Ali Arshad Sahi

01-134181-008

&

Muhammad Bilal Noor

01-134181-037

Fruit Disease Detection and Grading

Bachelor of Science in Computer Science

Supervisor: Mehroz Sadiq

Department of computer Science Bahria University, Islamabad

November 2021

Certificate

We accept the work contained in the report titled “Fruit Disease detection and Grading”, written by **Ali Arshad Sahi** and **Muhammad Bilal Noor (CS-S21-27)** as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by:

Supervisor: Madam Mehroz Sadiq

_____

Internal Examiner:

_____

External Examiner

_____

Project Coordinator: Dr Moazam

_____

Head of Department: Dr Arif-ur-Rehman

_____

Abstract

Fruit Doctor, a complete system that provides modern technology to the farmers for analyzing their fruit crop and helps farmers in increasing the yield. Fruit doctor diagnoses diseases in different fruits and provides the best way to deal with a particular disease problem. Fruit Doctor also updates the user about the current weather in the surrounding area. As farmers have to do a lot of farming related tasks, to organize those tasks, the system provides an interface that allows the user to write down their work. The system also allows the user to check whether a fruit is expired or not by checking the quality of the fruit which helps the users to identify the fruit is eatable or not.

Acknowledgement

Allah the almighty, who gave us the strength to work to complete thesis project on time and with the best possible quality and our family and friends who supported us in every step of life and mostly the past four years of university life. We would like to thank and sincerely acknowledge the help of our supervisor Madam Mehroz Sadiq whose complete guidance, support and encouragement gave us a real motivation in doing this project.

Our goals had been accomplished with the help and confidence given to us by her. We would also like to mention the faculty members who encouraged and helped us whenever we needed it.

We are also thankful to our parents who gave us the moral support we needed to accomplish our goals and without this moral support it would not had been possible for us to achieve these goals. We would also like to thank the people who helped us directly or indirectly in accomplishing these goals.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

ANN	Artificial Neural Network
CNN	Convolution Neural Network
OOP	Object Oriented Programming
SVM	Support Vector Machine
CCV	Color Coherence Vector
GLCM	Gray Level Co occurrence Matrix
SURF	Speeded-Up Robust Feature
DB	Database
SQL	Structured Query Language

Chapter 1

Introduction

1.1 Introduction

Did you know?

- 48% of labor force in Pakistan is directly engaged with agriculture.
- 70% people are related to agriculture directly or indirectly.
- Agriculture Industry's contribution towards Pakistan's GDP is a massive 25% which is highest among all the sectors.
- So, it is the major source of living or income.
- This dependency on agriculture industry is not prevalent in Pakistan but in many other countries. For example, in India 70% population relies on farming.

One of the major concerns for the agriculture industry worldwide is "Fruit Disease". It causes economies to bear losses of millions of dollars each year. According to the New York Times, a citrus fruit disease in Florida cost the state's economy a massive US $ 4.5 billion from 2006 to 2012 and claimed 8000 jobs.

This problem surely needs a solution. One solution is the traditional manual monitoring. But manual monitoring is an exhaustive, time and cost consuming, and an inconsistent solution which does not give satisfactory results all the time. Moreover, it always needs expert advice, where again there are chances of human error in disease detection.

So, there is a need of more reliable, consistent, accurate, cost effective and speedy solution which requires lesser human effort. This project aims at providing all those adjectives to in-field fruit monitoring process. The solutions developed so far for this problem mostly were developed in a similar fashion. Almost all of them focused on 4 steps, i.e., image acquisition, image segmentation, feature extraction and disease classification. Almost all of them used K-Means clustering technique for image segmentation. For feature extraction, some used Gray-Level Co-occurrence Matrix (GLCM) method, some used Speeded Up Robust Features (SURF) algorithm, some used Otsu's method. For disease classification, mostly Artificial Neural Network (ANN) or Support Vector Machine (SVM) was used, otherwise, fuzzy logic and inference was also used in a couple of projects.

This project going to undergo same 4 steps for disease detection, but the implementation is a little different. K-Means clustering method is used for image segmentation, Blob Analysis or Color Coherence Vector (CCV) to be used for feature extraction, and finally ANN/SVM concept is used for disease classification.

1.2 Brief Overview

Fruit doctor is an application not only for the farmers, but it is for the government food authorities as well as fruit farming related researchers. The application provides the latest features to all its stakeholders. The Desktop application is built on .Net Framework. The database used in this project is SQL Server Express DB that is the world largest use database. Overall, this Desktop application is built on the latest technologies. Many different latest technologies that are integrated in application like for the disease and grading prediction also latest machine learning algorithms are used in which top libraries are used like Keras, TensorFlow and pandas.

Latest tools for the development, debugging and testing purpose are been used. Visual Studio Code, Visual Studio 2019 and SQL Server Express are the main tools that are been used.

1.3 Problem Statement

Fruit Disease is one of the major concerns for the agriculture industries worldwide. Manual monitoring and expert advice during fruit growth period have been the traditional methodologies to combat fruit diseases. These methods are costly, time-consuming, exhaustive, and unsatisfactory. Many times, the farmer could not find the disease in early stages and later face a financial loss. The other important problem is that the farmer is not updated by the current diseases in agriculture. Through which farmers do not learn about the new seeds, method and techniques coming in the market for better the yield of the food crop.

1.4 Relevance to Course Modules

The course we studied is Artificial Intelligence and Data Science in which we learned python. So as our major work is also python related so through learning this course, we programmed our python code very efficiently.

As we used OOP methodology, the courses Programming fundamentals, OOP and Design Patterns really helped us to program and made our application efficient and optimal. Then the Visual Programming course really helped us in making the application and the Database. The Human Computer Interaction course really helps us to make a beautiful UI and UX for our application users.

1.5 Project Background

This project aims to shift the in-field fruit disease detection process from manual monitoring to automated monitoring. The main objective is to ensure time and cost-effective monitoring of fruits with lesser human effort and greater consistency and accuracy in fruit disease detection, which ultimately improves fruit yield.

1.6 Methodologies and Software Lifecycle for this Project

An incremental software process is be used to develop this project. Incremental approach towards software development allows you to divide your whole project into several increments. The reason that incremental process methodology is chosen is that this project is required to be delivered in parts, i.e., after 40% and after 100%. So, in the first increment, a 40% functionality was aimed, and a complete 100% functionality was aimed in the second increment. Modules wise, it would simply mean that if there are 5 modules, then 2 were completed in the first increment and 3 in the second one.

This project needs some libraries, frameworks and tools which provide some built-in components or classes. So, obviously, an object-oriented design methodology is used. .Net, which provides some built-in components, is used for the front end. Tensor Flow is used for the ML models, where most of the core functionality is implemented.

Chapter 2

Literature Review

2.1 Literature Review

(Dubey 2012) Diseases of organic produce cause financial misery and devastating problems for creation in the horticultural industry around the world. This paper proposes and tentatively approves answers to the discovery and classification of apple disease. The proposed approach based on image preparation is based on the basic steps that accompany it. The first step uses K-Mean's bunching strategy for image segmentation, the second step extracts best-in-class highlights from the fragmented image, and finally uses multi-class to classify the image. It is classified into one. Support vector machine. Our exploratory results show that the proposed arrangement can also radically improve the accurate localization and programmed sequence of apple disease. The order accuracy of the proposed placement reaches up to 93%.

