**COMSATS University Islamabad,**

**Attock Campus, Attock Pakistan**

**FARMER’S TOUCH**

***By***

**USAMA BIN ARSHAD CIIT/SP20-BSE-020/ATK**

**USAMA NAWAZ CIIT/SP20-BSE-037/ATK**

***Supervisor***

**MR. MUHAMMAD WASEEM KHAN**

***Bachelor of Science in Software Engineering (2020-2024)***

**The candidate confirms that the work submitted is their own and appropriate Credit has been given where reference has been made to the work of others**.

**COMSATS University Islamabad,**

**Attock Campus, Attock Pakistan**

**FARMER’S TOUCH**

**A project presented to**

**COMSATS University, Islamabad**

**In partial fulfillment**

**Of the requirement for the degree of**

***Bachelors of Science in Software Engineering (2020-2024)***

**By**

**USAMA BIN ARSHAD CIIT/SP20-BSE-020/ATK USAMA NAWAZ CIIT/SP20-BSE-037/ATK**

**DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied the report entirely based on our personal efforts. If any part of this project is proved to be copied from any source or found to be a reproduction of some other, 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.

USAMA BIN ARSHAD USAMA NAWAZ

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (SE) “FARMER’S TOUCH” was developed by

**USAMA BIN ARSHAD (CIIT/SP20-BSE-020/ATK)** and **USAMA NAWAZ (CIIT/SP20-**

**BSE-037/ATK))** under the supervision of “MR. MUHAMMAD WASEEM KHAN” and that in his opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Software Engineering.

**Supervisor**

**External Examiner**

**Head of Department**

**(Department of Computer Science)**

**Executive Summary**

Agriculture is an important sector in which our modes of life and business innovation combine. It has manifold roles in the economy of any nation and these roles include food security, poverty reduction, the industrial revolution, and economic growth, especially in developing countries.

In this application (**FARMER’S TOUCH**) we are mainly focused on the detection of the disease that is affecting our crops **(corn, potato, rice, and wheat)** and the medicine that the farmer should give to crops. This app will help farmers who don't know the disease name that is affecting the crops and the medicine that they should give to crops to control the disease.

This app also has other features like weather forecasting, medicine names, and Buying and selling the product this app will sum up all the features that farmers want in our app, If they want to check the weather update this app will help them If they don't know the medicine name of disease crops that are mention above this app will help them, If they want to sell or buy the product this app will help them

**Acknowledgement**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task. We are greatly indebted to our project supervisor “MR. MUHAMMAD WASEEM KHAN”. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are grateful to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us with the values of honesty & hard work.

USAMA BIN ARSHAD USAMA NAWAZ

**Abbreviations**

|  |  |
| --- | --- |
| **SRS** | Software Require Specification |
| **PC** | Personal Computer |
| **cv** | Computer Vision |
| **HCI** | Human Computer Interaction |
| **CNN** | Convolutional Neutral Network |
| **SDD** | Software Design Description |

**Table of Contents**

1. [Introduction 1](#_bookmark0)
   1. [Vision Statement 2](#_bookmark1)
   2. [Related System Analysis/Literature Review 2](#_bookmark2)
   3. [Project Deliverables 3](#_bookmark4)
   4. [System Limitations 3](#_bookmark5)
   5. [Tools and Technologies 4](#_bookmark6)
   6. [Relevance to Course Modules 4](#_bookmark8)
2. [Problem Definition 6](#_bookmark9)
   1. [Problem Statement 6](#_bookmark10)
   2. [Objectives of the Proposed System 6](#_bookmark11)
   3. [Scope 7](#_bookmark12)
   4. [Modules 7](#_bookmark13)

[Module 1: Disease Detection 7](#_bookmark14)

[Module 2: Medicine 8](#_bookmark16)

[Module 3: Weather 8](#_bookmark17)

[Module 4: Buying/Selling 8](#_bookmark18)

1. [Requirement Analysis 10](#_bookmark19)
   1. [User classes and characteristics 10](#_bookmark20)
   2. [Requirement Identifying Technique 10](#_bookmark21)
      1. [Farmer use case Diagram 10](#_bookmark22)
      2. [Admin Use case Diagram 11](#_bookmark24)
      3. [Detail Use case Diagram 12](#_bookmark26)
   3. [Functional Requirements 21](#_bookmark36)
      1. [Disease Detection 21](#_bookmark37)
      2. [Medicine 21](#_bookmark39)
      3. [Weather forecasting 22](#_bookmark41)
      4. [Buying and Selling 22](#_bookmark43)
   4. [Non-Functional Requirements 23](#_bookmark45)
      1. [Maintainability 23](#_bookmark46)
      2. [Usability 23](#_bookmark47)
      3. [Performance 23](#_bookmark48)
      4. [Security 23](#_bookmark49)
   5. [External Interface Requirements 24](#_bookmark50)
      1. [User Interfaces Requirements 24](#_bookmark51)
      2. [Software Interface 24](#_bookmark52)
2. [Design and Architecture 26](#_bookmark53)
   1. [Architectural Design 26](#_bookmark54)
      1. [Box and Line Diagram 26](#_bookmark55)
      2. [Multi-tiered 27](#_bookmark57)
   2. [Design Models for Object Oriented Development Approach 28](#_bookmark59)
      1. [Activity Diagram 28](#_bookmark60)
      2. [Class Diagram 29](#_bookmark62)
      3. [Sequence Diagram 30](#_bookmark64)
      4. [State Transition Diagram 31](#_bookmark66)
   3. [Data Design 32](#_bookmark68)
      1. [Data Dictionary 32](#_bookmark69)
   4. [Human Interface Design 33](#_bookmark70)
      1. [Screen Images 33](#_bookmark71)
      2. [Screen Objects and Actions 35](#_bookmark75)
3. [Implementation 37](#_bookmark76)
   1. [Algorithm 37](#_bookmark77)
   2. [External APIs/SDKs 38](#_bookmark79)
   3. [User Interface 39](#_bookmark81)
      1. [Login Screen 39](#_bookmark82)
      2. [Signup Screen 40](#_bookmark84)
      3. [Home Screen 41](#_bookmark86)
      4. [Medicine 42](#_bookmark88)
      5. [Corn medicine 43](#_bookmark90)
      6. [Rice medicine 44](#_bookmark92)
      7. [Potatoes Medicine 45](#_bookmark94)
      8. [Wheat Medicines 46](#_bookmark96)
      9. [Weather 47](#_bookmark98)
      10. [Buying/Selling 48](#_bookmark100)
   4. [Deployment 51](#_bookmark104)
4. [Testing and Evaluation 53](#_bookmark105)
   1. [Unit Testing 53](#_bookmark106)
   2. [Functional Testing 55](#_bookmark110)
   3. [Business Rules Testing 58](#_bookmark117)
   4. [Integration Testing 59](#_bookmark121)
5. [Conclusion and Future Work 62](#_bookmark124)
   1. [Conclusion 62](#_bookmark125)
   2. [Future Work 62](#_bookmark126)

[8. References 64](#_bookmark127)

**List of Figures**

[Figure 1: Farmer Use case Diagram 10](#_bookmark23)

[Figure 2: Admin Use case Diagram 11](#_bookmark25)

[Figure 3: Box and Line Diagram 26](#_bookmark56)

[Figure 4: Multi-tiered Diagram 27](#_bookmark58)

[Figure 5: Activity Diagram 28](#_bookmark61)

[Figure 6: Class Diagram 29](#_bookmark63)

[Figure 7: Sequence Diagram 30](#_bookmark65)

[Figure 8: State Transition Diagram 31](#_bookmark67)

[Figure 9: Login Screen 33](#_bookmark72)

[Figure 10: Sign up Screen 34](#_bookmark73)

[Figure 11: Home screen 35](#_bookmark74)

[Figure 12: Login Screen 39](#_bookmark83)

[Figure 13: Signup Screen 40](#_bookmark85)

[Figure 14: Home Screen 41](#_bookmark87)

[Figure 15: Medicine Screen 42](#_bookmark89)

[Figure 16: Corn Medicine Screen 43](#_bookmark91)

[Figure 17: Rice Medicine Screen 44](#_bookmark93)

[Figure 18: Potatoes Medicine Screen 45](#_bookmark95)

[Figure 19: Wheat Medicine Screen 46](#_bookmark97)

[Figure 20: Weather Screen 47](#_bookmark99)

[Figure 21: Crop Disease Detection 48](#_bookmark101)

[Figure 22: Buying screen 49](#_bookmark102)

[Figure 23: Selling Screen 50](#_bookmark103)

**List of Table**

[Table 1: Related System Analysis 3](#_bookmark3)

[Table 2: Tool and Technologies 4](#_bookmark7)

[Table 3: Disease Name 7](#_bookmark15)

[Table 4: Detail use case for login 12](#_bookmark27)

[Table 5: Detail use case for sign up 13](#_bookmark28)

[Table 6: Detail use case for disease detection 14](#_bookmark29)

[Table 7: Detail use case for Input image 15](#_bookmark30)

[Table 8: Detail use case for Weather forecast 16](#_bookmark31)

[Table 9: Detail use case for Medicine 17](#_bookmark32)

[Table 10: Detail use case for Buying and Selling 18](#_bookmark33)

[Table 11: Detail use case for Login 19](#_bookmark34)

[Table 12: Detail use case for Sign up 20](#_bookmark35)

[Table 13: Description for Disease Detection 21](#_bookmark38)

[Table 14: Description for Medicine 21](#_bookmark40)

[Table 15: Description for Weather Forecasting 22](#_bookmark42)

[Table 16: Description for Buying and Selling 22](#_bookmark44)

[Table 17: Algorithm 37](#_bookmark78)

[Table 18: Details of APIs used in the project 38](#_bookmark80)

[Table 19: Unit Testing-1 53](#_bookmark107)

[Table 20: Unit Testing-2 54](#_bookmark108)

[Table 21: Unit Testing-3 54](#_bookmark109)

[Table 22: Functional Testing-1 55](#_bookmark111)

[Table 23: Functional Testing-2 55](#_bookmark112)

[Table 24: Functional Testing-3 56](#_bookmark113)

[Table 25: Functional Testing-4 56](#_bookmark114)

[Table 26: Functional Testing-4 57](#_bookmark115)

[Table 27: Functional Testing-4 57](#_bookmark116)

[Table 28: Business Testing-1 58](#_bookmark118)

[Table 29: Business Testing-2 58](#_bookmark119)

[Table 30: Business Testing-3 59](#_bookmark120)

[Table 31: Integration Testing-1 59](#_bookmark122)

[Table 32: Integration Testing-2 60](#_bookmark123)

**Chapter: 01**

# Introduction

Agriculture is an important sector in which our modes of life and business innovation combine. It has manifold roles in the economy of any nation and these roles include food security, poverty reduction, the industrial revolution, and economic growth, especially in developing countries [1]. Agriculture is the art of cultivating crops and raising livestock, it plays a very huge role in the economies of underdeveloped or developing countries.

