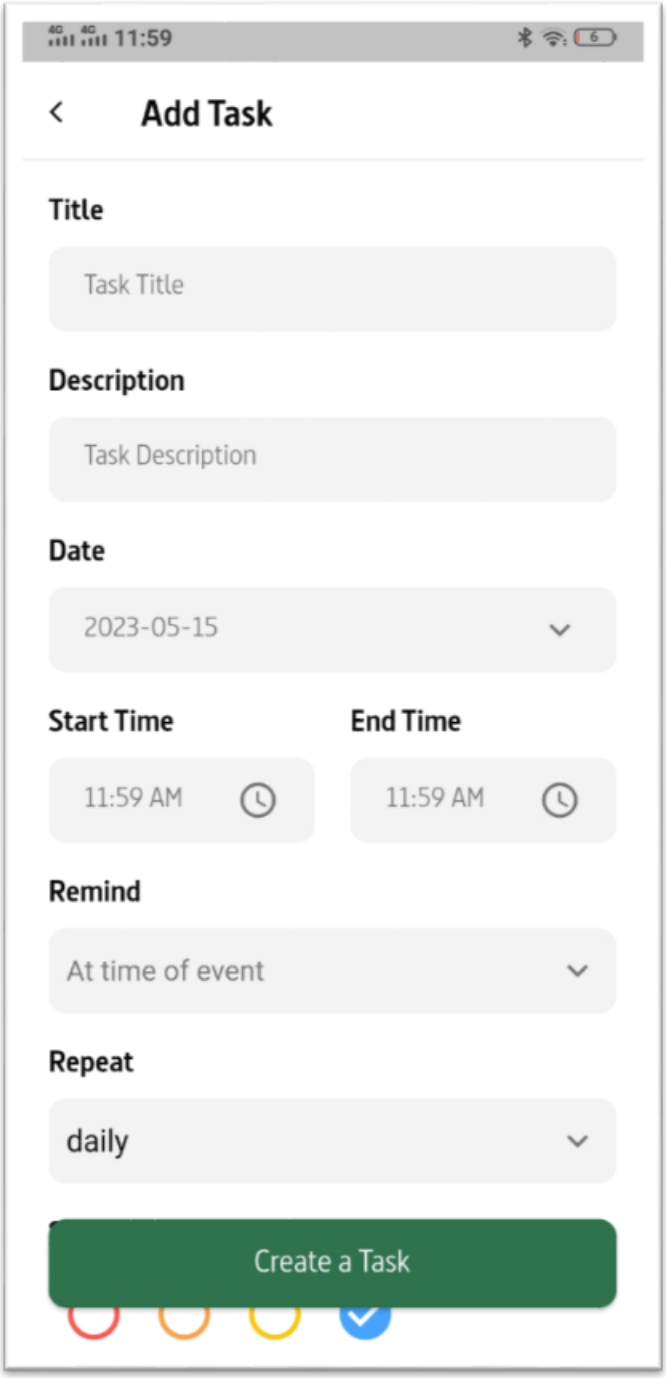

Figure 22: Add task screen

5.4.11 Favorite Task Screen

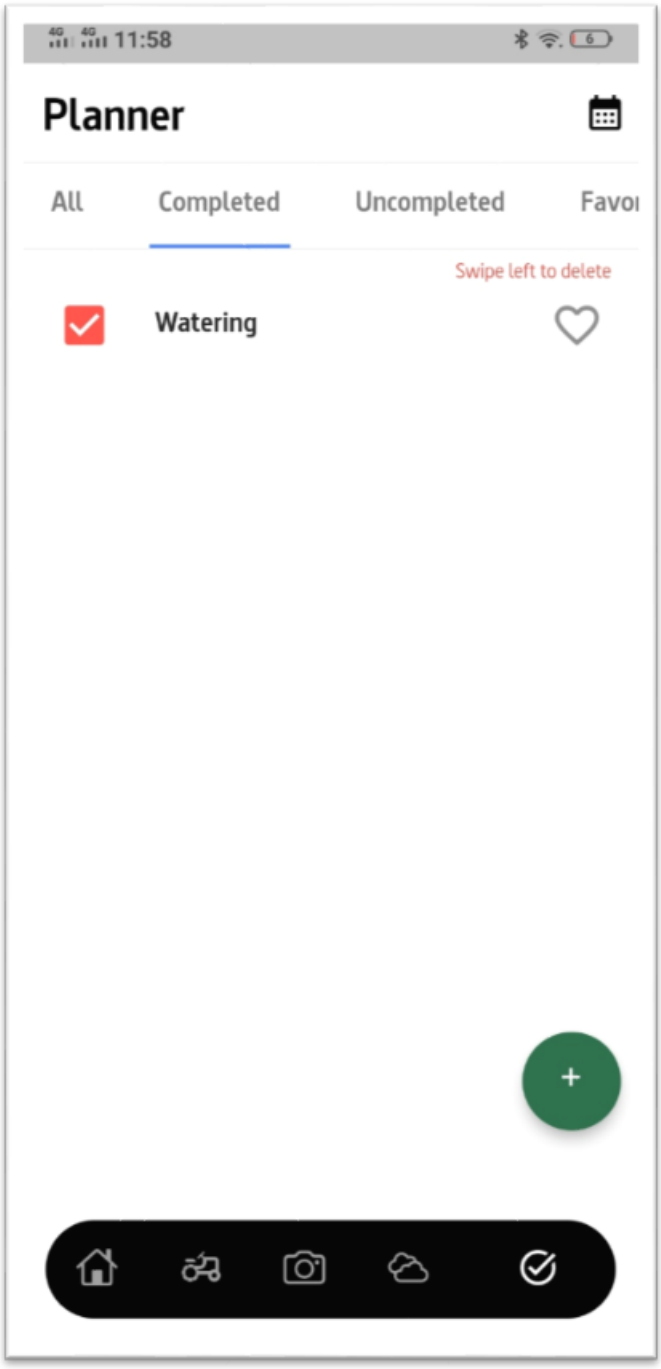

Figure 23: Favorite Task screen

5.4.12 Completed Task Screen

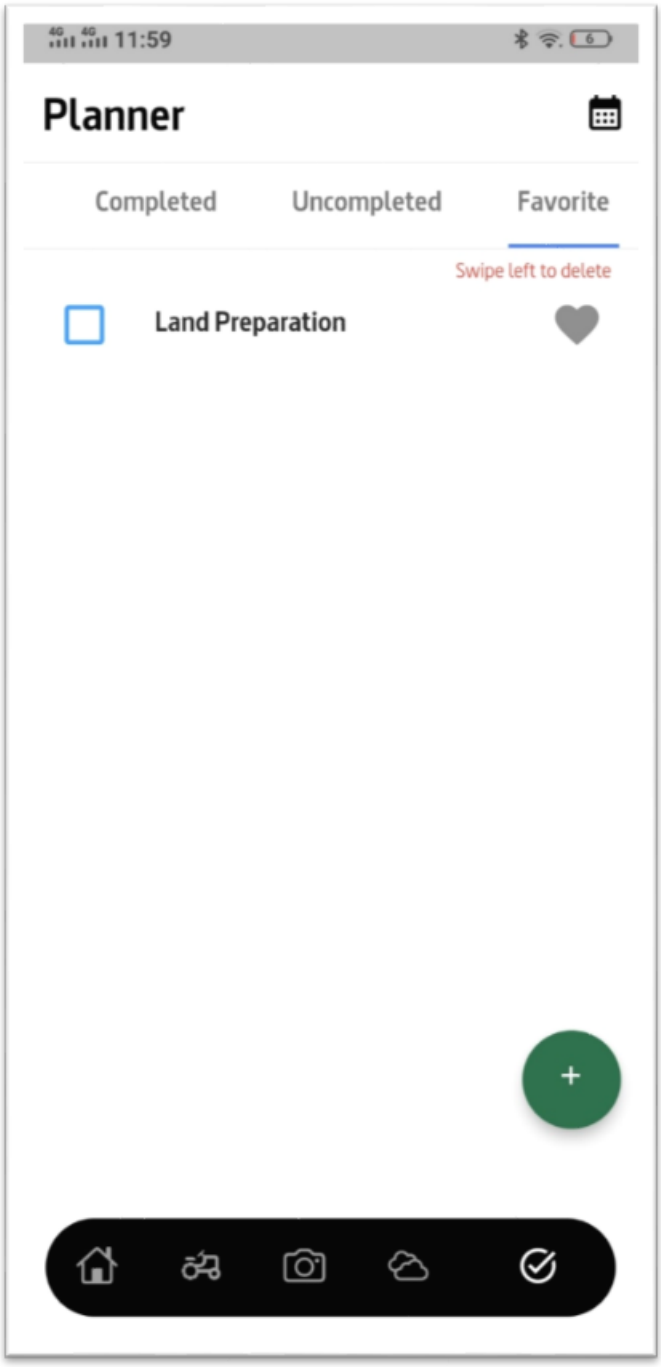

Figure 24: Completed Task Screen

5.4.13 Uncompleted Task Screen

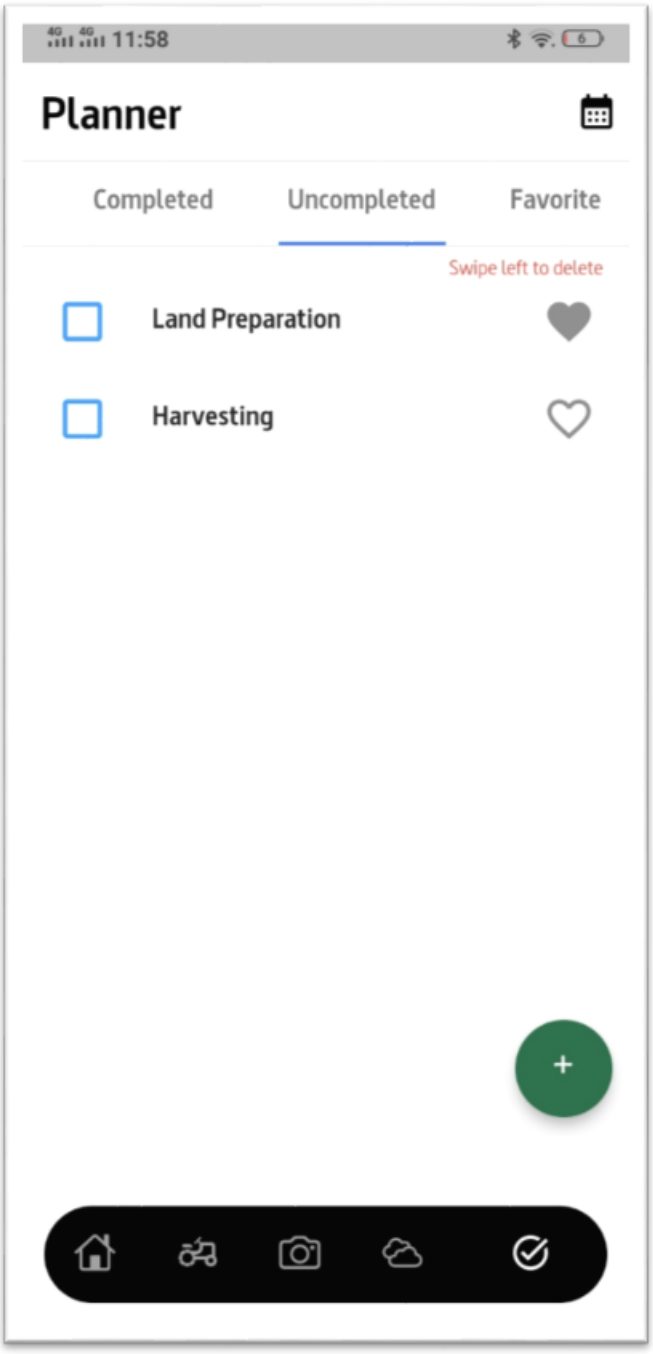

Figure 25: Uncompleted Task Screen

5.4.14 Schedule Screen

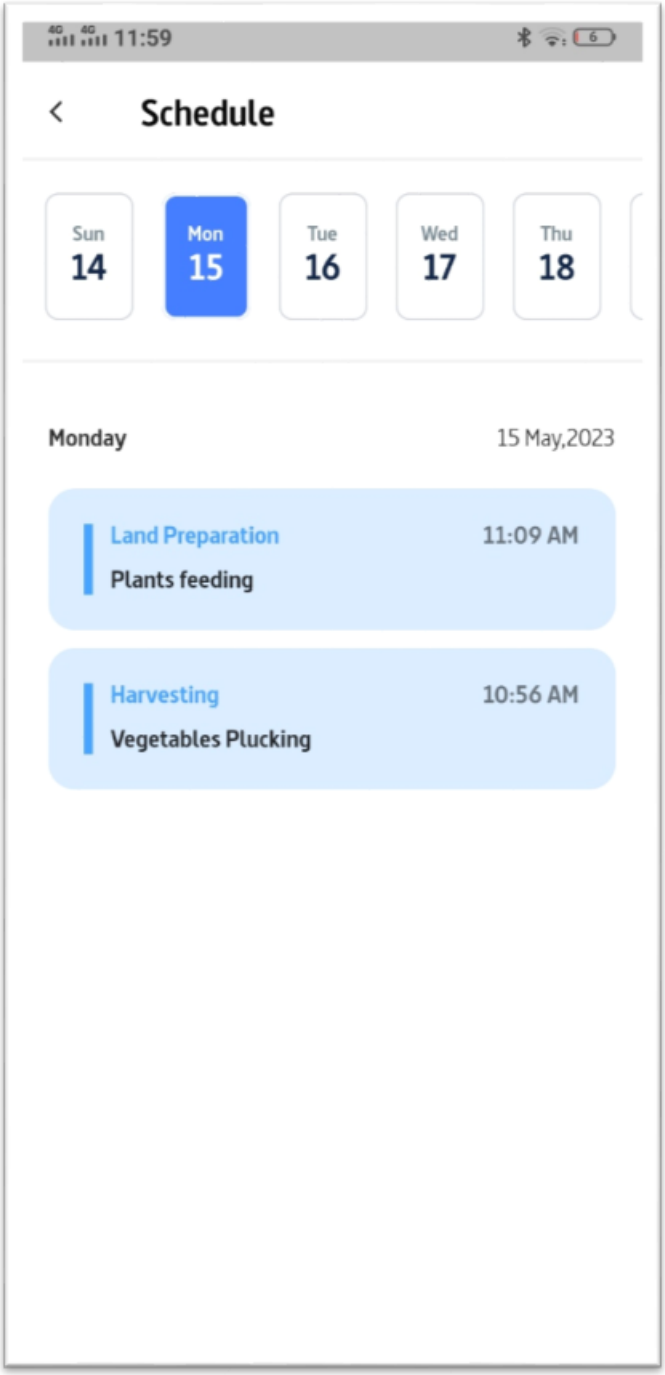

Figure 26: Schedule Task Screen

5.4.15 Farming Screen

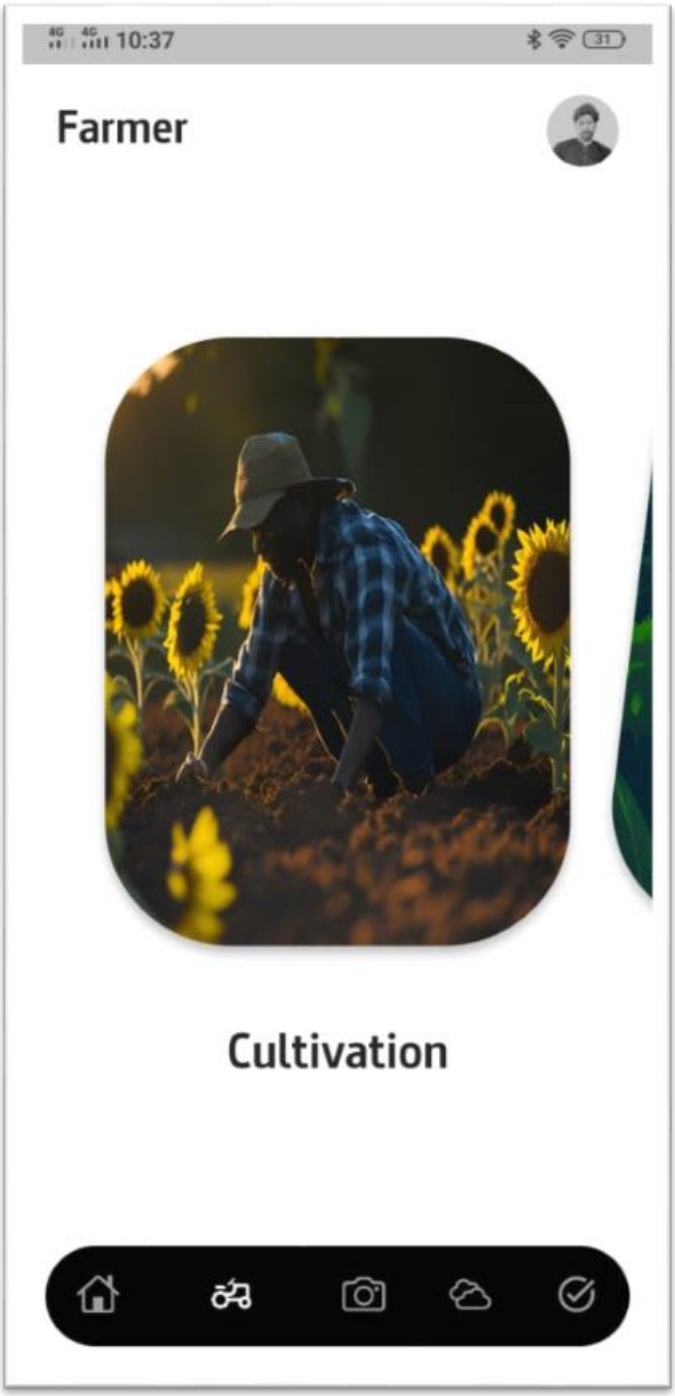