(Gavhale 2014)This work shows a written audit aimed at identifying and verifying natural products for image processing systems. By capturing images of bio products, various calculations are available to separate components from the properties of natural products. With the help of this component, organic products are distinguished and evaluated according to natural products. This should be possible with various classifiers. The evaluation framework has the advantages of high accuracy, high speed, and simplicity. It take a decent position in identifying and reviewing areas of natural product quality. This work was going to think about the various types of calculations used in quality assessment to build calculations for recognizing and ordering organic products from profit images. Components such as morphological elements are used and shading can be removed. It is further used to distinguish the classes of organic products using the nervous system.

(Jhuria & Kum 2013) This study presents an ingenious approach to consistently identify and classify diseases of the natural product pomegranate. Diseased pomegranate plants have the special symptom of shaded spots that appear in pomegranate's natural products. Therefore, it is important to check during the pomegranate development stage and collection. The proposed framework is a productive module for detecting bacterial rot of pomegranate natural products, cercospora natural product staining, fruit rot, and alternaria natural product staining diseases. This overview describes the critical issues identified in disease detection and building a world-class disease containment system. Tests are performed to identify the type of illness and characterize the clinical picture to a degree based on severity.

(Khirade & Patil 2015) Recognizable evidence of plant disease is how to maintain unhappiness in yield and quantity of produce. Testing for plant diseases is understood as testing for externally detectable specimens found in plants. In addition, the detection of plant diseases is very important for sustainable agriculture. Physically investigating plant diseases is very difficult. It requires a great deal of work, knowledge of plant diseases, and, in addition, excessive preparation time. Therefore, image preparation is used to discover plant diseases. Disease identification includes means such as image acquisition, image preprocessing, and image fragmentation (including extraction and placement). In this article, I explained how to identify plant diseases based on leaf images. This article also described some divisions and highlight extraction calculations that are used as part of plant disease locations.

(Samajpati & Degadwala 2016)Abroad is undoubtedly expanding in various countries today. Bounty hunter organic products are outside alternative countries such as oranges and apples. Manually detecting contaminated organic products is surprisingly boring. The use of imaging strategies has a significant impact on the study of agriculture-based applications. In any case, the identification of natural product diseases using images remains dangerous due to the general change in skin tone of unmistakable organic matter. In this article, three typical contaminations of natural apples: apple scab, apple rot, and apple stains. The proposed philosophy based on image manipulation is removed from the test image with some prime shades and surface highlights. If the product is contaminated with any disease, the contaminated area is subdivided using the Kimplis grouping system. The accuracy of disease characterization is improved by a combination of levels.

(Singh 2015) The profitability of horticulture depends heavily on the Indian economy. This is one of the reasons why localization of plant diseases plays an important role in agriculture, as plant diseases are so common. If they are not properly cared for there, this can have real consequences for the plant and thus can affect the quality, quantity, or efficiency of the individual items. Detecting plant diseases through a programmed process saves a great deal of effort associated with inspecting crops in large homes and, in the early stages, shows signs of disease when appearing in plant clearing. It is useful because it detects it. In this paper, in addition to the calculation of the image splitting method used for programmed localization, grouping of plant leaf diseases and various disease ordering methods that can be used to identify plant leaf diseases. Here is an overview of. Image fragmentation, an important aspect of plant leaf disease detection, is completed using genetic calculations.

(Tichkule 2016) Agribusiness is a very inspiring and outdated profession in India. Since India's economy depends on agricultural production, it is imperative to pay close attention to food production. Pests such as infectious diseases, parasites and microbes make the quality and quantity of diseased plants unhappy. The loss of farmers continues. Therefore, proper plant care is essential for them. This paper provides an overview of the use of imaging techniques to identify various plant diseases. Working with images provides a more skilled approach to identifying diseases caused by parasites, microbes, or plant infections. The non-essential

perceptions of the eye to distinguish illnesses are not accurate. Overdose of pesticides causes dangerous and permanent illness to people because they are not legally washed. Abundance also takes advantage of the quality of plant supplements. It causes a huge loss of creation to the rancher. Therefore, distinguishing and ranking illnesses using image processing techniques is useful in rural applications.

2.2 Comparison Table of Literature Review

Table 2.1: Comparison Table of Literature Review

Title	Techniques	Parameter	Merits	Demerits
Detection and Classification of Apple Fruit Diseases using Complete Local Binary Patterns Shiv (Dubey 2012)	K – means clustering	Feature extraction, image segmentation	accurate detection and automatic classification of apple fruit diseases, accuracy achieved	Complex to solve
Unhealthy Region of Citrus Leaf Detection Using Image Processing Techniques (Gavhale 2014)	K- means Clustering, GLCM Algorithm, SVM	Accuracy, time consumption	Ensure healthier environment	reduces quantity and degrades quality of the agricultural products
Image Processing for Smart Framing: Detection of Disease and Fruit Grading(Jhuria & Kum 2013)	Neural Network, Back Propagation	Vector morphology	Monitor the plant during growth period	obtaining accuracy in detecting and classifying diseases is not tough
Plant Disease Detection Using Image Processing(Khirade & Patil 2015)	RELLIEF- F	Pre processing Segmentation	Accurately identify diseases	Time consuming as compare to SVM
Hybrid Approach for Apple Fruit Diseases Detection and Classification Using Random Forest Classifier (Samajpati & Degadwala 2016)	K- Means Clustering Histogram equivalence	Color feature Texture feature	Improve the performance of disease classification	Cannot applied on Multiple features
Detection of unhealthy region of plant leaves using Image	Genetic Algorithm	Image acquisition Segmentation	less computational efforts and	Recognition rate should be improved

Processing and Genetic Algorithm(Singh2015)			optimum results were obtained,	
Plant Diseases Detection Using Image Processing Techniques(Tichkule 2016)	K-means Clustering, Neural Network Back propogation GLCM Algorithm Water Shed Template matchingAlgorithm	Accuracy, time consumption	Potential to use in Agrobot system	obtaining accuracy in detecting and classifying diseases is not tough

Chapter 3

Requirement Specification

3.1 Existing System

The solutions developed so far for this problem mostly were developed in a similar fashion. Almost all of them focused on 4 steps, i.e. image acquisition, image segmentation, feature extraction and disease classification. Almost all of them used K-Means clustering technique for image segmentation. For feature extraction, some used Gray-Level Co-occurrence Matrix (GLCM) method, some used Speeded Up Robust Features (SURF) algorithm, some used Otsu's method. For disease classification, mostly Artificial Neural Network (ANN) or Support Vector Machine (SVM) was used, otherwise, fuzzy logic and inference was also used in a couple of projects. There is only one such system in a mobile application which is freely available on the Web app store. The analysis for this system is provided in the table below.

Application Name	Weakness	Proposed Project Solution
Plantix	It has very limited features. i.e. it does not consider fruits other than apples. It is a crop-oriented application rather than a fruit- oriented application considering only one fruit crop.	The proposed project consider at least 3 fruits in its first version and it is a fruit-oriented application.
	There are weaknesses in the functionality. i.e. if a diseased apple's picture is taken, it gives results for tomato diseases.	Functional mistakes like correct prediction of the object i.e. an apple, but wrong classification of the disease i.e. of a tomato, is definitely not be there.
	It gives more than one results for the picture of a diseased fruit and then asks the user to select one according to the symptoms that his fruit is showing. Thus, there is human dependence. Humans still cannot always be accurate in deciding which symptoms their fruit has	The proposed project reduce human dependency as much as possible. The project aim at providing singular and the most probable output.

	developed or is developing.	