**"Advances in medicine and agriculture have saved vastly more lives than have been lost in all the wars in history." [Carl Sagan]**

In Pakistan agriculture used to be a leading sector in economic growth, during the time of independence agriculture made up around 53% of the country's GDP, and more than 65% of our labor was employed in this sector [2]. These figures changed dramatically because of the lack of interest in agriculture and those who have an interest don't know about the disease and medicine of the crops. Advancements in technology can be used to increase crop production by detecting the diseases that affect the crops.

“A Farmer's Touch" is a mobile application for the ease of farmers. In this application Farmer can see the medicine's name and can detect disease that occurs in the crops by just simply taking pictures of the diseased crop and uploading them to the application if they want to check the weather update this application will also provide that or want to sell/buy the product this application will help them.

The classifier we choose for our model is CNN, and why we choose it because it is a subtype of Neural Networks that is mainly used for applications in image and speech recognition. Its built-in convolutional layer reduces the high dimensionality of images without losing their information. During the training of CNN, the neural network is fed with a large dataset of images labeled with their corresponding class labels. The CNN network processes each image with its values being assigned randomly and then makes comparisons with the class label of the input image. The main advantage of CNN compared to its predecessors is that it automatically detects significant features without any human supervision which made it the most used

## Vision Statement

Agriculture plays an important role in the development of a country. Agriculture not only fulfills the food requirement but also provides revenue that plays an important role in the economy of the country. Pakistan is an agricultural country, but due to a lack of knowledge of farmer production crops decreased. By using advanced technology, we can increase production. We can increase production by providing information to the farmer who not only detect the disease of a crop but also see medicine for the disease of their crops, farmers can get weather updates daily by daily that can help them to manage water resources for their crops. Farmers also can sell /buy crops in the market at the best price and time.

The basic idea of this project is to provide all these features on one platform that can former to increase the production of crops and plays its part in the development of the country.

## Related System Analysis

Following are the related system:

* + - **GO GREEN:** It is a mobile application, and it has also other features but when compared with our application the common feature is weather updated.
    - **Growers Edge Farm Manager:** It is a mobile application, and it has also other features but when compared with our application the common feature is weather update, and buying/selling [4].
    - **KISSAN ZAR ZAMEEN:** It has also other features but when compared with our application the common feature is weather update [5].
    - **Green Farming:** It has also other features but when compared with our application the common feature is Medicine and weather update.
    - **Gramophone:** It has also other features but when compared with our application the common feature is Weather update and Buying/Selling [6].

**Table 1: Related System Analysis**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Disease**  **Detection** | **Medicines** | **Weather** | **Buying** | **Selling** |
| Farmer’s Touch |  |  |  |  |  |
| GO GREEN |  |  |  |  |  |
| Growers Edge Farm  Manager |  |  |  |  |  |
| KISSAN ZAR  ZAMEEN |  |  |  |  |  |
| Green Farming |  |  |  |  |  |
| Gramophone |  |  |  |  |  |

## Project Deliverables

Following are the project's deliverables.

* + - User interface design.
    - Database design and development.
    - Image processing.
    - Integration by third-party service.
    - User documentation.
    - Testing and quality insurance.
    - Maintenance.

## System Limitations/Constraints

*LI-1: It's a mobile application users should have mobile and internet connectivity.*

*LI-2: It only gives the medicine name of crops (corn, potato, rice, and wheat).*

## Tools and Technologies

The following are tools that we used in our project.

**Table 2: Tool and Technologies**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tools And**  **Technologies** | **Tools** | **Version** | **Rationale** |
| Visual Studio Code | 2015 | IDE |
| Firebase | 2015 | DBMS |
| Adobe Photoshop | CSC 6 | Design Work |
| **Technology** | **Version** | **Rationale** |
| React Native | 17 | Front-end Development |
| JavaScript | 3 | Back-end Development |
| Python | 8 | Back-end Development |

## Relevance to Course Modules

To complete our project we require course knowledge that is mentioned below.

* + - **Computer Vision**: We studied deep learning which will help us while training our data.
    - **Mobile Application Development**: we studied how to develop Mobile Applications by using React-Native. This will help us while making a Mobile application project.
    - **Report Writing Skills**: This course is about how to make reports, so for our project, we need to write documentation, and this course is helping us with this task.
    - **Database Management System**: This course is about how to store data, so for our project, there are two modules related to databases this course will help us in this task.
    - **Human Computer Interaction**: This is a course about how to make good design, so this course will also help us in making a good design for our project.

‘

**Chapter # 02**

# Problem Definition

## Problem Statement

The agriculture industry faces many challenges, which include the diseases of the crops that cannot only destroy the crops but also cause significant economic loss. So the early detection of crop disease can prevent the damage caused by the disease, However, Identifying crop disease is a very challenging task. To address this problem our application is designed which automatically detect the disease of the crops from images using computer vision techniques and a deep learning algorithm.

After detecting the disease of the crops the next problem they faced was that they didn’t know the medicine name of the crops disease that was detected, so as we detecting the disease of only the four crops (corn, rice, wheat, potatoes) so our application will display the medicine name of these crops in medicine module where Farmers can see the required medicine name.

Similarly, the next problem the farmer faces is the difficulties in selling their crops at a very fair price and lack of direct access to buyers because they have to rely on a middleman, who may not offer them the best price for their crops. To address this information our application is designed where all information about crops like their name, quantity, price, etc. will be stored in the database, and the buyer who wants to buy the crops can easily buy them from our application.

Another problem the farmers faces is that they are highly dependent on the weather condition for their agricultural activities. So our application will also provide weather information to the farmers so that they can improve their efficiency and profitably while minimizing their environmental impact.

## Problem Solution

The problem statement outline the challenge faced by the agricultural industry, To tackle these:

* Our application use deep learning model to detect disease in crops
* Our application use their own database to allow the user to sell/buy their product
* Our application use weather api to daily update the user about weather
* Our application use their own database to allow the user to see the medicine name

## 2.3 Objectives of the Proposed System

The following is the objective of our application

* + - Provides a platform to predict diseases of crops.
    - Selling and buying farming products
    - Provide medicine of that crop
    - Provide a weather update

## Scope

This application will provide you with the best platform for the detection of the crop's disease. It will give you the required medicine for the crops that are affected by the disease. It will provide you with the Seller list where the buyer can easily buy their crops from the database of this application. It will also provide the weather update.

One of the advantages of having these modules is that the early detection of diseases is that it allows the farmer to take the necessary step to prevent the spread of the disease and minimize the loss of the crop.

The Second module is the medicine module which helps the farmers to select the right medicine for their crops based on the disease. The next module is buying/selling which helps the farmer to make a better decision about when to buy or sell their crops. The Weather update module helps the farmer to have weather information, and plan their farming activities which can be planting, etc.

## Modules

The Module that we are going to use in our System are as follows:

### Module 1: Disease Detection

* + - From this option, you can upload a pic of your disease crops and it will tell you the disease name

**The following disease is going to predict**

Table 3: Disease Name

|  |  |
| --- | --- |
| **Crops Name** | **Disease Name** |
| Corn | Gray Leaf spot |
| potatoes | Late Blight |
| Rice | Brown Spot |
| Wheat | Brown Rust |

### Module 2: Medicine

* + - After successfully getting the disease name the farmer can see the medicine name of the disease

### Module 3: Weather

* + - They can get a weather update

### Module 4: Buying/Selling

* + - They can buy/sell a product they want

**CHAPTER # 03**

# Requirement Analysis

Requirement analysis is the process of identifying, gathering, and documenting the need and constraints of the stakeholder for the system or product. It is the foundation of software development, and it helps ensure that the end product meet requirements and expectation of its user.

## User classes and characteristics

1. **Admin:** Admin can add medicine names or add more crops in the future because right now we have 4 crops that we are working.
2. **Farmer:** The farmer can check the medicine name, check the disease name of crops, check the weather forecast, and buy or sell online.

## Requirement Identifying Technique

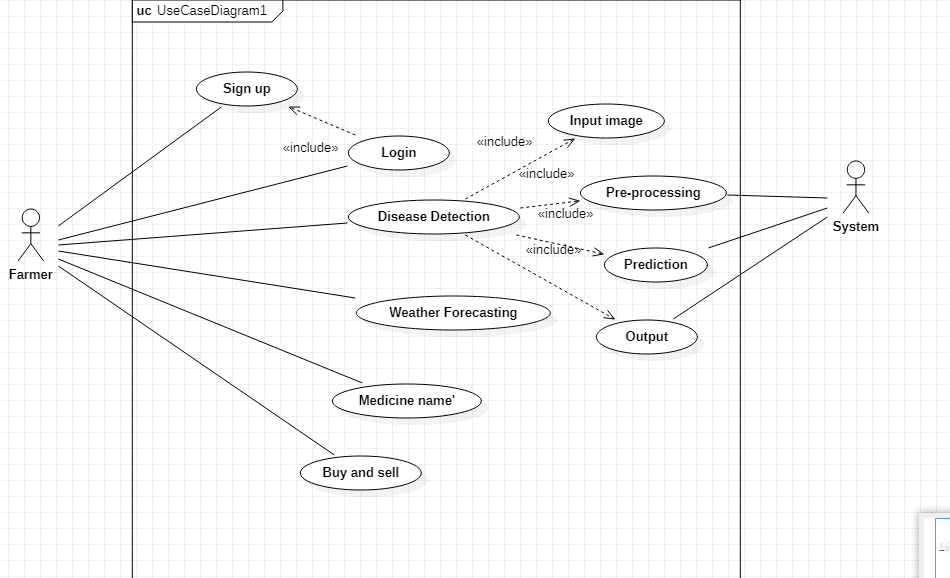
* + 1. **Farmer use case Diagram**

Figure 1: Farmer Use case Diagram

**Description of Farmer Use Case Diagram:**

In the above Diagram, the Farmer has to log in to use our application, if there is a new farmer then firstly he has to create his account which will be in the Signup option. After successfully creating an account and login into the account Home screen will pop up and there will 4 options in that:

* Disease Detection
* Medicine
* Weather
* Buying/selling

In disease detection, farmer have to input the image and system will preprocess the image, make a prediction on it display the output to farmer.

In Medicine, Farmer has to select which crop medicine he want, then after selecting it the medicine of the crop disease will appear on it.