Figure 27: Farming Screen

5.4.16 Cultivation Plant list Screen

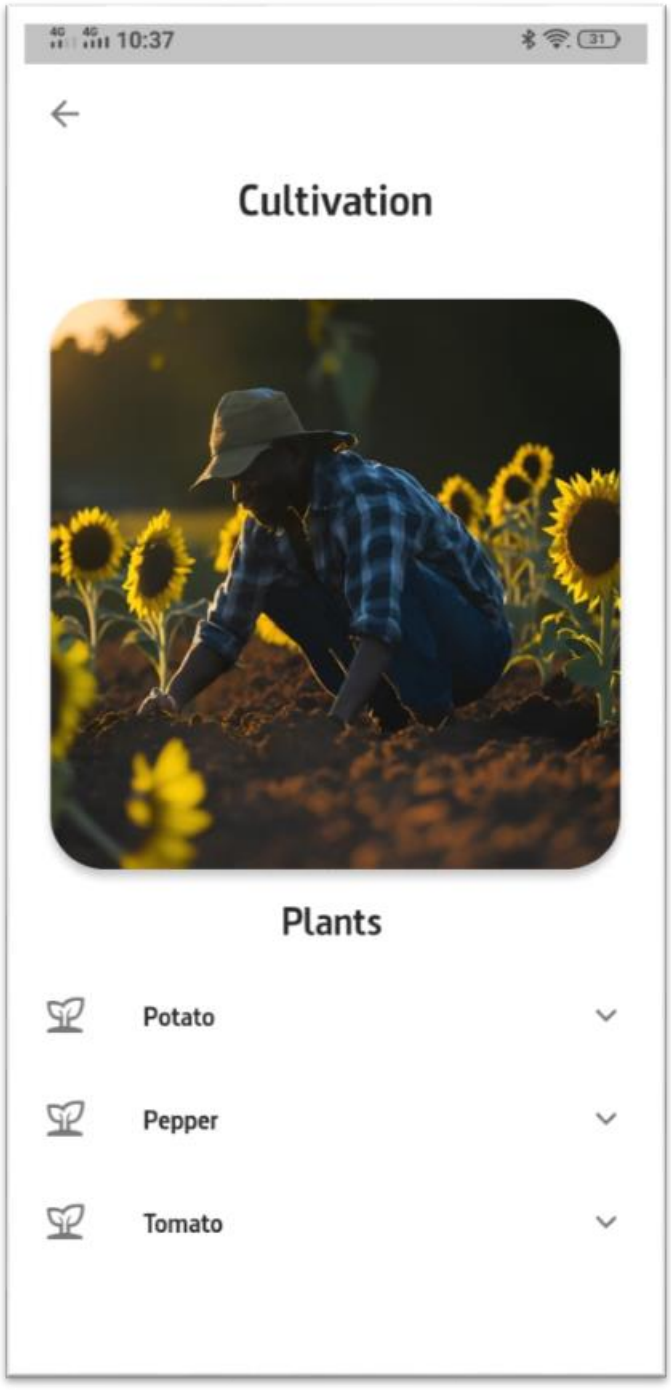

Figure 28: Cultivation Plant List screen

5.4.17 Plant Cultivation weeks list Screen

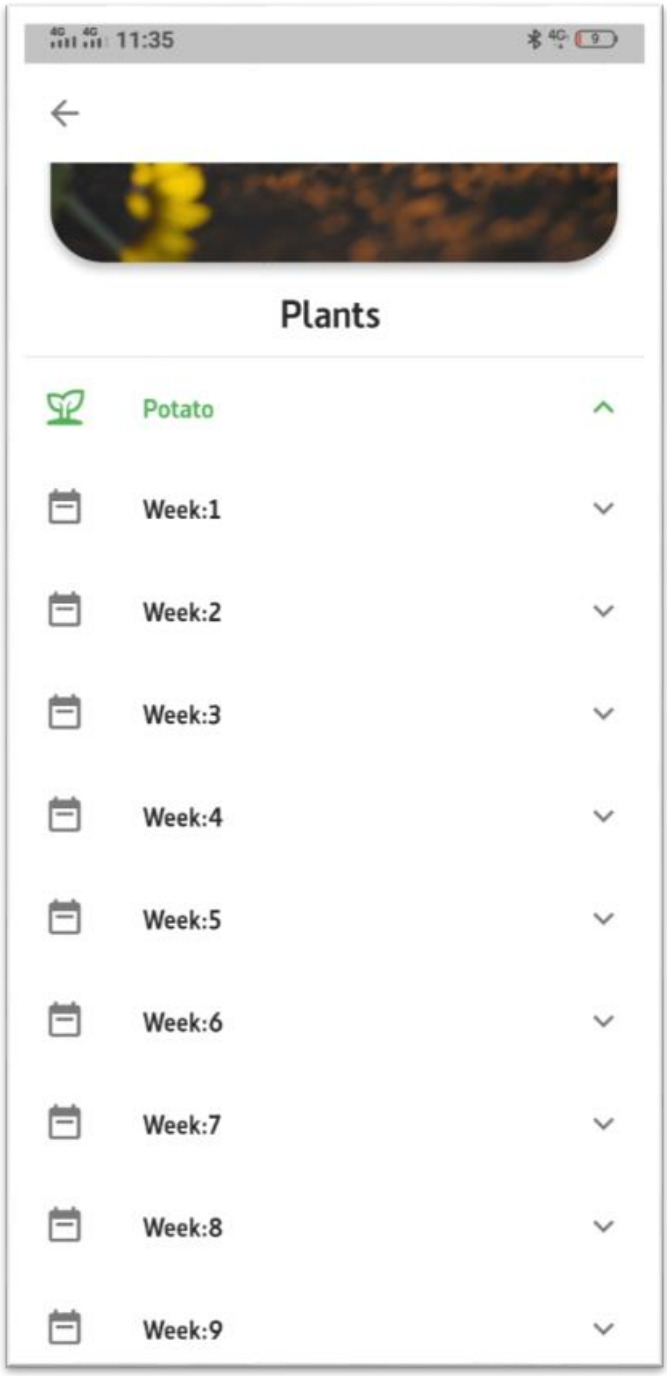

Figure 29: Plant cultivation weeks list screen

5.4.18 Cultivation Information Screen

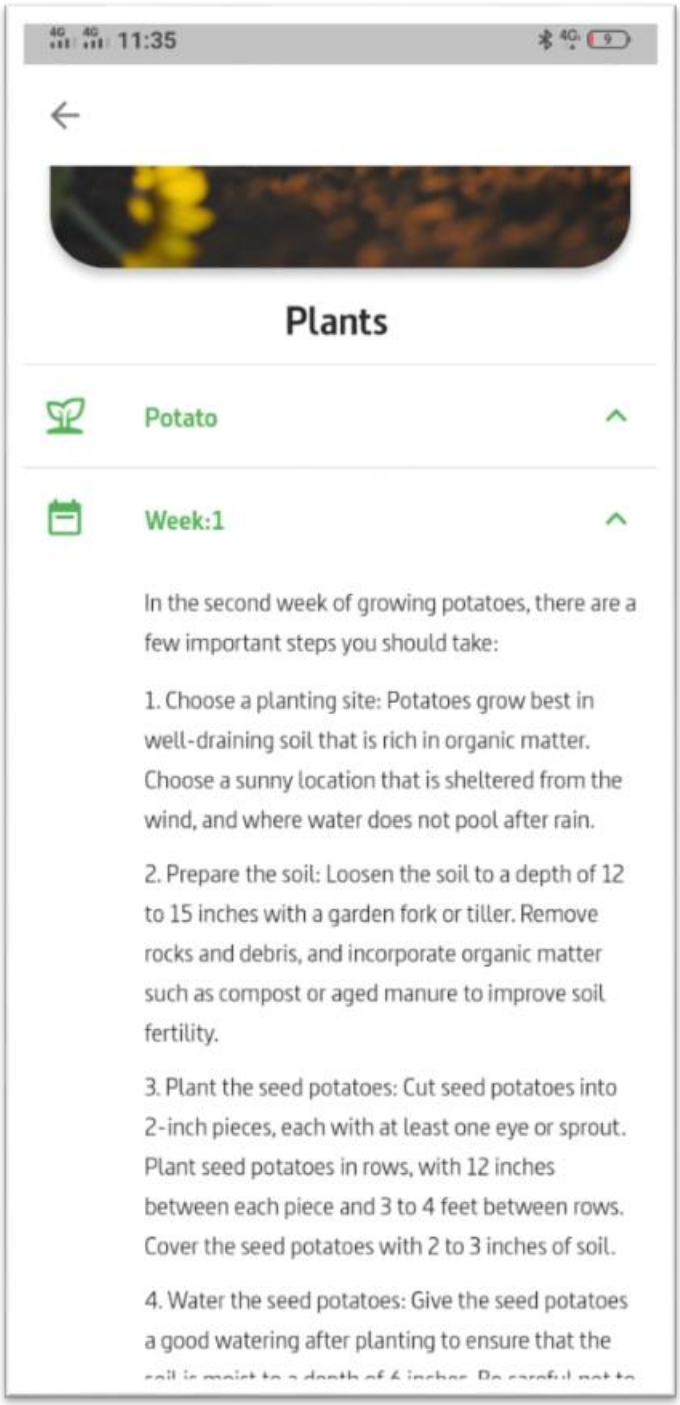

Figure 30: Cultivation Info screen

5.4.19 Plant Disease Screen

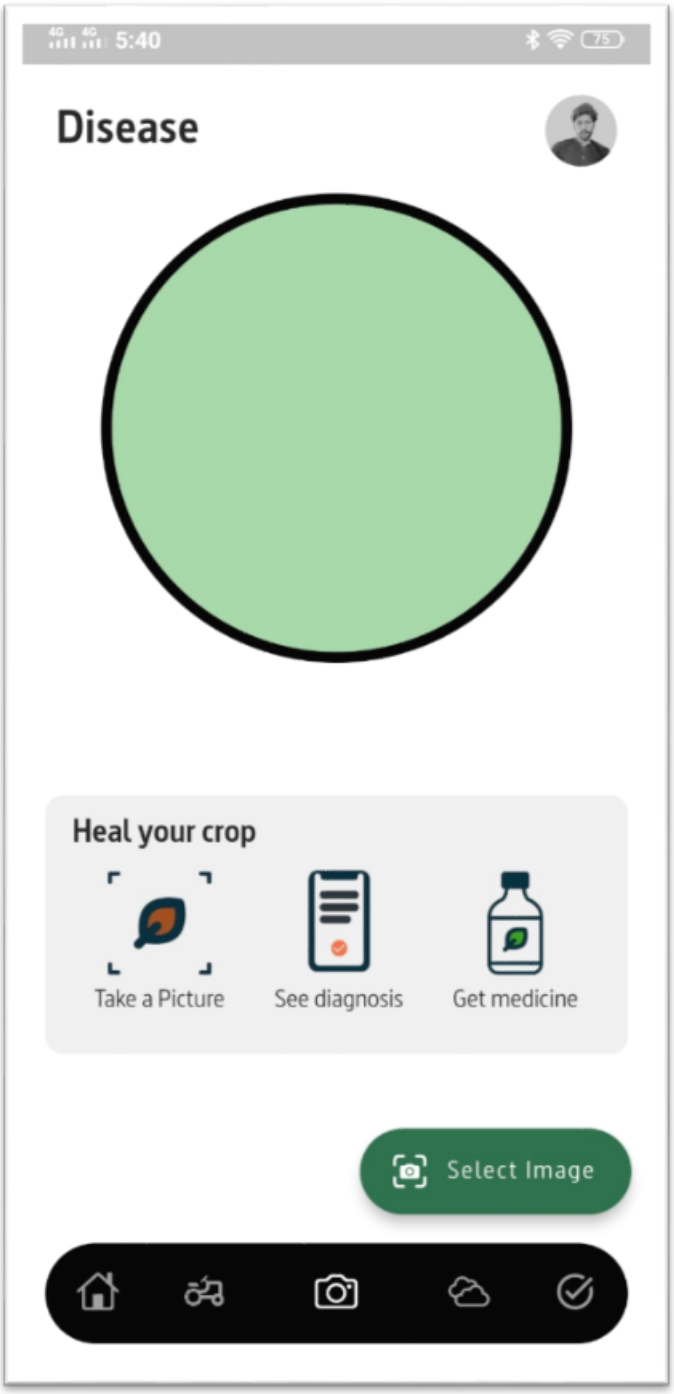

Figure 31: Disease Detection Screen

5.4.20 Disease Diagnose Screen

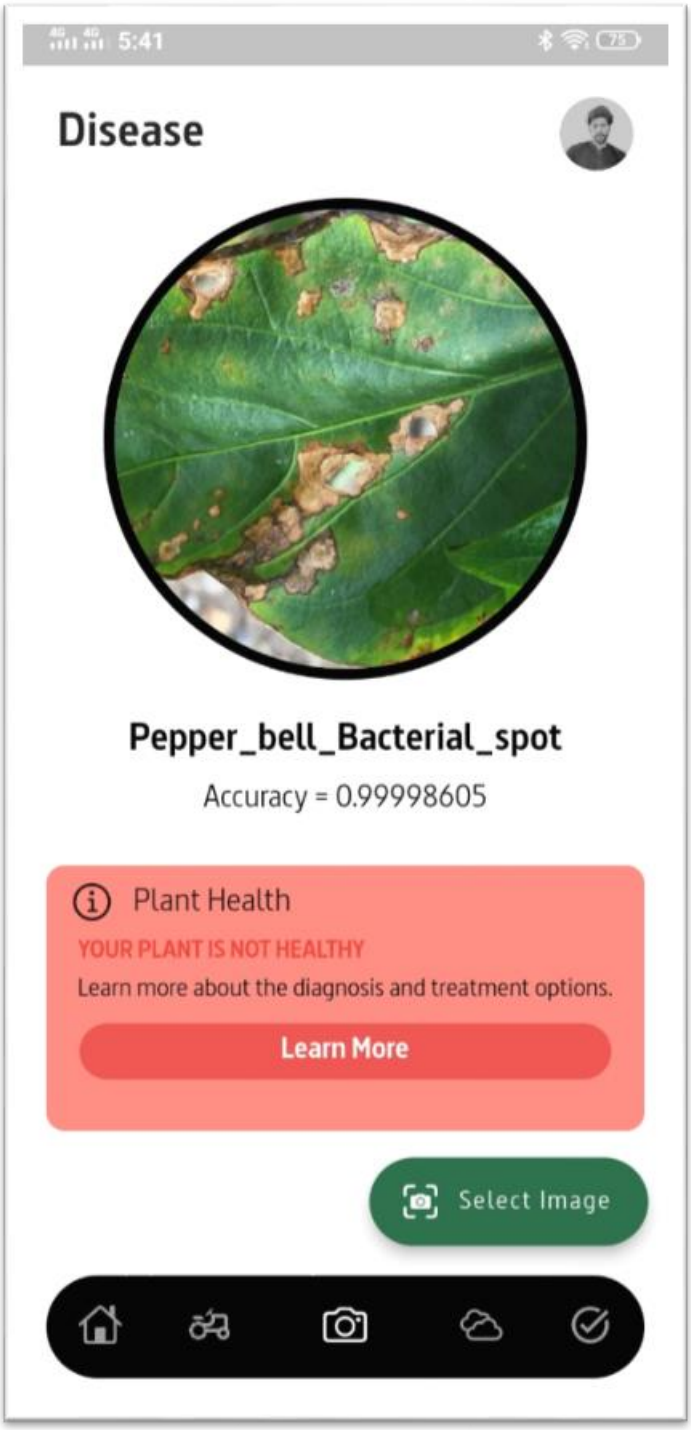