3.2 Functional Requirement

3.2.1 Functional Requirement 1

Table 3.1: Description of FR-1

Identifier	FR-1
Title	Login
Requirement	If a new user drops in on the system, then the system shall display the authentication option, which includes login with a local account, if one exists.
Source	User (All classes)
Rationale	The rationale is to take the user through the possible login options and to ensure that if someone is using the application, then he/she is a verified user.

3.2.2 Functional Requirement 2

Table 3.2: Description of FR-2

Identifier	FR-2
Title	Sign up
Requirement	If a user wants to sign up, then the system shall show option for signing up in the system using a local account.
Source	User (All classes)
Rationale	The aim is to show the user different ways to register themselves on this application and get the credentials which they can use to login to the application.
Priority	Medium

3.2.3 Functional Requirement 3

Table 3.3: Description of FR-3

Identifier	FR-3
Title	Reset password
Requirement	If a user enters wrong password, then the system shall show a "forgot password?" clickable text to the user and shall take the user through password reset process.
Source	User (All classes)
Rationale	The aim is to provide the user a way to reset their password if they forget it somehow.
Priority	Low

3.2.4 Functional Requirement 4

Table 3.4: Description of FR-4

Identifier	FR-4
Title	Sign out
Requirement	The user shall be able to sign out of their current account if they want to login to another account or if they want to sign out for a random reason.
Source	Users (All classes)
Rationale	The aim is to allow the user to be able to sign into the app using different accounts if they so wish to. They can do this by logging out of their current account and signing in again with a different one.
Priority	Low

3.2.5 Functional Requirement 5

Table 3.5: Description of FR-5

Identifier	FR-5
Title	Show user profile
Requirement	The user shall be able to see their profile on the app if they are logged in
Source	Users (All classes)
Rationale	The aim is to allow the user to be able to see their profile data which includes their display name, display picture, email address and change password.
Priority	Low

3.2.6 Functional Requirement 6

Table 3.6: Description of FR-6

Identifier	FR-6
Title	Edit user profile
Requirement	The user shall be making changes to their existing profile if they are logged in
Source	Users (All classes)
Rationale	The aim is to allow the user to be able to make changes to their current profile
Priority	Low

3.2.7 Functional Requirement 7

Table 3.7: Description of FR-7

Identifier	FR-7
Title	Detect disease
Requirement	If the user clicks on “Detect Disease”, then the system shall take the user through the process of disease detection for a fruit, i.e. taking a clear picture of a fruit and showing diagnosis results.
Source	Farmers, Fruit buyers and sellers
Rationale	The user can see a disease in the fruit crop and can prevent the rest of the crop from that disease by getting the appropriate medicine suggestions from the app.
Priority	High

3.2.8 Functional Requirement 8

Table 3.8: Description of FR-8

Identifier	FR-8
Title	Take a picture of the fruit for disease detection
Requirement	If the user clicks on “Detect Disease”, then the system shall take the user to the first step in fruit disease detection, i.e. taking a clear picture of the fruit, and should allow the user to upload the image from the system of already existing images.
Source	Farmers, Fruit buyers and sellers
Rationale	The aim is to allow the user to upload an image, so that they can detect it for diseases.
Priority	High

3.2.9 Functional Requirement 9

Table 3.9: Description of FR-9

Identifier	FR-9
Title	See diagnosis results
Requirement	Once the user provides a clear image of the fruit, the system shall process the image to detect any diseases. If a disease is found, then the system should display the diagnosis to the user.
Source	Farmers, Fruit buyers and sellers
Rationale	The user can see a disease in their fruit crop and can prevent the rest of the crop from that disease by getting the appropriate medicine suggestions from the application.
Priority	High

3.2.10 Functional Requirement 10

Table 3.10: Description of FR-10

Identifier	FR-10
Title	Get Medicine
Requirement	After the user has seen the diagnostic results for the disease of the fruit, and clicks on get medicine, then the system shall show all the possible solutions for the detected disease.
Source	Farmers
Rationale	After the user knows the disease their fruit crop has got, they should be able to know how the disease can be cured and can be stopped from spreading further.
Priority	High

3.2.11 Functional Requirement 11

Table 3.11: Description of FR-11

Identifier	FR-11
Title	Disease estimation and fruit grading
Requirement	The users shall be able to take picture of fruit and after system's processing on an image user should be able to see how much the fruit is affected by the disease and see the quality grading of that fruit.
Source	Farmers, Fruit Buyers and sellers
Rationale	The aim is to basically provide an ease especially for fruit buyers and sellers to agree and decide on a price according to the quality of the fruit, where none of the two have any problems. Also, it is aimed to help fruit buyer to check if the fruit is consumable or not.
Priority	High

3.2.12 Functional Requirement 12

Table 3.12: Description of FR-12

Identifier	FR-12
Title	Get Weather Updates
Requirement	According to the user location the system shall updates user about the weather condition of the city.
Source	Users (All classes)
Rationale	The aim is to facilitate farmers so they should be updated by the weather conditions and take measure while doing farming according to weather. This helps them to manage their work of farming with respect to weather conditions
Dependencies	None
Priority	Medium

3.3 Non-Functional Requirement

To consider all the targeted users, Fruit Doctor focuses on providing each one of the stakeholders an excellent experience with the system. For that purpose, the non-functional requirements that were consider are following.

3.3.1 Usability

USE-1: A 95% of the intended user population should express positive comments about a specific functionality while using the tools

USE-2: Equal to or less than 5% of the intended user population should misinterpret the information provided by a display

USE-3: Average usability rating from the users should not be less than 4.

3.3.2 Performance

PER-1: When a picture of a fruit is taken, it should not take more than 5 seconds from the time when the user uploads the picture to the time when the user gets the results over a stable machine.

3.3.3 Modifiability

MOD-1: Modifying any module should not take more than 2 days for 5-member maintainability team.

3.3.4 Consistency

CON-1: The system shall use the same color scheme for all the user interfaces in the application.

CON-2: The system shall use the same font type in all the user interfaces in the application.

3.3.5 Availability

AVA-1: The users should get the desired service from the system 99/100 times. It’s near to impossible to ensure 100% availability.

3.3.6 Simplicity

SIM-1: A user should be able to completely comprehend the logical flow and the design of the system in its first 3 uses at maximum.

Chapter 4

Design

4.1 System Architecture

This is a Desktop application developed using .Net Framework. This desktop application is our first major component. The second major component is the Business Layer. The third major component is the SQL Server Express database. It has already been explained in the earlier sections of this document about how these individual components collaborate but let us have a pictorial overview of everything explained earlier in the following diagram.