In weather, Farmer can check the weather

In buying/Selling, When farmer select the Buying/Selling option then two option will appear Buy and sell option. In the sell option Farmer have to give the product related information if he want to sell his product, and then this information will be shown in buy option

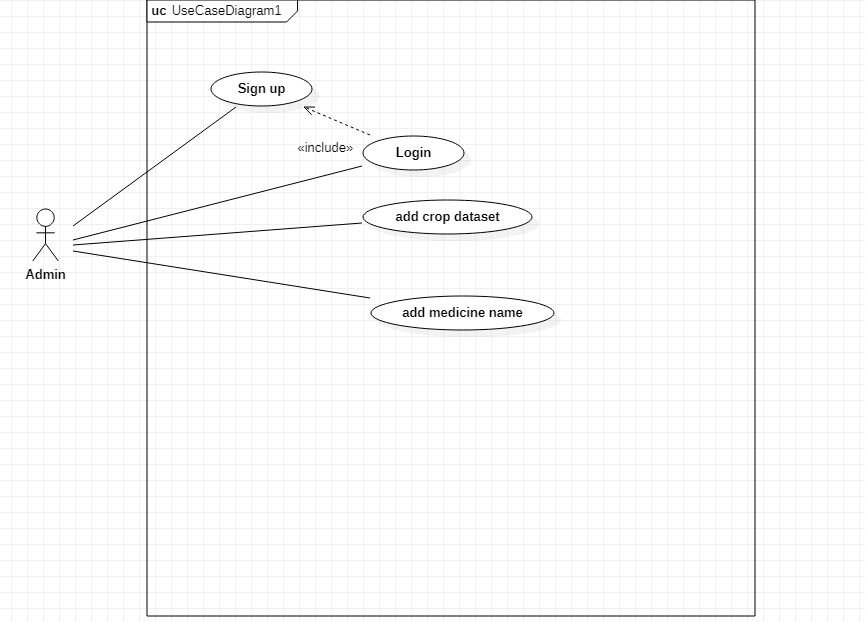
* + 1. **Admin Use case Diagram**

Figure 2: Admin Use case Diagram

**Description of Admin Use case Diagram**

The above diagram showing us that if admin want to add more crop dataset in future he can easily add it, and also if he want to more medicine name in it he can easily add it

* + 1. **Detail Use case Diagram**
       1. **Login**

Table 4: Detail use case for login

|  |  |
| --- | --- |
| Use case ID | 1 |
| Use case name | Login |
| Actor | Farmer |
| Description | The Farmer has to login into his account to use the application |
| Trigger | The farmer indicates that he wants to use our application |
| Pre-condition | Give Email and password to access the application |
| Post-condition | Farmers can use the application |
| Normal Flow | 1. Farmers want to login into our application 2. Farmers open the application 3. Login screen appears 4. Entering the correct Email and password 5. Farmers now can use our application |
| Alternative flow | None |
| Exception | None |
| Business rule | None |
| Assumption | Assume that 40 percent of farmers use our application at the same  time |

* + - 1. **Sign up**

Table 5: Detail use case for sign up

|  |  |
| --- | --- |
| Use case ID | 2 |
| Use case name | Sign up |
| Actor | Farmer |
| Description | The farmer has to register his account to use the application |
| Trigger | New users want to use our application |
| Pre-condition | Farmers have no account |
| Post-condition | Farmer have account |
| Normal flow | 1. New users want to use our system 2. The Login screen page appears 3. Sign up text is written on login screen 4. By tapping on that text Sign up page appear 5. Entering the required information 6. Farmer created his account 7. Now he can use our application |
| Alternative flow | None |
| Exception’ | None |
| Business rule | None |
| Assumption | Assume that many people create their accounts at the same time |

* + - 1. **Disease Detection**

Table 6: Detail use case for disease detection

|  |  |
| --- | --- |
| Use case ID | 3 |
| Use case name | Disease detection |
| Actor | Farmer |
| Description | Farmer has to detect the disease of the crops |
| Trigger | A Farmer indicates that they want to check the name of the disease |
| Pre-condition | Farmers do not know the disease's name |
| Post-condition | Farmers know the disease's name |
| Normal Flow | 1. After Logging into our account Home screen appear 2. Many modules will be available on the home screen 3. Farmers click on the disease detection module and they can check the disease name |
| Alternative flow | None |
| Exception | None |
| Business rule | None |
| Assumption | Assume that many people use this module |

* + - 1. **Input image**

Table 7: Detail use case for Input image

|  |  |
| --- | --- |
| Use case Id | 4 |
| Use case name | Input image |
| Actor | Farmer |
| Description | Farmers have to upload an image of the disease crop |
| Trigger | Farmers indicate that they want to check the name of the disease so  by clicking on the disease detection module they upload their image |
| Pre-condition | Farmers did not upload the image of the disease crop |
| Post-condition | Farmers upload the image of the disease crop |
| Normal Flow | 1. After Logging into our account Home screen appear 2. Many modules will be available on the home screen 3. Farmers click on the disease detection module 4. In that module, user upload their image |
| Alternative flow | None |
| Exception | None |
| Business rule | None |
| Assumption | Assume that many people use this module |

* + - 1. **Weather forecast**

Table 8: Detail use case for Weather forecast

|  |  |
| --- | --- |
| Use Case ID | 5 |
| Use case name | Weather forecast |
| Actor | Farmer |
| Description | Farmers can see the weather update |
| Trigger | Farmers indicate that they want to check the weather |
| Pre-condition | Farmers do not know the weather |
| Post-condition | Farmers know the weather |
| Normal flow | * After Logging into our account Home screen appear * Many modules will be available on the home screen * Farmers click on the Weather module * In that module, user can check their weather |
| Alternative flow | None |
| Exception | None |
| Business rule | None |
| Assumption | Assume that many people use this module |

* + - 1. **Medicine**

Table 9: Detail use case for Medicine

|  |  |
| --- | --- |
| Use case ID | 6 |
| Use case name | Medicine |
| Actor | Farmer |
| Description | Farmers can see the medicine name of the disease crop |
| Trigger | A Farmer indicates that they want to check the medicine for the  required disease |
| Pre-condition | Farmers do not know the medicine name of the disease crop |
| Post-condition | Farmers know the medicine name of the disease crop |
| Normal flow | 1. After Logging into our account Home screen appear 2. Many modules will be available on the home screen 3. Farmer click on the Medicine module 4. In that module, the user can check the medicine name of the required disease |
| Alternative flow | None |
| Exception | None |
| Business rule | None |
| Assumption | Assume that many people use this module |

* + - 1. **Buying and selling**

Table 10: Detail use case for Buying and Selling

|  |  |
| --- | --- |
| Use case ID | 7 |
| Use case name | Buying and selling |
| Actor | Farmer |
| Summary | Farmers can buy or sell their product |
| Trigger | A Farmer indicates that they want to Buy or sell their product |
| Pre-condition | Farmers cannot buy or sell their product |
| Post-condition | Farmers can buy or sell their product |
| Normal flow | * After Logging into our account Home screen appear * Many modules will be available on the home screen * Farmer click on Buying and Selling module * In that module, user can Buy or Sell their product |
| Alternative flow | None |
| Exception | None |
| Business rule | None |
| Assumption | Assume that many people use this module |

* + - 1. **Login**

Table 11: Detail use case for Login

|  |  |
| --- | --- |
| Use case ID | 8 |
| Use case name | Login |
| Actor | Admin |
| Description | Admin has to login into his account to use the application |
| Trigger | Admin indicates that he wants to use the application |
| Pre-condition | Give Email and password to access the application |
| Post-condition | Admin can use the application |
| Normal Flow | * Admin wants to login into our application * Admin opens the application * Login screen appears * Entering the correct Email and password * Admin now can use our application |
| Alternative flow | None |
| Exception | None |
| Business rule | None |
| Assumption | No Assumption Only 2 to 3 admins can use this application |

* + - 1. **Sign up**

Table 12: Detail use case for Sign up

|  |  |
| --- | --- |
| Use case ID | 9 |
| Use case name | Sign up |
| Actor | Admin |
| Description | Admin has to register his account to use the application |
| Trigger | Admin wants to use our application when the application created |
| Pre-condition | Admin has no account |
| Post-condition | Admin have account |
| Normal flow | * Admin wants to use our system * The Login screen page appears * Sign up text is written on the login screen * By tapping on that text Sign up page appear * Entering the required information * Admin created his account * Now he can use the application |
| Alternative flow | None |
| Exception’ | None |
| Business rule | None |
| Assumption | No Assumption Only 2 to 3 admin can use this application |

## Functional Requirements

Functional requirements define what an application is capable of doing. These requirements describe the function, features, and capabilities of the system.

* + 1. **Disease Detection**

Table 13: Description for Disease Detection

|  |  |
| --- | --- |
| **Identifier** | FR-1 |
| **Title** | Disease Detection |
| **Requirement** | In this requirement, the Farmers shall be able to detect the disease of crops |
| **Source** | Supervisor |
| **Rationale** | The Farmer can easily detect the disease in crops |
| **Dependencies** | None |
| **Priority** | High |

* + 1. **Medicine**

Table 14: Description for Medicine

|  |  |
| --- | --- |
| **Identifier** | FR-2 |
| **Title** | Medicine |
| **Requirement** | In this requirement, the Farmers shall be able to find out the medicine name of the disease crops |
| **Source** | Supervisor |
| **Rationale** | The Farmer can easily find out the medicine name of the disease crops |
| **Dependencies** | None |
| **Priority** | High |

* + 1. **Weather forecasting**

Table 15: Description for Weather Forecasting

|  |  |
| --- | --- |
| **Identifier** | FR-3 |
| **Title** | Weather Forecasting |
| **Requirement** | In this requirement, the Farmers shall be able to check the weather update |
| **Source** | Supervisor |
| **Rationale** | The Farmer can easily check the weather update |
| **Dependencies** | None |
| **Priority** | Medium |

* + 1. **Buying and Selling**

Table 16: Description for Buying and Selling

|  |  |
| --- | --- |
| **Title** | Buying and Selling |
| **Requirement** | With this requirement, the Farmers can easily buy or sell their product |
| **Source** | Supervisor |
| **Rationale** | The Farmer can easily find can easily buy or sell their product |
| **Dependencies** | None |
| **Priority** | Medium |

## Non-Functional Requirements

Non-Functional Requirements are specifications that describes the characteristic of a system that is not related to its specific function but rather it is the overall quality of the system.

* + 1. **Maintainability**

Our system will be easy to maintain and update our time, with minimal disruption to its user.

* + 1. **Usability**

Our System will be user-friendly so that the user will not face any difficulty while using our application.

* + 1. **Performance**

The performance of our system will be that it can handle a certain number of a user within a specified time frame*.*

* + 1. **Security**

Our system will be secure in that it protects user data from unauthorized access.