Figure 32: Disease Diagnose Screen

5.4.21 Healthy Diagnose screen

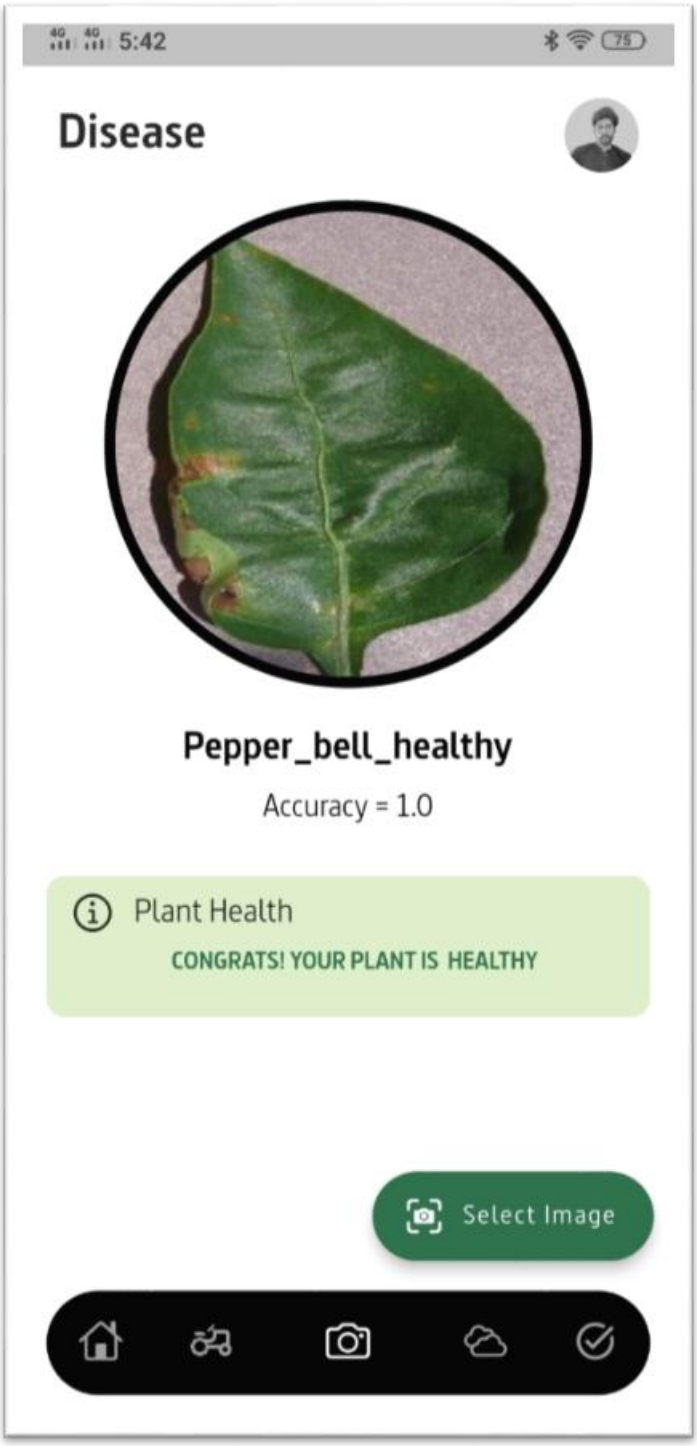

Figure 33: Healthy Plant Diagnose Screen

5.4.22 Invalid Picture Diagnose Screen

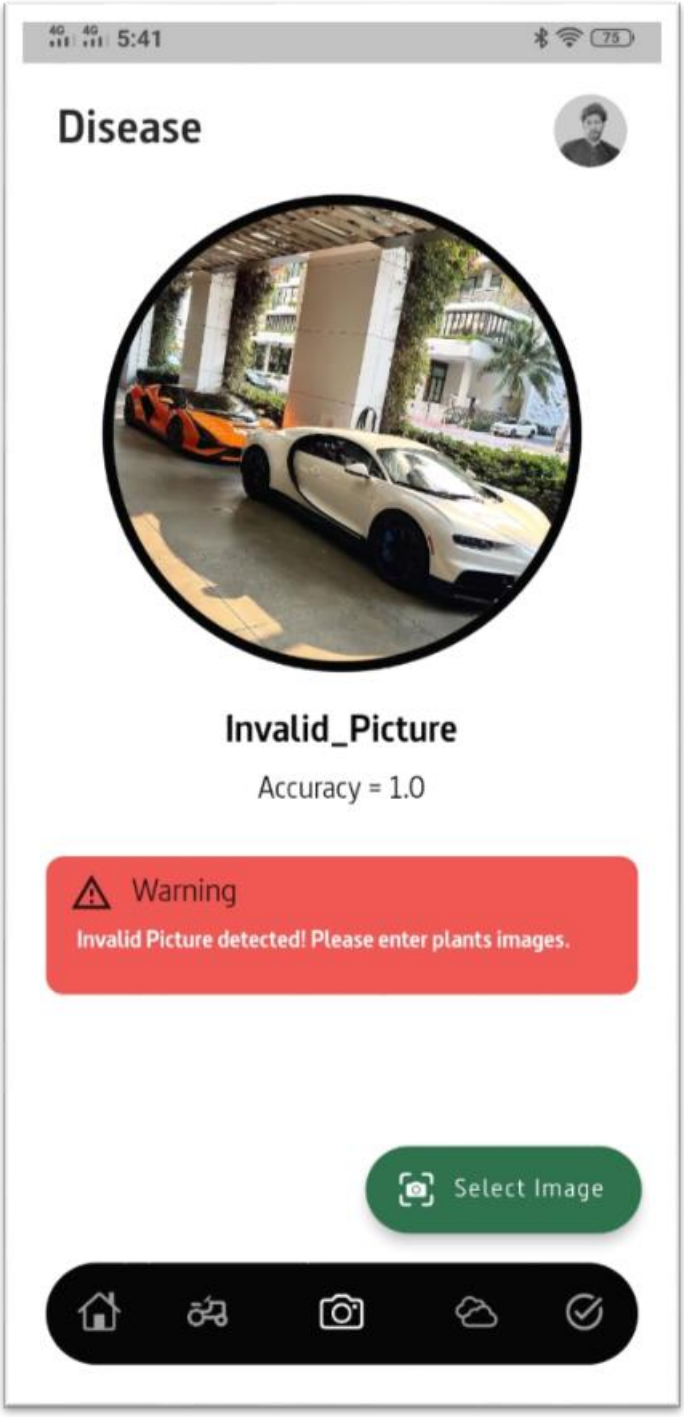

Figure 34: Invalid Image Diagnose Screen

5.4.23 Recommendation of Recovery procedure screen

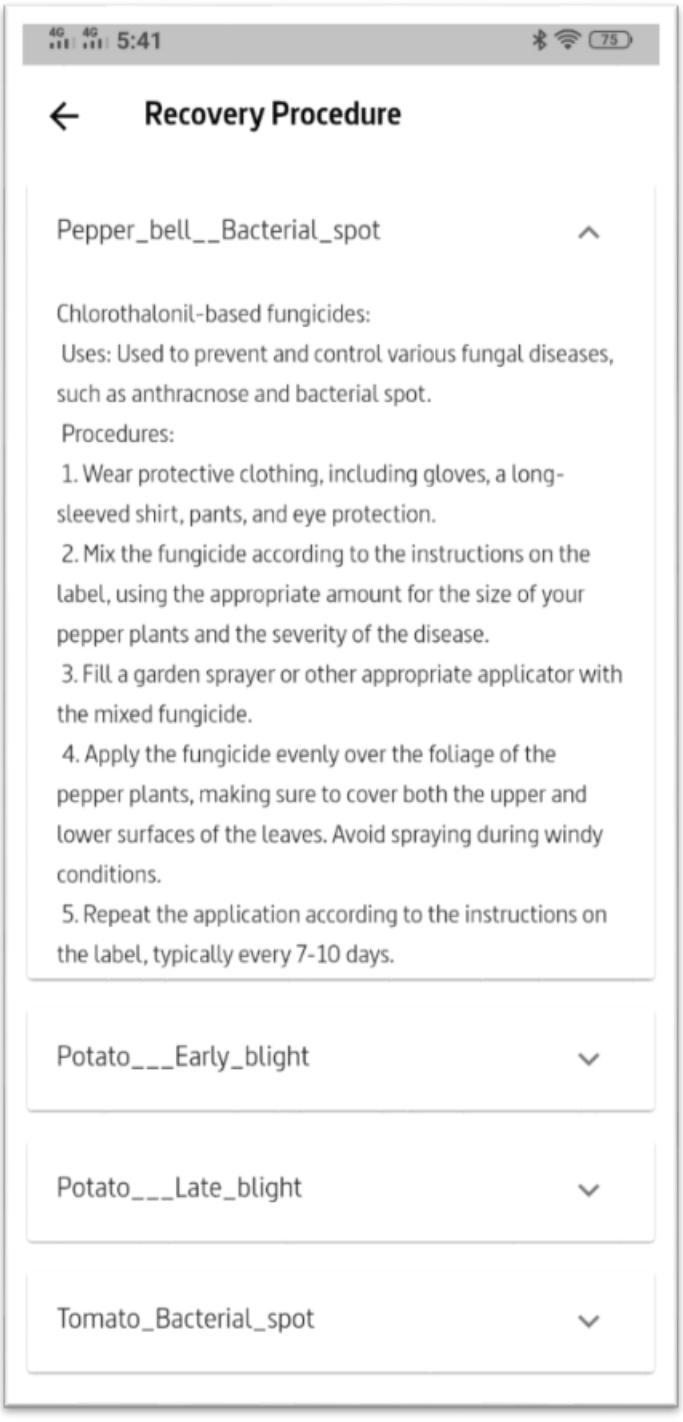

Figure 35: Recovery Procedure Recommendation Screen

Chapter 6

TESTING AND EVALUATION

6 Testing and Evaluation