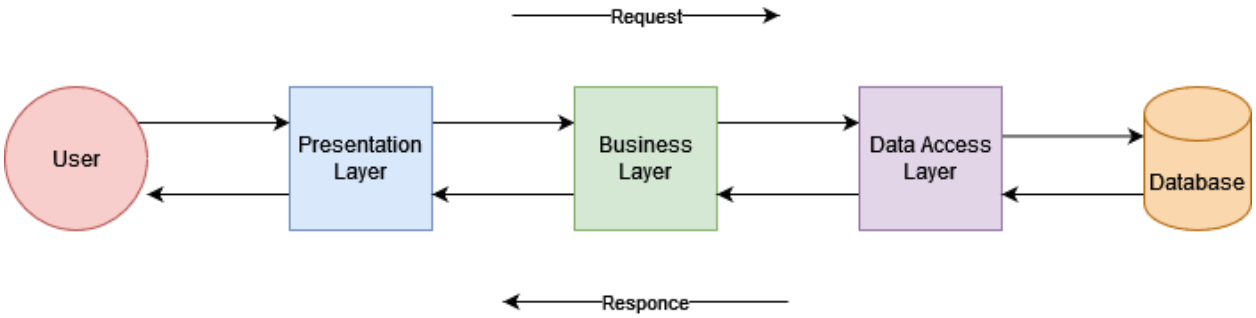

Figure 4.1: System Architecture Diagram

The Architectural style that our application reflects is three tier architecture, with the Windows Forms being responsible for all the presentation tier activities, the Business Logic Layer being responsible for application processing and finally the Data Access layer being responsible for accessing data from SQL Server Express database.

4.2 Data Representation

User's data is being stored in a user model in the SQL Server Express database, then in a list of strings for now in the Desktop application, which transformed later into a user class, to provide a consistent mapping between the objects that exist on the database.

For permanent storage of user data, a persistent SQL Server Express database is used, and for some functionality, like storing user's notes.

4.3 Process Flow/Representation

All the process flows are demonstrated in the activity diagram section.