## External Interface Requirements

* + 1. **User Interfaces Requirements**

The Following are the user Interface Requirements of our application:

* **User-Friendly:** The User interface should be easy to use, with clear navigation, understandable instruction, and clear button.
* **Consistent design:** The app should have a consistent design across all pages and screens to provide a better user experience.
* **Responsive design:** The app should be designed to work well on different devices such as smartphones, tablet.
* **Performance:** The user interface must be responsive and provide fast feedback to the user.
* **Security:** The user interface must provide appropriate security measures to protect user data and prevent unauthorized access.
  + 1. **Software Interface**
* **API:** To check the weather using API
* **Database:** To store information using the Firebase database

**CHAPTER # 04**

# Design and Architecture.

## Architectural Design

* + 1. **Box and Line Diagram**

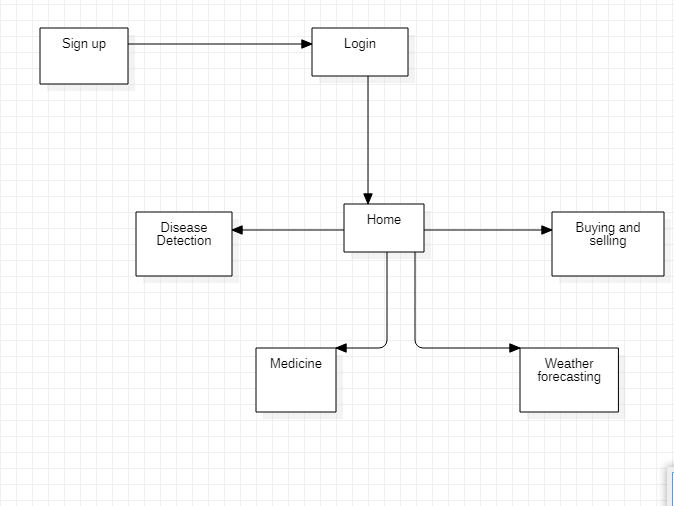


Figure 3: Box and Line Diagram

**Description of Box and Line Diagram**

The above diagram is that firstly farmer has to sign up for the account to use our application, after signing up the farmer has to log in to use our app. After login home page will appear in the home following option will be displayed:

* Disease Detection
* Medicine
* Buying and Selling
* Weather forecasting
  + 1. **Multi-tiered**

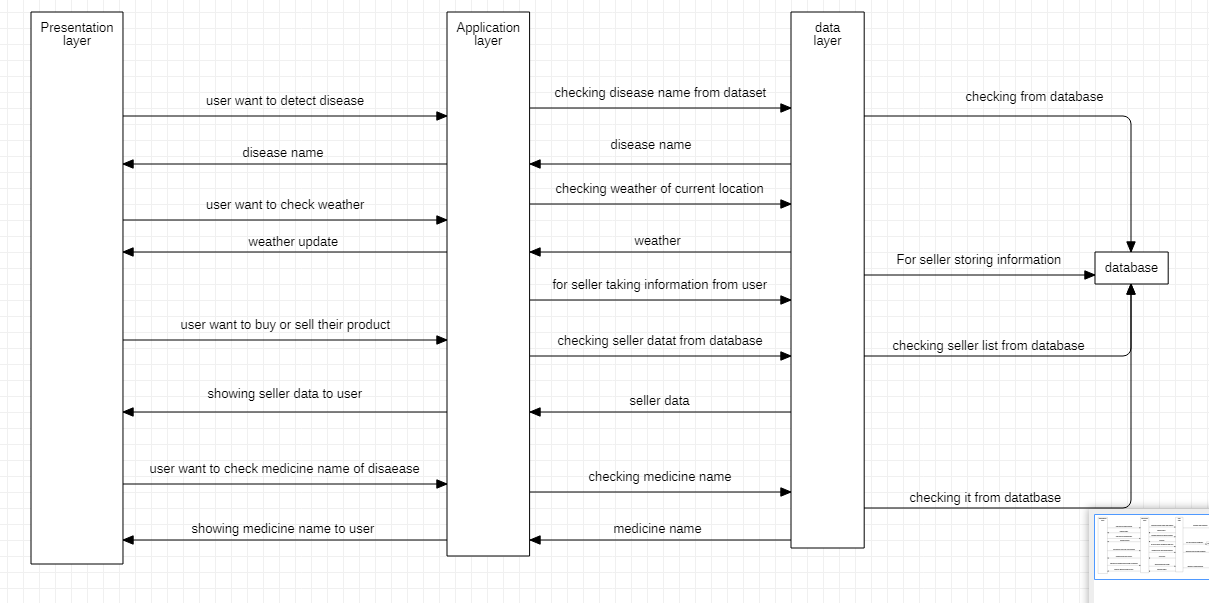


Figure 4: Multi-tiered Diagram

**Description of Multi-tiered**

The above diagram is showing that when a farmer wants to detect a disease the crops will interact with the presentation layer, and then the presentation will process the request to the application layer, the application layer is responsible for processing user requests after processing it will check in the database layer, after finding it will move the result to the presentation layer and presentation layer will display the result, similarly if a farmer wants to buy/sell, check the medicine name or check whether the same process will be carried out.

## Design Models for Object Oriented Development Approach

* + 1. **Activity Diagram**

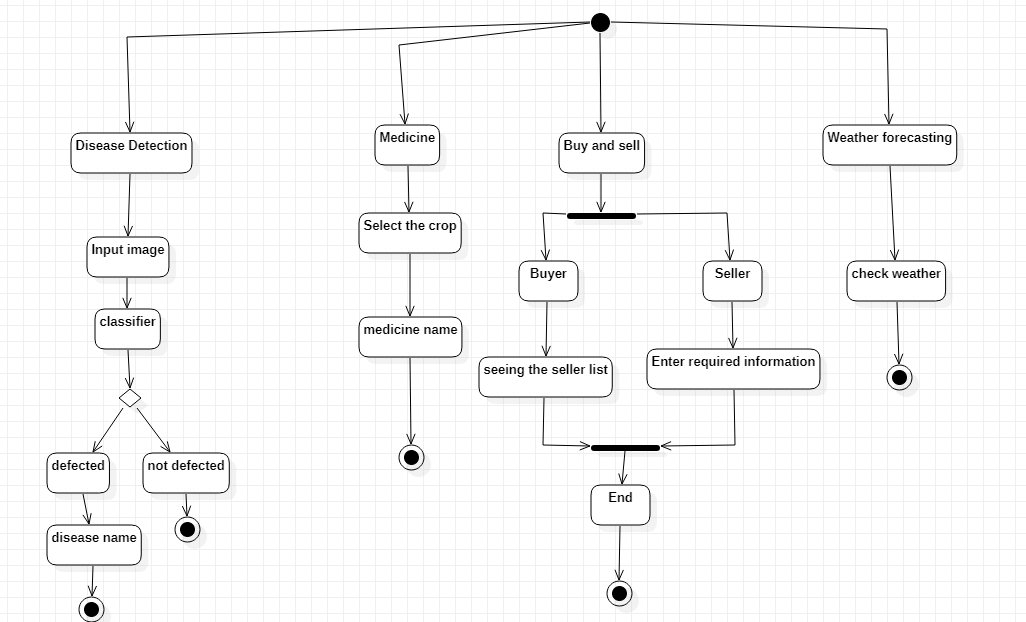


Figure 5: Activity Diagram

**Description of Activity Diagram**

The above Diagram telling us that we have 4 module:

* Disease Detection
* Medicine
* Buy/sell
* Weather forecasting

**Disease Detection:** In the Disease Detection, farmer have to input the image, the system will apply the classifier on it then it will decide that the crops have a disease or not. If it have a disease then what is the disease name

**Medicine:** In the medicine, farmer select the crop that will be mentioned in the application, then after selecting the medicine the medicine name of the disease that we working on it will be shown

**Buy/Sell:** In the Buy/sell, two option will appear buy and sell. In the sell option farmer have to enter the required information and that information will be shown in buy option

**Weather forecasting:** In weather forecasting, the farmer can check the weather

* + 1. **Class Diagram**

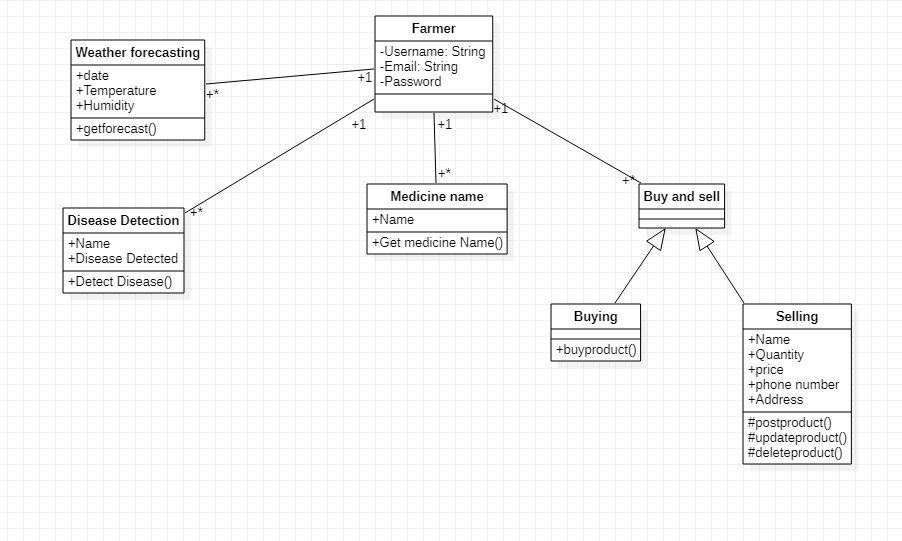


Figure 6: Class Diagram

**Description of Class Diagram:**

The above diagram is telling us that we have main class which is connected to other class. The main class is Farmer which has a relation of many to many relation with weather forecasting, many to many relation with Disease Detection class, many to many relation Medicine name, and Buying/Selling. In buying/selling class, Buying/Selling is a super class and the buying and selling classes are the subclass of the super class. The relationship between them is generalization