The most important phase in the development process of any software is testing. In this phase, both the quality and the working functionality of the software are tested. This testing is done to confirm that the application is working properly. It is very important in the development phase. It is also used to check the errors in the application if any issues then we remove the error from the application. Also, when we talk about testing, we talk about Evaluation, at this stage we analyze the performance of the application by performing different tests to check the effectiveness.

6.1 System Testing

It is an experimental examination that testing directed to give identify the backer's data about the nature of the item below the test. The procedure of testing is used to identify errors in the software that is built. Testing plays a basic role in assuring the quality and reliability of programming. The consequences of testing are used later amidst support too.

6.2 Testing Objective

The enthusiasm behind testing is to show that a project is working properly without any errors. The main aim of testing is to find errors that might be available in the system. Hence, we do not initiate testing with the plane of demonstrating that a system works, yet the goal is to demonstrate that the project does not work. The procedure of testing is used for executing a project to find errors.

6.3 Manual Testing

In manual testing, there is no use of any automated tool, and test cases are executed manually. It helps to find out visible and hidden defects in a system. Firstly, the documentation is observed by the tester to know about testing areas of the system. Each line of code is examined and then the functionality of each module is checked.

6.4 Unit Testing

In this stage of testing, we divide the project into units or small parts, and each unit is separately identified and tested to ensure that each part of the system is working properly and according to the requirement of the system.

6.5 Testing Environment

Environment testing is an essential part of the testing stage because with time devices, systems, and software may change or upgrade, so we consider both old and upcoming new environments for our project. We test our project on Flutter and make it compatible with the maximum version of Flutter.