4.4 Design Models

4.4.1 Activity Diagram

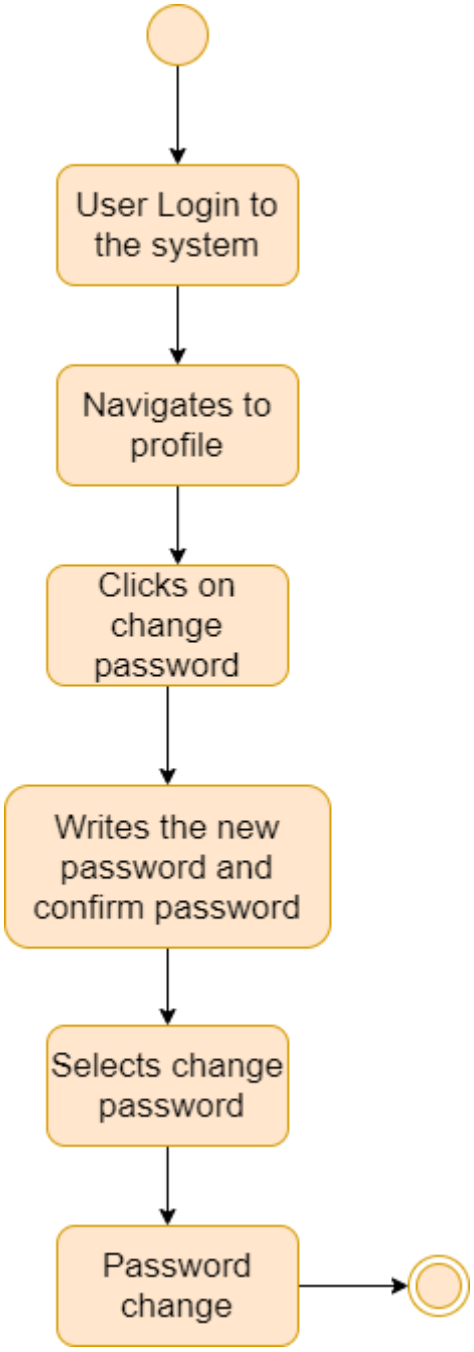

Figure 4.2 Password Update

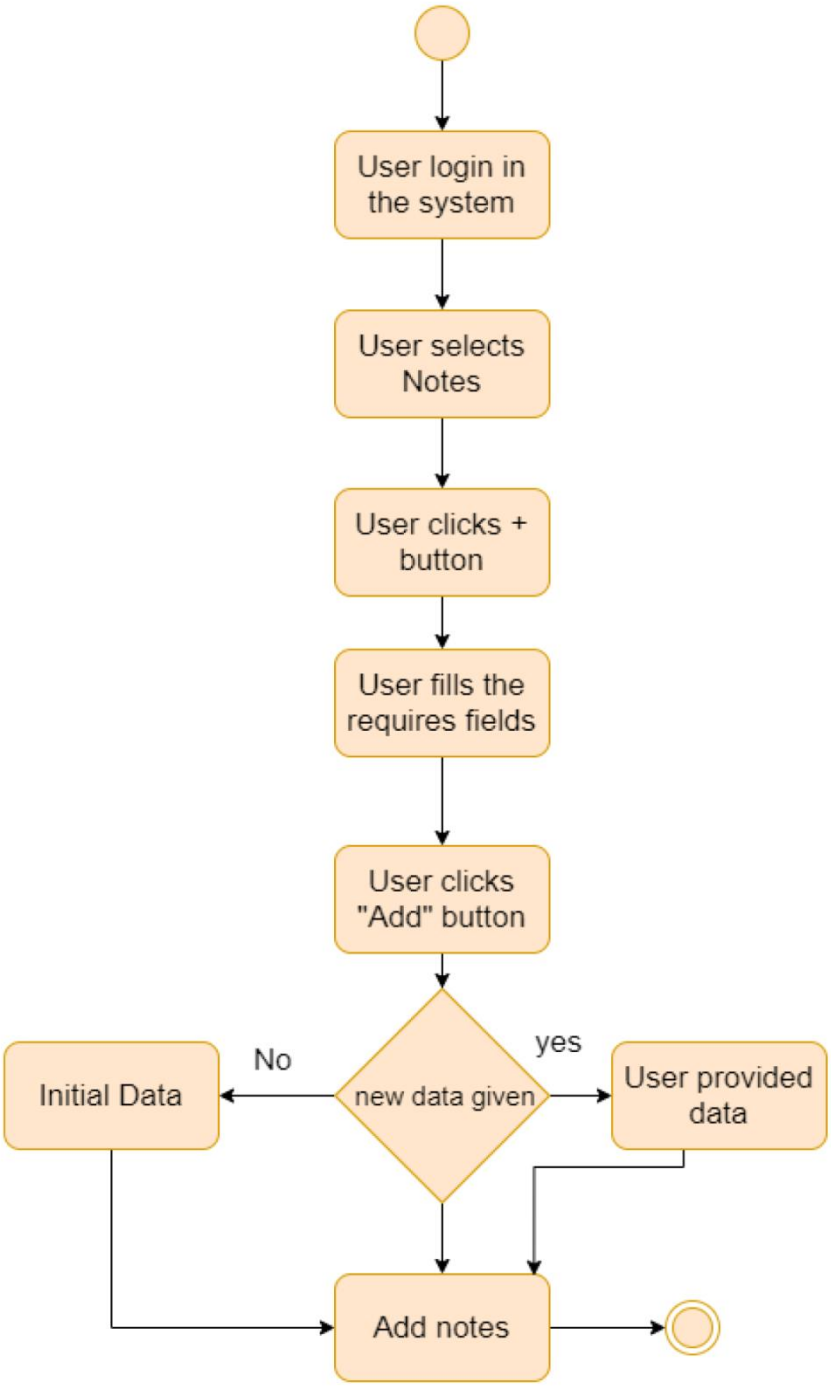

Figure 4.3: Add Notes

4.4.2 State Machine Diagram

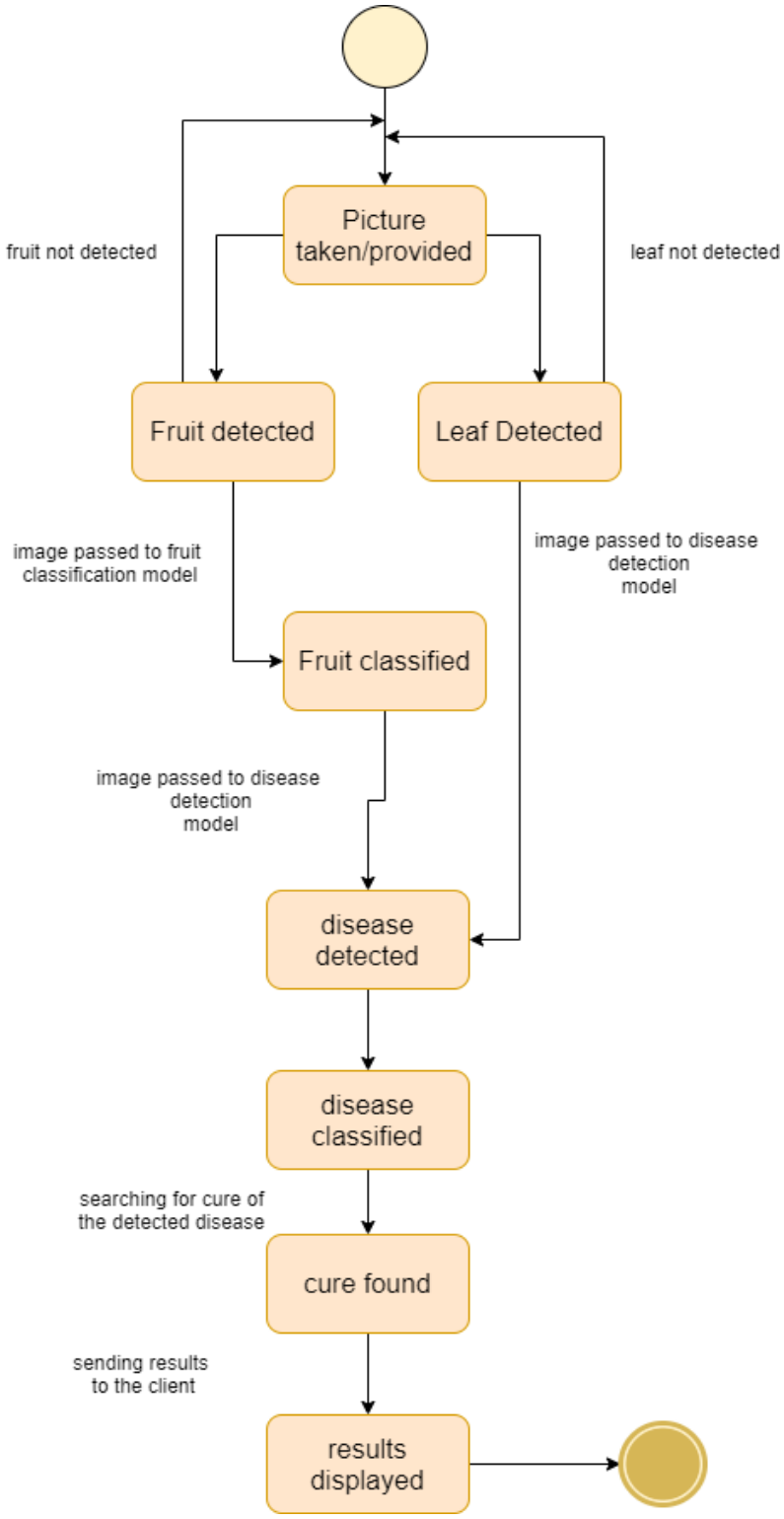

Figure 4.4: Disease Detection

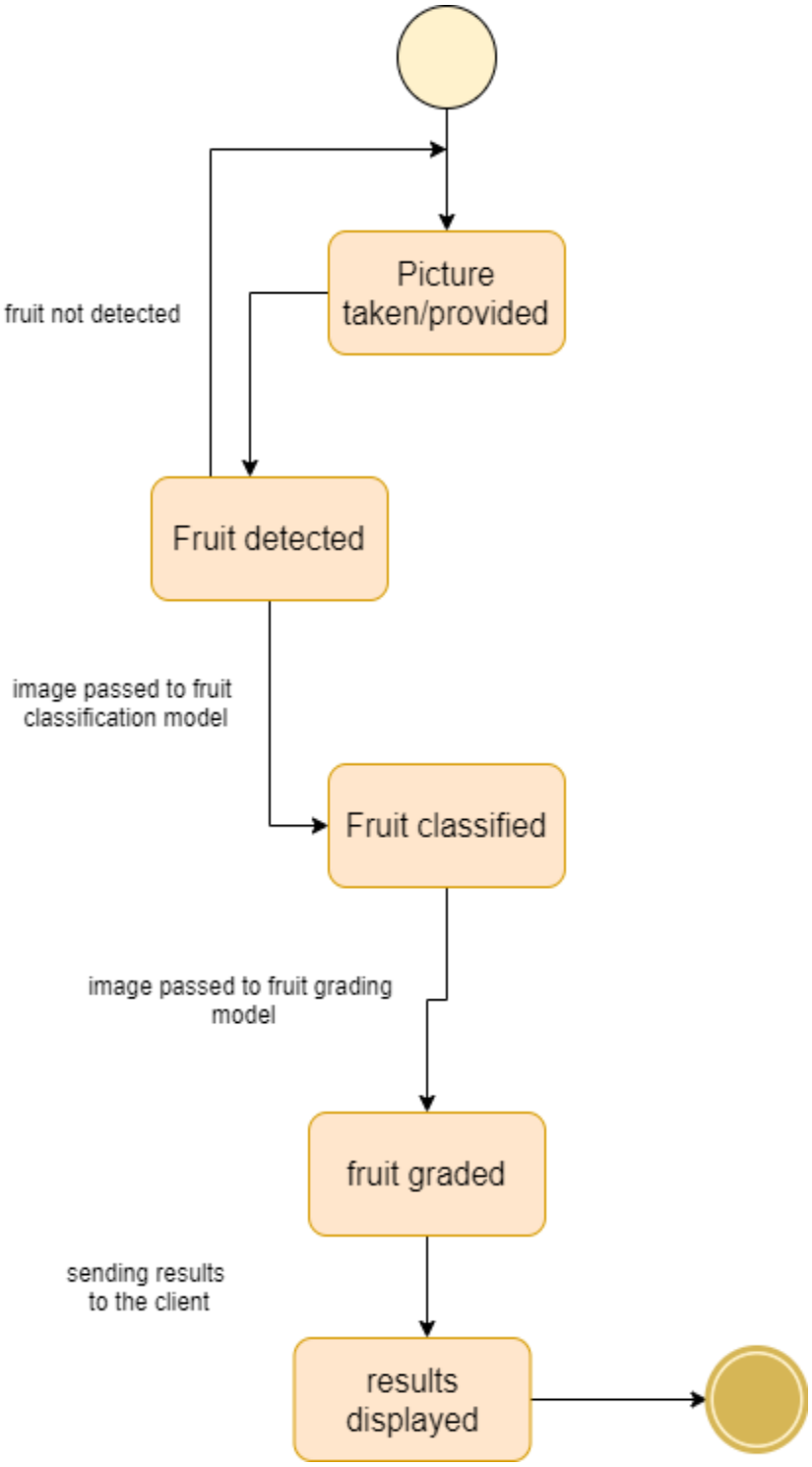

Figure 4.5: Fruit Grading

4.4.3 Sequence Diagram

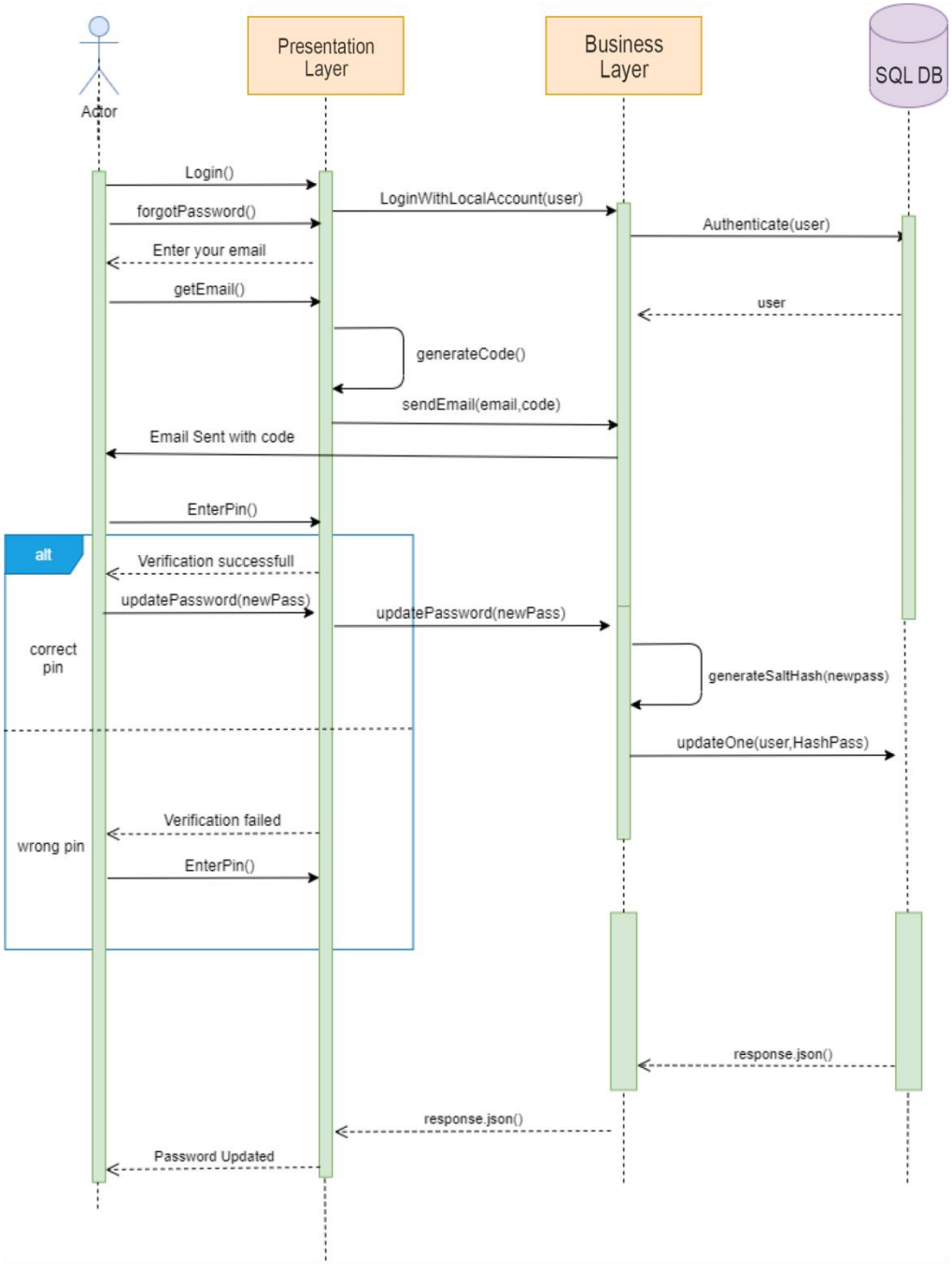

Figure 4.6: Update Password

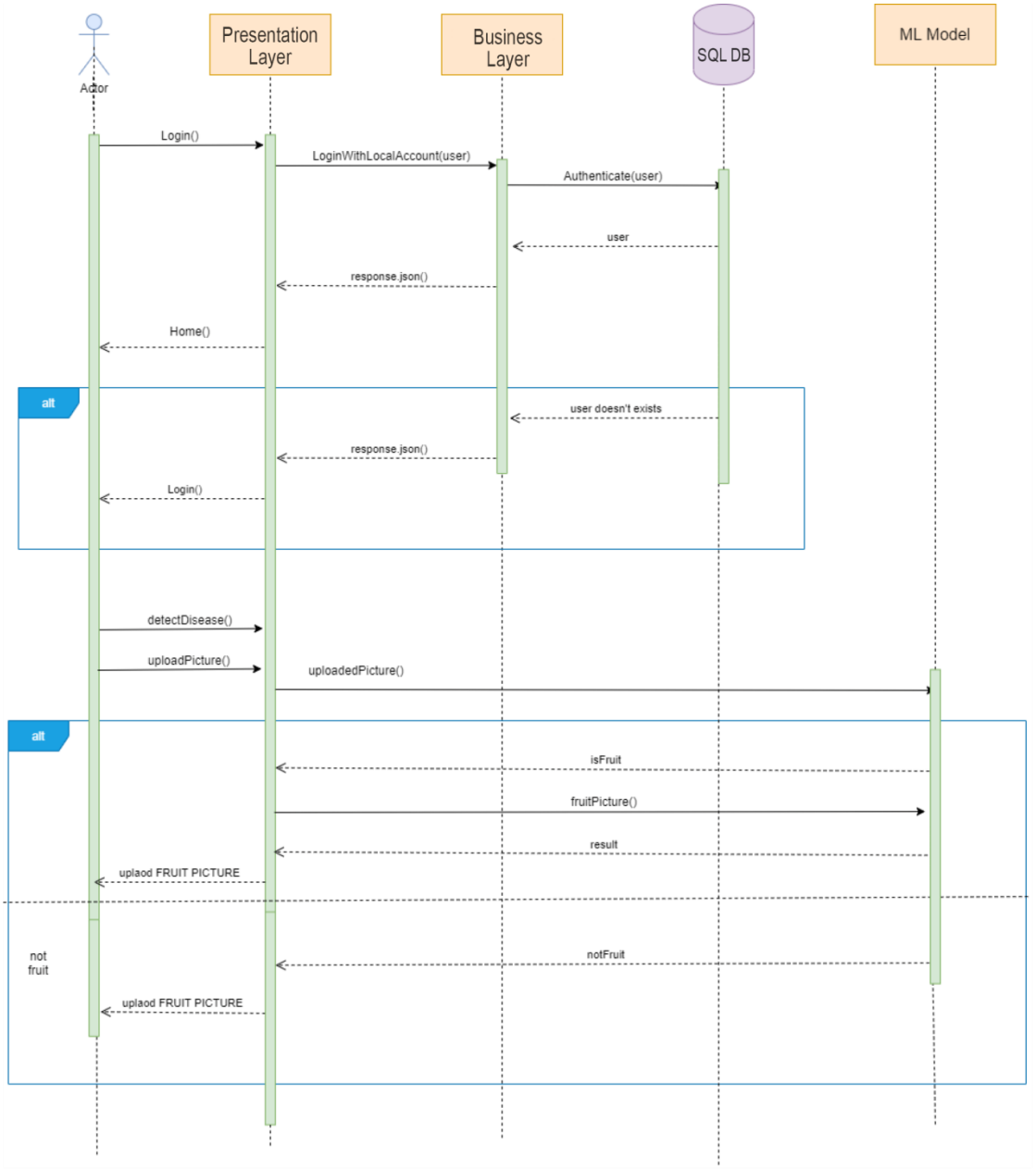

Figure 4.7: Fruit Grading

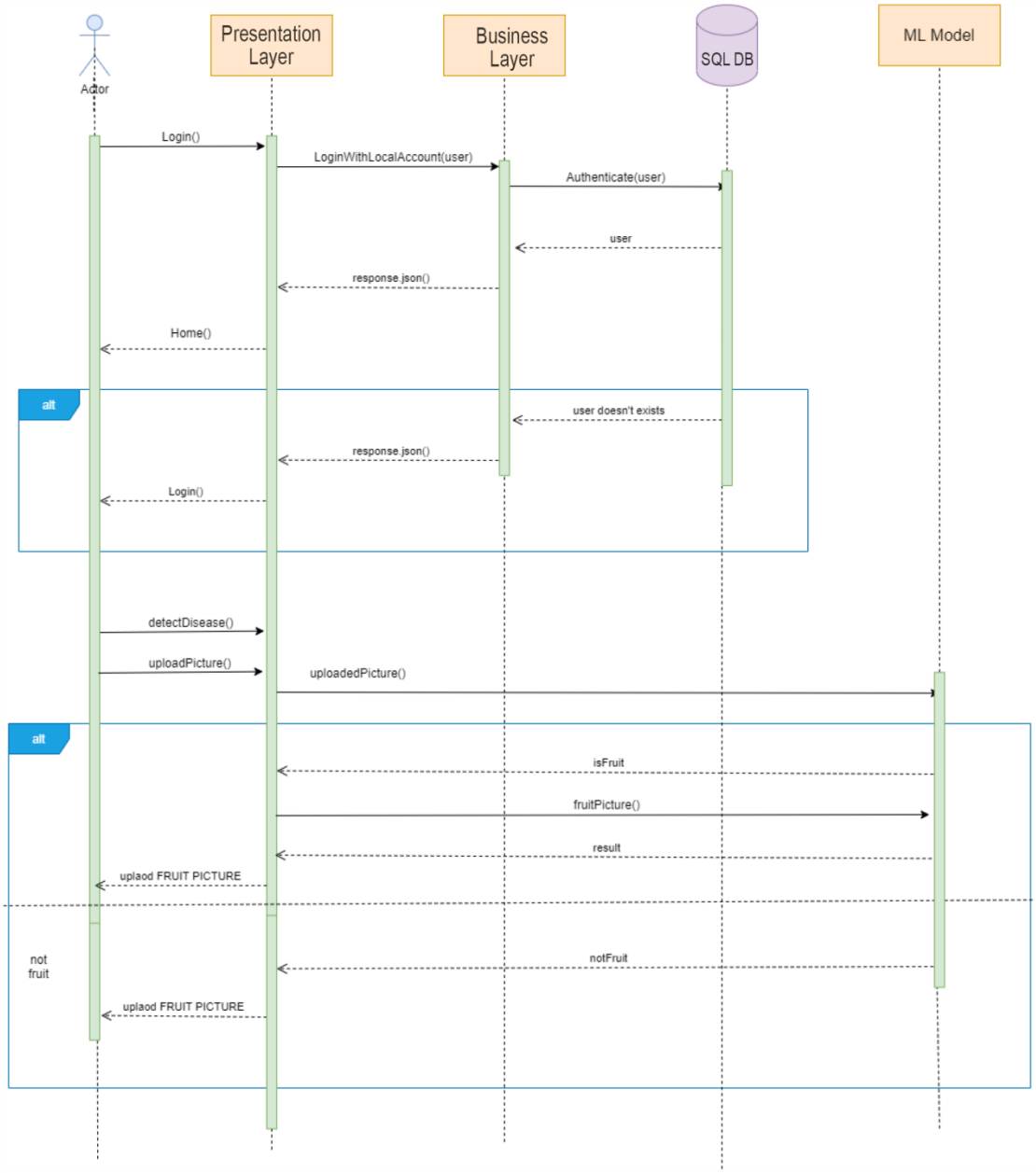

Figure 4.8: Disease Detection

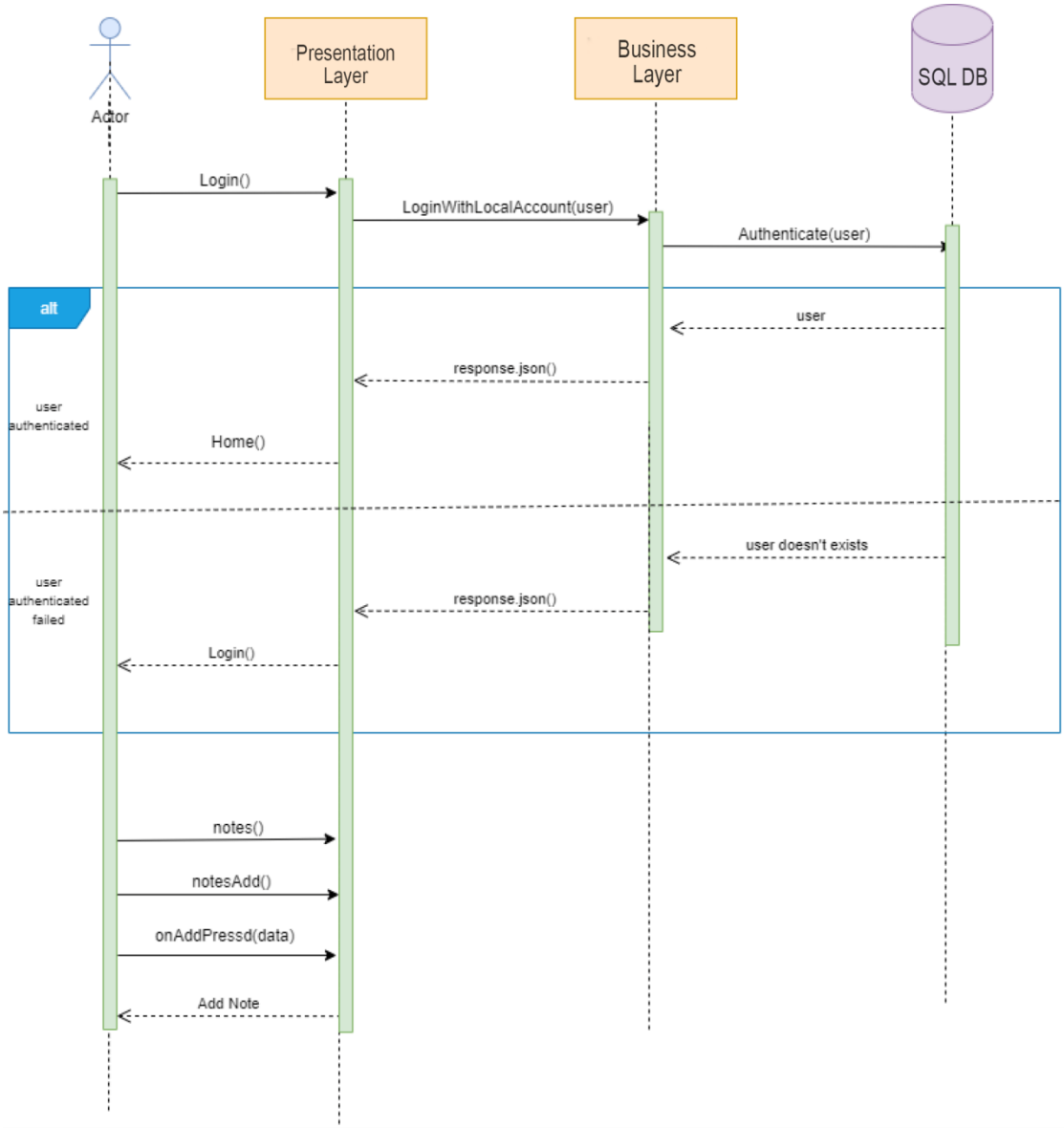

Figure 4.9: Add Notes

Chapter 5

Implementation

5.1 Algorithm

Let's identify the major objects and demonstrate what their corresponding member functions are responsible for in a pseudo code.

These are the 3 major components of the system: Desktop Application, Business Logic Layer, and the SQL Server Express database. Let's write the pseudo for the member functions of each of those components separately.

5.1.1 Disease Detection

5.1.1.1 EfficientNetB7

Transfer learning is the best approach towards any object detection or classification problem. It gives the best possible results, instead of wasting time on developing your own architecture which take lots of time and is not as efficient as the already tried, tested and widely used ML models. EfficientNetB7 is one of those. Obviously not all of those predefined architectures give the best results for a given dataset, but some work the best on a particular dataset, like EfficientNetB7 did on our fruit grading dataset.

5.1.1.2 VGG16

It is another predefined architecture that is used in this application. VGG16 is used mostly in disease detection module. It is efficient than some of the other ML models if it fits the best on a particular dataset, because its size is way lesser and gives predictions in no time. So, it is way more efficient.