* + 1. **Sequence Diagram**

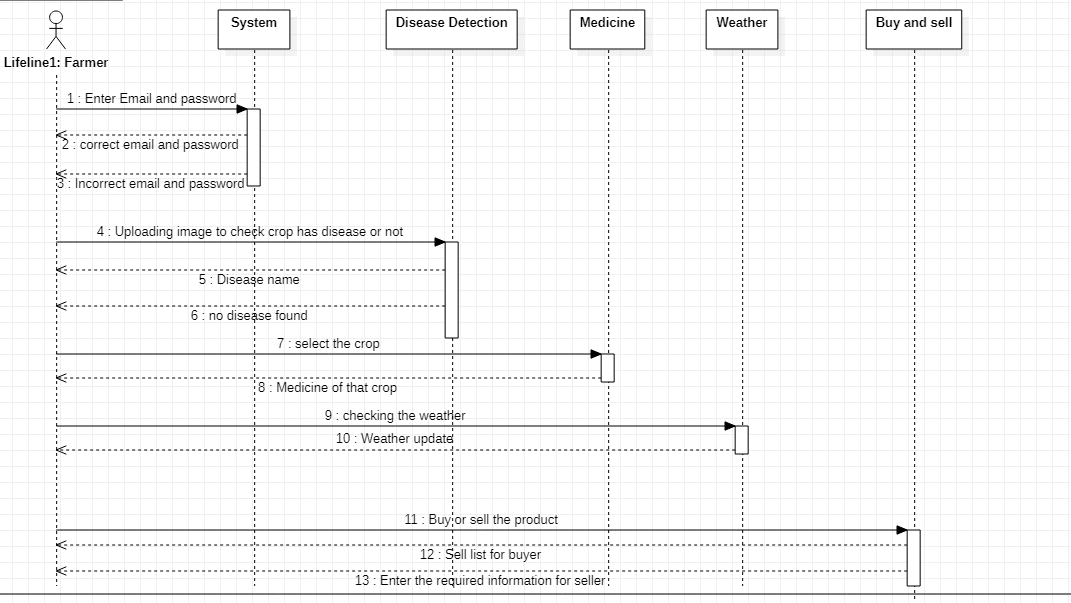


Figure 7: Sequence Diagram

**Description of Sequence Diagram**

The above diagram showing the sequence of our application, like first Farmer has to login into our Application to use it. After successfully login into our application farmer has now access to our application. After login the home screen will pop up. In home screen all the option will shown:

* **Disease Detection:** In the disease detection, Farmer has to upload image to find the disease name
* **Medicine:** Farmer can check the medicine name of the crops disease. By selecting the crops name and the medicine of the disease that we working on it will be shown
* **Weather:** In weather, farmer can check the weather update
* **Buy/sell:** In buy/sell, farmer can sell the product that they want and the product the farmer want to sell will be shown in the buy option
  + 1. **State Transition Diagram**

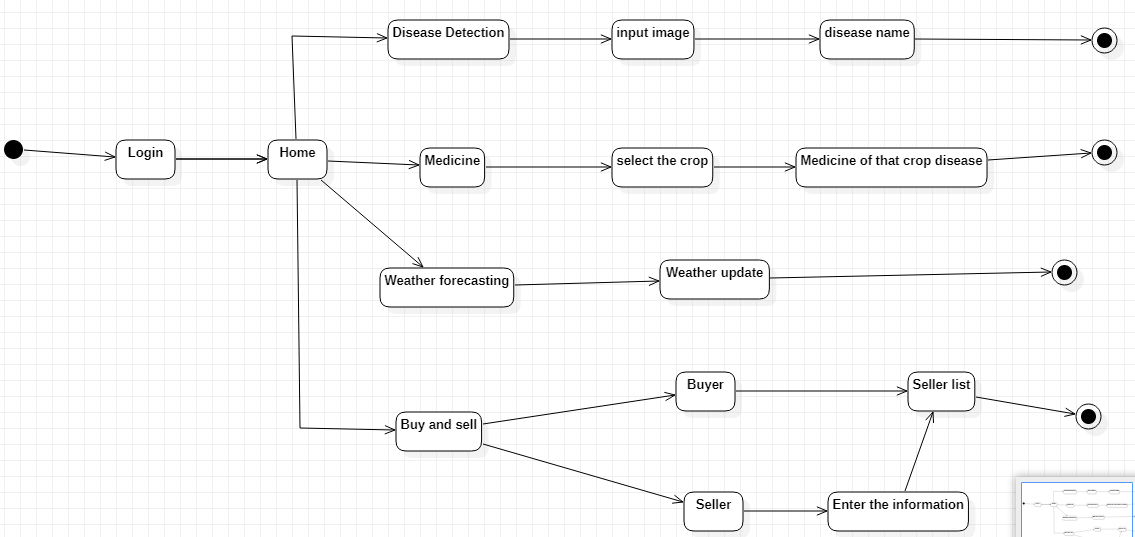


Figure 8: State Transition Diagram

**Description of State Transition Diagram**

The above diagram telling us that firstly farmer will be in login state, after successfully login it will move to home state. In home state four state will be shown, now it is the farmer choice that at which state he want to go.

The following state that will appear are:

* **Disease Detection state:** In the disease detection state, farmer will move toward to input image state, after inputting the image the farmer will get the disease name in the disease name state.
* **Medicine state:** In the medicine state, Farmer will move toward to select the crop state, In this state Farmer will select the crop, and then farmer will move toward to medicine of that crop disease state. In this state, Medicine of the disease that we are working on it will be shown.
* **Weather forecasting state:** In weather forecasting state, Farmer will move toward to weather update state. In this state farmer will get the weather update.
* **Buy/sell state:** In Buy/sell state, the farmer have the choice that weather he want to go to buyer state or seller state. In the buyer state, Farmer will move toward to seller list state where the information of seller will be shown. If the farmer is in the seller state, it will move toward to enter the information state, where farmer enter the information and that information will be shown into the seller list state.

## Data Design

Firebase database we are using, in this database the major data will be stored, and the data will be retrieved in buying and selling module

**4.3.1 Data Dictionary Login**

In login, we are going to use variable

* **Email:** email used to store user email using use state
* **Password:** password used to store user password using use state
* **Click me:** function to authenticate the user
* **Button:** To authenticate and navigate the user

**Sign up**

In Sign up, we are going to use variable

* **Email:** email used to store user email using use state
* **Password:** password used to store user password using use state
* **Sign up:** function to store user credentials in the database to create an account
* **Button:** for navigation

**Home screen**

In the Home screen, we are going to use variable

* **Disease detection:** By clicking it will navigate to the disease detection page
* **Medicine name**: By clicking it will navigate to the Medicine name page
* **Buying and selling:** By clicking it will navigate to Buying/selling page
* **Weather forecasting:** By clicking it will navigate to the Weather forecasting page

## Human Interface Design

Firstly, the user has to login into the system, if the account is not created then the user has to create his account. After successfully creating the account the user can now have access to our application. After successfully login into our application the home page will appear, in this home page we have four option

* + - **Disease detection:** In Disease detection, the user can find out the disease name
    - **Buying and selling:** In Buying/selling option users can buy or sell their crops
    - **Medicine:** In Medicine, users can find out the medicine name of the disease crops
    - **Weather forecasting:** Weather forecasting users can see the weather update
    1. **Screen Images Login screen**

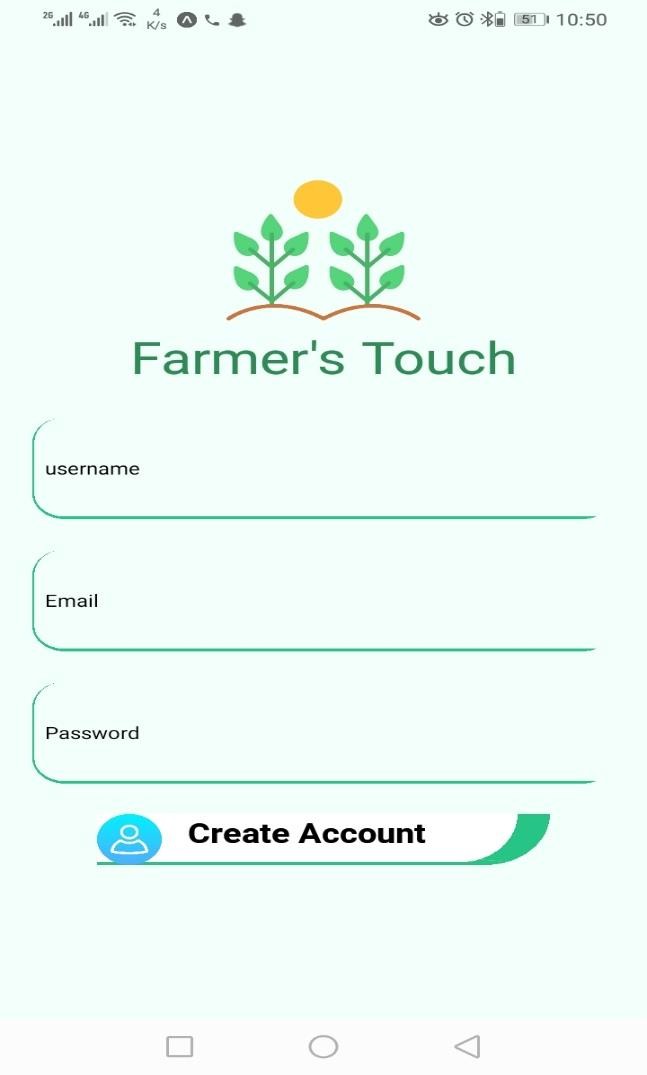
.

Figure 9: Login Screen

**Description**

When we open the application the first screen will appear it will be the login screen. In the login screen, we will enter the email and password, if the email and password are correct it will move to the home page otherwise not if there is a new user then the user will press the signup button to move to the signup screen

**Sign up Screen**



**Description**

Figure 10: Sign up Screen

In the signup screen, the user will create his account by entering a username, Email, and password to create his account

**Home screen**

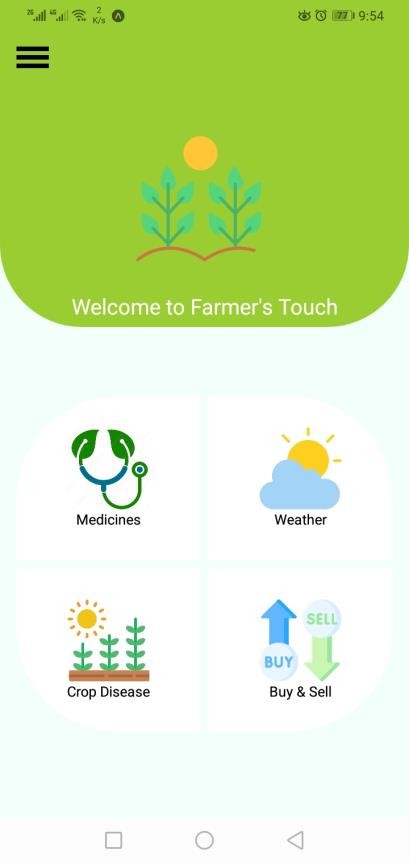


Figure 11: Home screen

* + 1. **Screen Objects and Actions Screen object**
* **Dashboard:** A Home screen that provides some functionalities
* **Setting:** In the setting, the user can see his profile and log out of his account
* **Disease detection:** In Disease detection, Farmer can see the disease name
* **Buying/selling:** In Buying/Selling, Farmer can buy/sell their product
* **Medicine:** In Medicine, Farmer can see the medicine name
* **Weather forecasting:** In Weather forecasting, Farmer can see the weather update