6.6 Functional Testing

Some of the functional testing is performed and their test cases are discussed below:

6.6.1 Sign up Test case

Table 21: Sign up Test case

No.	Test Case	Expected Result	Actual Result	Test Results
1	Open Crop Care App	Application Opened	Application Opened	Pass
2	Click on the signup/create account option	Signup Form/ screen appears	Signup Form/ screen appears	Pass
3	Enter Name	The name written by the user appears in text box	The name written by the user appears in text box	Pass
4	Enter email	Email written by the user appears in text box	Email written by the user appears in text box	Pass
5	Phone Number	The phone number written by the user appears in text box	The phone number written by the user appears in text box	Pass
5	Enter Password	Password written by the user Appears in text box	Password written by the user Appears in text box	Pass
6	Click on the ‘Signup Now button	A pop message “Account created successfully”	A pop message “Account created successfully”	Pass

		appears and user will be directed to the home page	appears and user is directed to the home page	

6.6.2 Login Test case

Table 22: Login Test case

No.	Test Case	Expected Result	Actual Result	Test Results
1	Open Crop Care App	Application Opened	Application Opened	Pass
2	Click on the Login button	Login Form appears	Login Form appears	Pass
3	Enter Email	Email written by the user appears will appear in text box	Email written by the user is appeared in text box	Pass
4	Enter Password	Password written by the user will appears in text box	Password written by the user is appeared in text box	Pass
5	Click on the 'Login ' button	User will be directed to home page	User is directed to the home page	Pass

6.6.3 Disease Detection Test case

Table 23: Disease Detection Test case

No.	Test Case	Expected Result	Actual Result	Test Results

1	Open Crop Care App	Application Opened	Application Opened	Pass
2	Login	Login successful	Login successful	Pass
3	Click on the ‘Camera’ icon	The Detection screen will open	The Detection screen opened successfully	Pass
4	Clink on the “select” button	User will directed to gallery	User is directed to the gallery	Pass
5	Select the plant image that you want to predict	The predicted Label will be displayed	The predicted label for the given image is displayed	Pass

6.6.4 Activity Planner Test case

Table 24: Activity Planner Test Case

No.	Test Case	Expected Result	Actual Result	Test Results
1	Open Crop Care App	Application Opened	Application Opened	Pass
2	Login	Login successful	Login successful	Pass
3	Click on the ‘Planner’ icon	The Planner screen will open	The Planner screen opened successfully	Pass
4	Clink on the “Complete” tab	List of completed tasks will be displayed	List of completed tasks is be displayed	Pass

s5	Clink on the "Incomplete" tab	List of uncompleted tasks will be displayed	List of uncompleted tasks is be displayed	Pass
6	Clink on the "Favorite" tab	List of favorite tasks will be displayed	List of favorite tasks is be displayed	pass
7	Click on the " + " button	Add task screen will appeared	Add task screen is appeared	Pass
8	Fill out all the info and click " Create a Task" button	The task will be added to the list	The task is added to the list	Pass
9	Swipe away the that you want to delete	The task will be deleted from the list	The task is deleted from the list	Pass

6.6.5 Farming Information Test case

Table 25: Farming Test case

No.	Test Case	Expected Result	Actual Result	Test Results
1	Open Crop Care App	Application Opened	Application Opened	Pass
2	Login	Login successful	Login successful	Pass
3	Click on the 'Farming' icon	The Farming screen will open showing a slider that contain cultivation, diseases and fertilizer tabs	The Farming screen is opened showing a slider that contain cultivation, diseases and fertilizer tabs	Pass

4	Clink on the "Cultivation" tab	A screen will appears showing the lists of plants	A screen is appeared showing the lists of plants	Pass
5	Click on the plant to view cultivation info	A screen will appears showing cultivation weeks list	A screen is appears showing cultivation weeks list	Pass
6	Click on "week" tab	The information about that week will be displayed	The information about that week is displayed	Pass

6.6.6 Profile Test case

Table 26: Profile Test case

No.	Test Case	Expected Result	Actual Result	Test Results
1	Open Crop Care App	Application Opened	Application Opened	Pass
2	Click on the "profile" icon	Profile screen will appears	Profile screen is appeared	Pass
3	Click on "Record" button	Record screen will appears, showing user record	Record screen appears, showing user record	Pass
4	Click on "website" option	User will be directed to our official website	User is directed to our official website	Pass
5	Click on "About us" option	User will be directed to about us page	User is directed to the about us page	Pass

6	Click on "switch mode" option	The app mode will be changed from black to white or white to black	The app mode is changed from black to white or white to black	Pass

Chapter 7

Conclusion and Future Work

7.1 Conclusion

At last, we have developed an application in full working condition performing some of our main functionalities. In this mobile application, a CNN is used for disease detection of vegetable plant. For the implementation first, we have taken image dataset from Kaggle. Then the model of CNN is trained for the test of the 15different classes of 3 vegetable plants and I invalid class for dealing with images other than plants. After the model is trained on the given images, the model is able to classify the vegetable plant into 16 different classes. For this current work, 95% accuracy is achieved with CNN architecture. This high percentage is due to the use of CNN as our classifier and epochs. After the prediction of disease, app will provide a treatment procedure for those diseases. Planner module help the farmer to plan their everyday activities. We have achieve our goal of farming education by providing a farmer module where cultivation, diseases, and fertilizers information is provided. Updating farmer about weather condition is achieved with weather forecast feature.

7.2 Future Work

In near future, we will create a desktop version of Crop Care App. This would help the large farming centers to check the diseases of crop and vegetable plants avoiding the traditional method of examination of the crop and vegetable plants that is done by the humans through which many of the diseases are predicted incorrectly. The monitoring of plant diseases is done using CCTV cameras or drones. We will also going to work on creating a dataset that contain high quality and diverse images of almost all the diseases of vegetables and crops. So that our app can detect the diseases of almost all type of vegetables crop plants. We will create a web application that provide all the functionality of our app in web platform.

Chapter 8

References

References

[1] https://docs.flutter.dev/get-started/install

[2] https://apps.apple.com/us/app/leaf-doctor/id874509900

[3] https://www.kaggle.com/emmarex/plantdisease

[4] https://www.researchgate.net/publication/338628689_PlantDoc_A_Dataset_for_Visual_Plant_Disease_Detection

[5] https://www.cropdiagnosis.com/portal/crops/en/home

[6] https://www.learndatasci.com/tutorials/convolutional-neural-networks-image-classification/#Downloadingandpreparingthedataset

[7] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7599959/#:~:text=The%20Xception%20model%20attained%20the,disease%20on%20the%20PlantVillage%20dataset