5.1.1.3 InceptionResNetV2

InceptionResNetV2 is a heavier model in terms of its size. It gives predictions after taking some time but gives best possible results on some datasets like the apple diseases dataset in this application. It has the highest depth in all the Keras's predefined CNN architectures. So, although its heavier, it's predictions are way more accurate.

5.1.1.4 Xception

Xception is a light weight architecture. It is suitable when the number of classes is lesser, e.g. 2. It gives results in no time because of its lower depth, lesser layers and lesser size.

5.1.2 Disease Cure

The next major module is disease cure module. This module is working all on the basis of a JSON file. Basically, there are some fruits and for each fruit there are some diseases and some pests. Our JSON file, namely cure.json file contains info about all the fruits considered in this application. The info includes the diseases a particular fruit can develop, for each disease there's a name, its picture, a YouTube link where the user can go and check how a particular disease of a particular fruit is taken care of, which pest causes that disease and what are the biological and chemical cures for that disease.

5.1.3 Authentication

5.1.3.1 End User

1. Take the user to the login screen.
2. Take email and password as inputs from the user.
3. Call the Data Access Layer Api.
4. If the Data Access Layer successfully returns the data and the credentials are correct, move forward.
5. Else if there is an error on the server while performing the query, show error toast.
6. If (email exists and password matches) log the user in.
7. Else if (!email) {show toast ("User does not exist")}.
8. Else if (password != user.password) {show toast ("Login failed, wrong password")}.