**Action**

* **Crop disease detection:** Farmers can upload an image of crops that are affected by the disease to see the disease name
* **Weather:** In the weather, Farmer can see the weather update
* **Buying/selling:** In Buying/selling, Farmer can buy/sell their crop
* **Medicine:** In Medicine, Farmer has to see the medicine name of the required disease

**CHAPTER # 05**

# Implementation

## Algorithm

Table 17: Algorithm

|  |
| --- |
| **Algorithm 1 Training the model** |
| **Input:** Image Dataset |
| **Output:** .h5 model |
| 1. import tensorflow as tf 2. from tensorflow.keras.applications import MobileNetV2 3. from tensorflow.keras.layers import Dense, GlobalAveragePooling2D 4. from tensorflow.keras.models import Model 5. from tensorflow.keras.preprocessing.image import ImageDataGenerator 6. train\_data\_dir = 'D:\Crop\_Disease\_Dataset\Train' 7. validation\_data\_dir = 'D:\Crop\_Disease\_Dataset\Valid' 8. num\_classes = 9 # Replace with the actual number of classes 9. train\_datagen = ImageDataGenerator( 10. rescale=1.0/255, 10. rotation\_range=20, 11. width\_shift\_range=0.2, 12. height\_shift\_range=0.2, 13. shear\_range=0.2, 14. zoom\_range=0.2, 15. horizontal\_flip=True, 16. fill\_mode='nearest'   18. )   1. validation\_datagen = ImageDataGenerator(rescale=1.0/255) 2. train\_generator = train\_datagen.flow\_from\_directory( 3. train\_data\_dir,   22. target\_size=(224, 224),   1. batch\_size=8, 2. class\_mode='categorical'   25. )   1. validation\_generator = validation\_datagen.flow\_from\_directory( 2. validation\_data\_dir,   28. target\_size=(224, 224),   1. batch\_size=8, 2. class\_mode='categorical'   31. )   1. base\_model = MobileNetV2(weights='imagenet', include\_top=False) 2. x = GlobalAveragePooling2D()(base\_model.output) 3. x = Dense(1024, activation='relu')(x) 4. predictions = Dense(num\_classes, activation='softmax')(x) 5. model = Model(inputs=base\_model.input, outputs=predictions) 6. optimizer = tf.keras.optimizers.Adam(lr=0.001) 7. model.compile(optimizer=optimizer, loss='categorical\_crossentropy', metrics=['accuracy']) 8. epochs = 50 9. model.fit( 10. train\_generator, 11. steps\_per\_epoch=len(train\_generator), 12. validation\_data=validation\_generator, 13. validation\_steps=len(validation\_generator), 14. epochs=epochs   46. )  47. model.save('crop\_disease\_model.h5') |

## External APIs/SDKs

Describe the third-party APIs/SDKs used in the project implementation in the following table

Table 18: Details of APIs used in the project

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of API and version** | **Description of API** | **Purpose of usage** | **List down the API endpoint/function/class in**  **which it is used** |
| Open weather | Used for getting weather update | For checking daily weather | https://api.openweathermap.or g/data/2.5/forecast?lat=${latitu de}&lon=${longitude}&appid  =${apiKey}&units=metric |

## User Interface

* + 1. **Login Screen**



Figure 12: Login Screen

**Description**

In login screen we have to enter the Email Address and Password to navigate to the home page and if there is new Customer who want to use our app he can simply create there account after clicking on the signup button

* + 1. **Signup Screen**

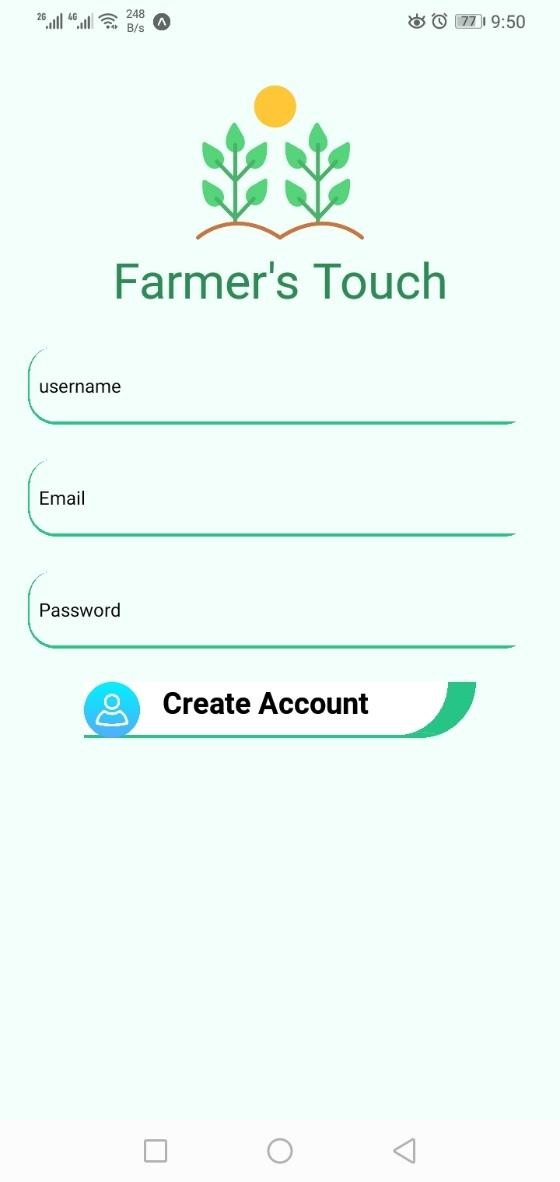


Figure 13: Signup Screen

**Description**

After clicking on the sign up in the login page this screen will popup where we have to enter Email Address, Password and username for creating the account

* + 1. **Home Screen**

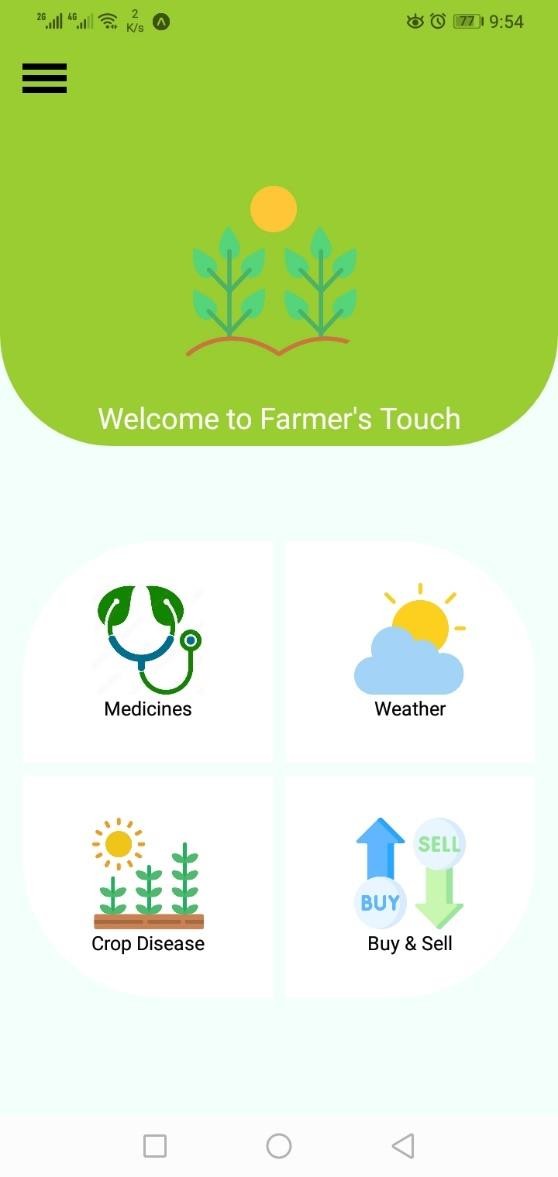


Figure 14: Home Screen

**Description**

After Entering correct Email Address and password in login Screen we will navigate to home page where all the module that we are working on in this app will shown

* + 1. **Medicine**

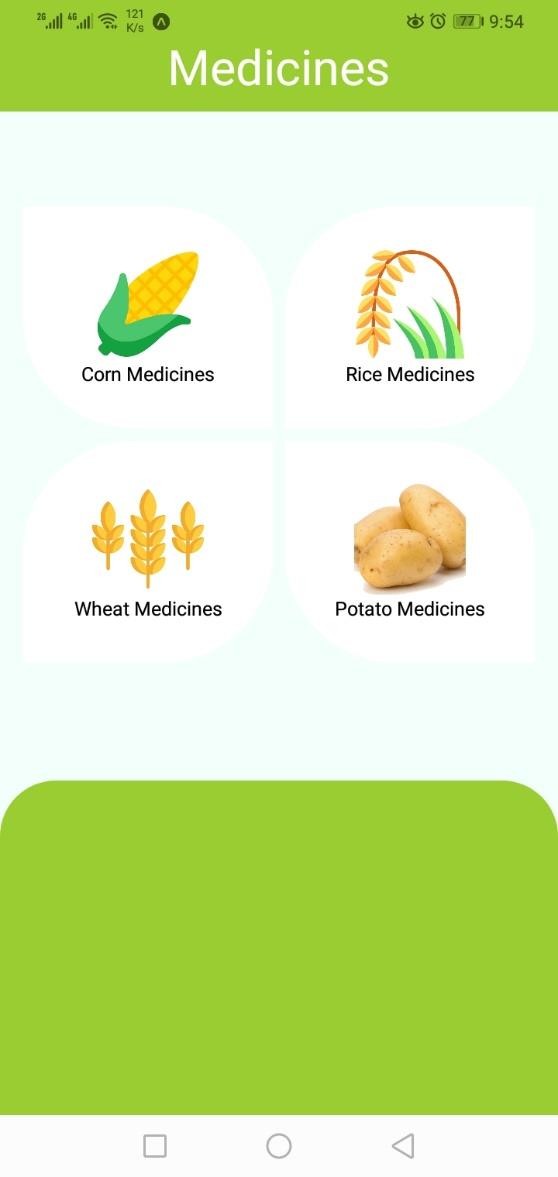


Figure 15: Medicine Screen

**Description**

After clicking on medicine, this screen will popup where we till which crop disease medicine we want either it is wheat, potatoes, corn or rice

* + 1. **Corn medicine**



Figure 16: Corn Medicine Screen

**Description**

After clicking on any Corn crop medicine this kind of interface will be shown where it tell the customer the medicine name of the crop disease and how we apply it

* + 1. **Rice medicine**



Figure 17: Rice Medicine Screen

**Description**

After clicking on any Rice crop medicine this kind of interface will be shown where it tell the customer the medicine name of the crop disease and how we apply it

* + 1. **Potatoes Medicine**



Figure 18: Potatoes Medicine Screen

**Description**

After clicking on any potatoes crop medicine this kind of interface will be shown where it tell the customer the medicine name of the crop disease and how we apply it

* + 1. **Wheat Medicines**



Figure 19: Wheat Medicine Screen

**Description**

After clicking on any Wheat crop medicine this kind of interface will be shown where it tell the customer the medicine name of the crop disease and how we apply it

* + 1. **Weather**

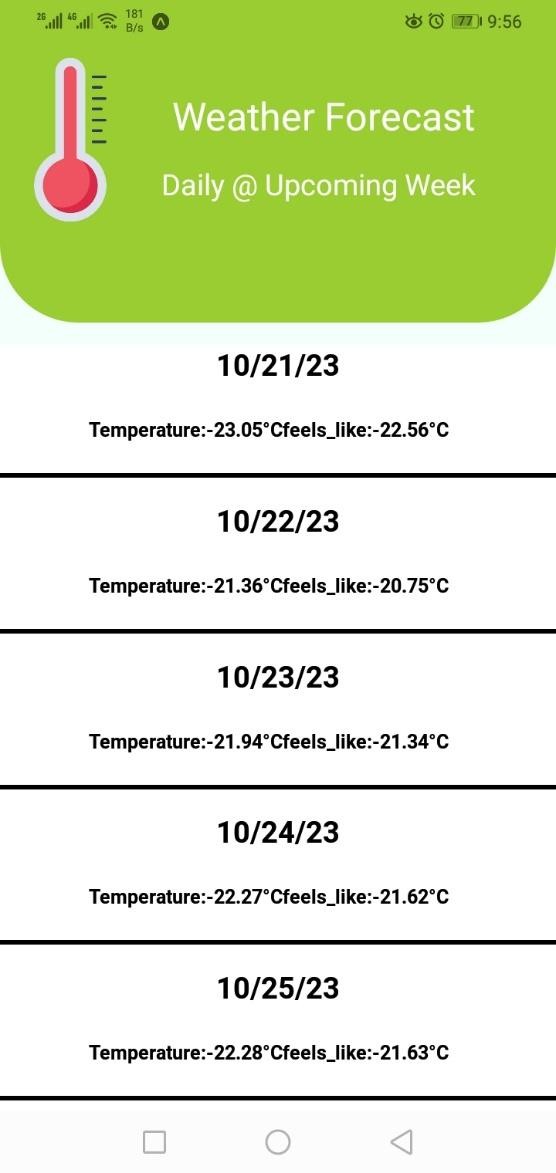
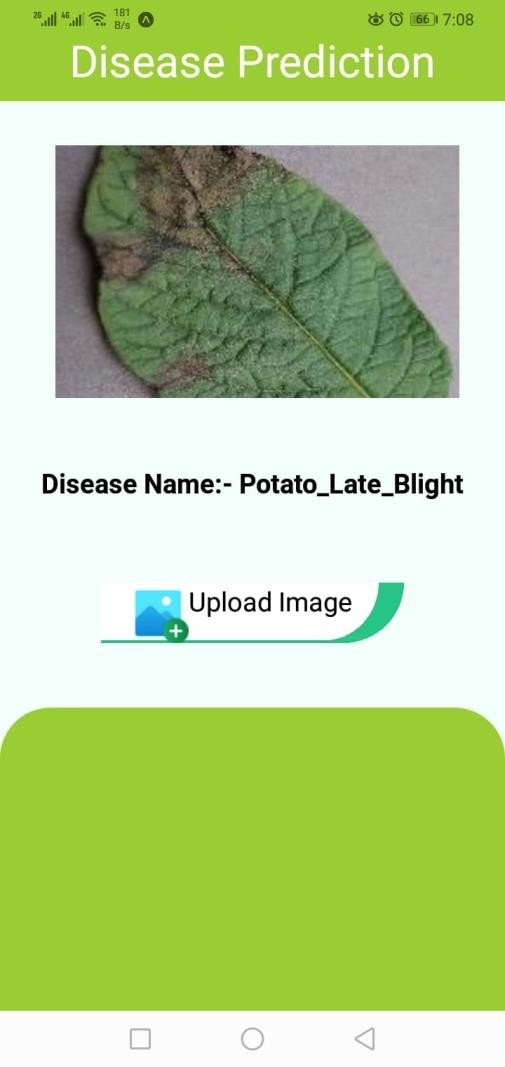


Figure 20: Weather Screen

**Description**

After clicking on weather module in home page, this kind of screen will be shown where it tell the weather of 6 days of the current location

* + 1. **Crop Disease Detection**



**Description**:

Figure 21: Crop Disease Detection

In this module Farmer can detect the disease of the crops that we are working on it, by just uploading the image and get the disease name.

* + 1. **Buying/Selling**

In this module where Farmer can buy or sell their product easily from this app

#### Buying

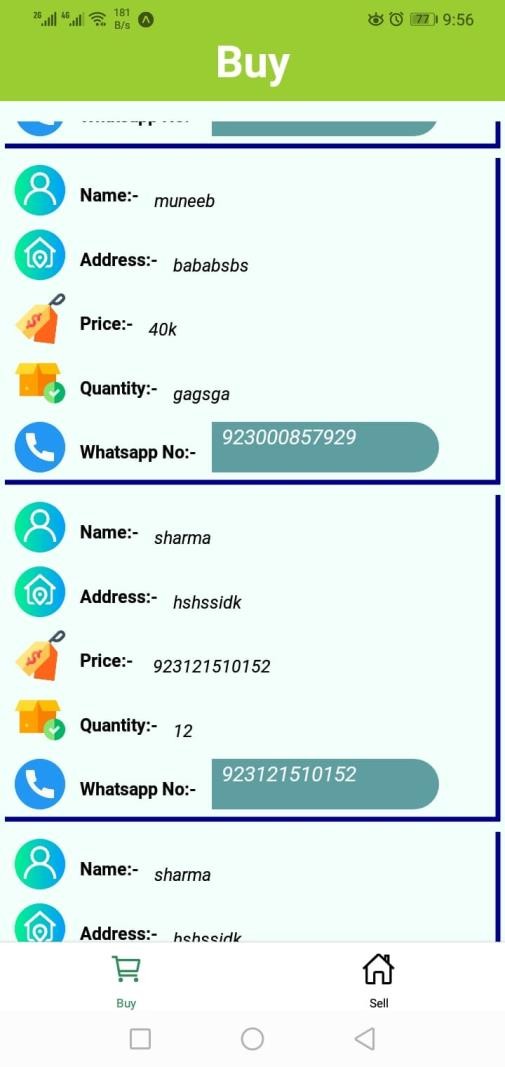


Figure 22: Buying screen

**Description**

After clicking on Buying interface, this kind of interface will be shown where the farmer can buy the product. In this interface, it will tell the customer the list of product that they can buy

#### Selling

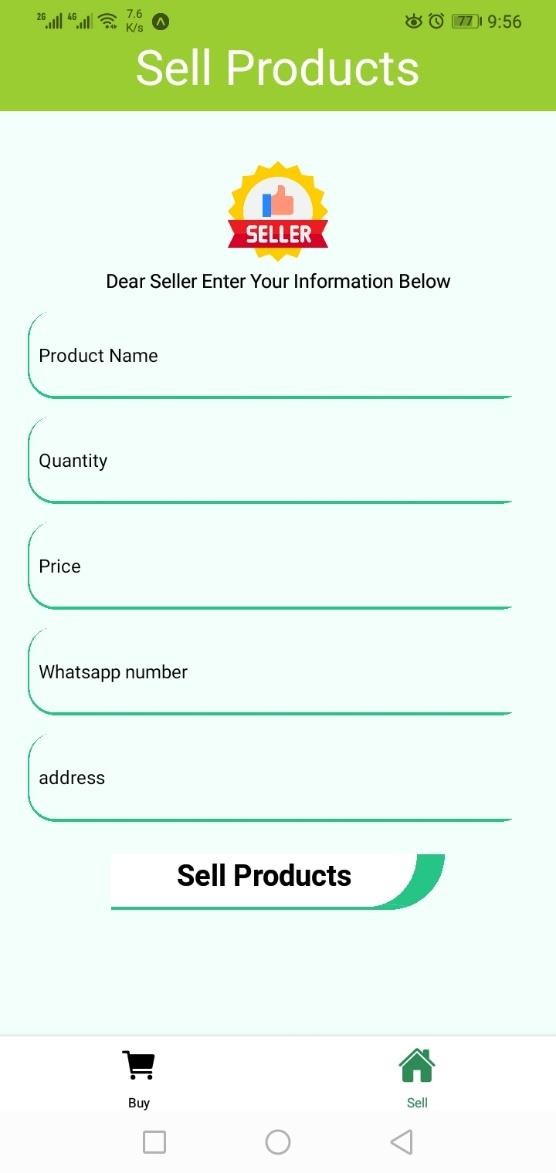


Figure 23: Selling Screen

**Description**

After clicking on Selling tab, this kind of interface will be shown where the farmer can sell their product. In this interface, the farmer can sell their product but filling all the information

## Deployment

**Mobile Application Deployment:**

* + - **Development Tools:** Visual Studio Code (VS Code) is used for mobile app development.
    - **Frameworks:** React Native is the framework used for building the mobile application.
    - **Versions:** 0.70

**Backend Server Deployment:**

* + - **Database:** Firebase is employed as the database for managing data.
    - **Development:** JavaScript, Python is chosen.
    - **Versions:** Python(3.10)

**CHAPTER # 06**

# Testing and Evaluation

## Unit Testing

It’s a level of software testing where individual units of a software/component are tested. The purpose is to validate that each unit of the software performs as designed.