5.1.3.2 SQL Server Express

When the server calls the find One function of the SQL Server Express database, the SQL Sever Express tries to locate a user with the provided credentials, if one found, it returns the corresponding user, if not, returns null.

5.1.4 Registration

5.1.4.1 End User

1. Take the user to the signup screen
2. Take email, full name and password as inputs from the user
3. Call the Data Access Layer API
4. The client also sends a code to the server to be sent to the client for email verification

5. The server sends the code in a mail through sendmail() API endpoint
6. The client asks the user to enter the sent pin
7. If (enteredPin == sentPin) showToast('Verification successful, signing up…') else showToast('Verification failed')
8. If the pin verification is successful, the add user API end pint of the Data Access Layer is hit.
9. If the server successfully registers the user, move forward.
10. Else if there is an error on the Data Access Layer while performing the query, show error toast.

5.1.4.2 SQL Server Express

When the Data Access Layer calls the save function of the SQL Server Express database, the SQL Server Express verifies all the database constraints on the corresponding user model and if all the constraints are met, it enters a new document in the user collection of the Fruit Doctor database.

5.1.5 Disease Detection

5.1.5.1 End User

1. The user logs into the system and taps on the button to Select the picture of the fruit.
2. The client sends the picture to the fruit detection model to verify that it's a fruit, not anything else.
3. If (Fruit) {send the same image to disease detection model}
4. Else {ask user to take another image}, go back to step 1
5. The disease detection model returns the results