**Unit Testing 1:** Login as Customer with valid and invalid Email/password

**Testing Objective:** To ensure the login form is working correctly with valid and invalid credentials/inputs.

Table 19: Unit Testing-1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check the Login page by entering invalid Email | Email: [abc123@gmail.com](mailto:abc123@gmail.com) Password:  123456 | Didn’t move to Home page | Pass |
| 2 | Check the Login page by entering invalid Password | Email: [usama123@gmail.com](mailto:usama123@gmail.com) Password:  123489 | Didn’t move to Home page | Pass |
| 3 | Check the Login page by entering invalid Email and password | Email: [abc123@gmail.com](mailto:abc123@gmail.com) Password:  123489 | Didn’t move to Home page | Pass |
| 4 | Check the Login page by entering valid Email and password | Email: [usama123@gmail.com](mailto:usama123@gmail.com) Password:  123456 | Move to Home page | Pass |

**Unit Testing 2:** Signup as New member with valid and invalid Email Format

**Testing Objective:** To ensure the Signup form is working correctly with valid and invalid Email Format

Table 20: Unit Testing-2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check the Signup page by entering invalid  Email Format | Email: [abc@123gmail.com](mailto:abc@123gmail.com) | Give error message of invalid  Email Format | Pass |
| 2 | Check the Signup page by entering valid Email  Format | Email: [usama123@gmail.com](mailto:usama123@gmail.com) | Successfully signup | Pass |

**Unit Testing 3:** Customer want to sell the product

**Testing Objective:** To ensure that in buying screen it accept the value and display it the selling the screen

Table 21: Unit Testing-3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check the Selling page that it is accepting all the value | Name Price Location  Phone Number | Save all the values | Pass |
| 2 | Checking the buying page that it is showing the list | Name Price Location  Phone Number | Showing the whole list | Pass |

## Functional Testing

In functional testing, the functionality of each of the module is tested. This is to ensure that the system produced meets the specifications and requirements.

**Functional Testing 1:** Login with as a member and not a member

**Objective**: To ensure that the login page is working correctly or not.

Table 22: Functional Testing-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Login as a member. | Email: [usama123@gmail.com](mailto:usama123@gmail.com) Password:  123456 | Home page is displayed | Logged in and Navigate to Home page. | Pass |
| 2. | Login as Not a member. | Email: [abc123@gmail.com](mailto:abc123@gmail.com) Password:  123489 | Home page is not displayed  . | Login failed – invalid credentials error | Pass |

**Functional Testing 2:** Customer want to check the Medicine name

**Objective**: To ensure that Name of the medicine of the crops that we are working on is displaying or not

Table 23: Functional Testing-2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Checking the medicine name | Opening the medicine name module | List of medicine should be displayed | List of medicine is displayed | Pass |

**Functional Testing 3:** Customer want to buy the product

**Objective**: To ensure the list of the product that they want to buy is shown or not

Table 24: Functional Testing-3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Checking the Product to buy | Opening the Buying/Selling module | List of product should be displayed in Buy tab | List of product is displayed in Buy tab | Pass |

**Functional Testing 4:** Customer want to sell the product

**Objective**: To ensure customer can easily sell the product by filling all the required text field

Table 25: Functional Testing-4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Want to Sell the product | Name Address Price Quantity  WhatsApp | Product should be listed in the Buy Tab | Product is listed in the buy tab | Pass |

**Functional Testing 4:** Customer want to check the weather update

**Objective**: To ensure customer can easily check the weather

Table 26: Functional Testing-4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Want to check the weather | Opening the weather module | Weather should be displayed | Weather is displayed | Pass |

**Functional Testing 4:** Customer want to check the disease name of the crops

**Objective**: To ensure customer can easily check the disease name of the crops that we mentioned above

Table 27: Functional Testing-4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Want to check the disease name of the crop | Uploading the disease crops image | The disease name should be displayed | The disease name is displayed | Pass |

## Business Rules Testing

The table contains conditions and actions are used for test cases where conditions as inputs and actions as outputs.

Here is the business rule testing of our project.

**Signup policy**

* + - Farmer fill form with all valid credentials will sign in.
    - Farmer with invalid valid credentials will not be able to sign in.

Table 28: Business Testing-1

|  |  |  |
| --- | --- | --- |
| Condition | Rule 1 | Rule 2 |
| Farmer with valid credentials. | Y | N |
|  |  |  |
| Sign in or login | Y | N |

**Disease detection policy**

* + - App will detect disease only when images are related to those that we are working on.
    - App will not detect disease without disease Images.
    - It will show Invalid message once you input without disease image.

Table 29: Business Testing-2

|  |  |  |  |
| --- | --- | --- | --- |
| Condition | Rule 1 | Rule 2 | Rule 3 |
| Input Disease image | Y | N |  |
| Input Simple Image |  |  | Y |
|  |  |  |  |
| App will detect Disease | Y | N |  |
| Display error message |  |  | Y |

**Final Decision Table:**

Table 30: Business Testing-3

|  |  |  |  |
| --- | --- | --- | --- |
| Condition | Rule 1 | Rule 2 | Rule 3 |
| Farmer is login | Y | Y | Y |
| Input Disease image | Y | N |  |
| Input Simple Image |  | Y |  |
|  |  |  |  |
| App will detect disease | Y | N |  |
| Display Invalid message | N | Y |  |

## Integration Testing

**Integration Testing 1:** Customer want to sell the product

**Testing Objective**: To ensure customer can easily sell the product by filling all the required text field

Table 31: Integration Testing-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Want to Sell the product | Name Address Price Quantity  WhatsApp | Product should be listed in the Buy Tab | Product is listed in the buy tab | Pass |

**Integration Testing 2:** Customer want to check the disease name of the crops

**Testing Objective**: To ensure customer can easily check the disease name of the crops that we mentioned above

Table 32: Integration Testing-2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Want to check the disease name of the crop | Uploading the disease crops image | The disease name should be displayed | The disease  name is  displayed | Pass |

**CHAPTER # 07**

# Conclusion and Future Work

This chapter concludes the project and highlights future work.

## Conclusion

A Farmer's Touch" is a mobile application designed to significantly enhance the farming experience for agricultural professionals. It empowers farmers with valuable tools and information to improve crop management, disease identification, weather forecasting, and even facilitates buying and selling of agricultural products. By allowing farmers to easily identify crop diseases through image recognition and providing weather updates, it can significantly contribute to increased crop yield, reduced losses, and overall efficiency in the agricultural sector. CNN model with a remarkable 98% accuracy in crop disease identification. The high accuracy of the CNN model ensures precise identification of crop diseases, enabling farmers to take targeted and timely actions to protect their crops. This precision contributes to increased crop yield and reduced losses.

## Future Work

However, there is still room for improvement and future development.

**Future Work:**

* + - **Expand Image Dataset:** Continuously update and expand the image dataset used for disease identification by ensuring that the app's image recognition capabilities become more comprehensive and accurate.
    - **Integration of Doctor Consultations:** This feature will help the farmers to connect with agricultural experts or plant pathologists who can provide real-time advice and guidance when crop diseases are detected. This feature can help farmers make informed decisions and receive professional assistance when needed, further improving crop health and yield.

**CHAPTER # 08**

# 8. References

1. A. Azam and M. Shafique, “Agriculture in Pakistan and its Impact on Economy―A Review,” *Int. J. Adv. Sci. Technol.*, vol. 103, pp. 47–60, Jun. 2017, doi: 10.14257/ijast.2017.103.05.
2. F. Zainab Jatoi, “Agriculture in Pakistan and its impact on Economic growth Institute of Business Management (IoBM) 2.”
3. [https://propakistani.pk/2022/03/31/pakistans-wheat-production-down-by-2-5-percent-](https://propakistani.pk/2022/03/31/pakistans-wheat-production-down-by-2-5-percent-over-the-last-year/#%3A~%3Atext%3DA%20high-powered%20Federal%20Committee%20on%20Agriculture%20noted%20on%2Cmuch%20as%2051.6%20percent%20over%20the%20last%20year) [over-the-last-year/#:~:text=A%20high-](https://propakistani.pk/2022/03/31/pakistans-wheat-production-down-by-2-5-percent-over-the-last-year/#%3A~%3Atext%3DA%20high-powered%20Federal%20Committee%20on%20Agriculture%20noted%20on%2Cmuch%20as%2051.6%20percent%20over%20the%20last%20year) [powered%20Federal%20Committee%20on%20Agriculture%20noted%20on,much%20as](https://propakistani.pk/2022/03/31/pakistans-wheat-production-down-by-2-5-percent-over-the-last-year/#%3A~%3Atext%3DA%20high-powered%20Federal%20Committee%20on%20Agriculture%20noted%20on%2Cmuch%20as%2051.6%20percent%20over%20the%20last%20year)

[%2051.6%20percent%20over%20the%20last%20year.](https://propakistani.pk/2022/03/31/pakistans-wheat-production-down-by-2-5-percent-over-the-last-year/#%3A~%3Atext%3DA%20high-powered%20Federal%20Committee%20on%20Agriculture%20noted%20on%2Cmuch%20as%2051.6%20percent%20over%20the%20last%20year)

[https://play.google.com/store/apps/details?id=com.growersedge.my&gl=US#:~:text=The](https://play.google.com/store/apps/details?id=com.growersedge.my&gl=US&%3A~%3Atext=The%20Growers%20Edge%20App%20is%20the%20best%20FREE%2Corganized%2C%20make%20informed%20decisions%2C%20and%20earn%20more%20money)

[%20Growers%20Edge%20App%20is%20the%20best%20FREE,organized%2C%20make](https://play.google.com/store/apps/details?id=com.growersedge.my&gl=US&%3A~%3Atext=The%20Growers%20Edge%20App%20is%20the%20best%20FREE%2Corganized%2C%20make%20informed%20decisions%2C%20and%20earn%20more%20money)

[%20informed%20decisions%2C%20and%20earn%20more%20money.](https://play.google.com/store/apps/details?id=com.growersedge.my&gl=US&%3A~%3Atext=The%20Growers%20Edge%20App%20is%20the%20best%20FREE%2Corganized%2C%20make%20informed%20decisions%2C%20and%20earn%20more%20money)

1. https://play.google.com/store/apps/details?id=pk.com.pakzarzameen.farmerapp
2. https://play.google.com/store/apps/details?id=agstack.gramophone&hl=en&gl=US
3. https://en.wikipedia.org/wiki/Agriculture
4. https:/[/www.nationalgeo](http://www.nationalgeographic.com/climate-change/how-to-live-with-it/crops.html)g[raphic.com/climate-change/how-to-live-with-it/crops.html](http://www.nationalgeographic.com/climate-change/how-to-live-with-it/crops.html)
5. https:/[/www.britannica.com/](http://www.britannica.com/plant/rice)p[lant/rice](http://www.britannica.com/plant/rice)
6. https:/[/www.britannica.com/plant/whe](http://www.britannica.com/plant/wheat)a[t](http://www.britannica.com/plant/wheat)
7. https:/[/www.britannica.com/plant/cor](http://www.britannica.com/plant/corn)n
8. https:/[/www.britannica.com/plant/pot](http://www.britannica.com/plant/potatoes)a[toes](http://www.britannica.com/plant/potatoes)
9. https://wheatworld.org/wheat-101/
10. https:/[/www.medic](http://www.medicalnewstoday.com/articles/318699)a[lnewstoday.com/articles/318699](http://www.medicalnewstoday.com/articles/318699)