6. If (anyDisease){showDisease(); showCure(detectedDisease)}
7. Else show toast 'Fruit is completely Healthy'

5.1.5.2 Fruit Detection Model

1. Takes an image as input from the client.
2. Applies algorithms like convolution and pooling and figures out if a fruit exists in the image or not
3. Returns the results to the End user

5.1.5.3 Fruit Grading Model

1. After the detection of the fruit in the provided image, the end user passes on this image to the fruit disease detection model.
2. This model takes this image and applies deep learning algorithms to detect a disease in the fruit
3. Sends back the results to the end user.

5.1.6 Fruit Grading

5.1.6.1 End User

1. The user logs into the system and taps on the camera button to take the picture of the fruit.
2. The client sends the picture to the fruit detection model to verify that it's a fruit, not anything else.
3. If (Fruit) {send the same image to fruit quality grading model}
4. Else {ask user to take another image}, go back to step 1
5. The quality grading model returns the results
6. If (grade == A){show toast 'Fruit is completely healthy'}
7. Else if (grade == B) show toast 'Fruit is 75% consumable, not bad'
8. Else if (grade == C) show toast 'Fruit is 50 50'
9. Else if (grade == D) show toast 'Better not eat this one'

5.1.6.2 Fruit Detection Model

1. Takes an image as input from the client.
2. Applies algorithms like convolution and pooling and figures out if a fruit exists in the image or not.
3. Returns the results to the client

5.1.6.3Fruit Grading Model

1. After the detection of the fruit in the provided image, the end user passes on this image to the fruit quality grading model.
2. This model takes this image and applies deep learning algorithms to grade the provided image of fruit within a scale of A to D
3. Sends back the results to the end user

5.1.7 External APIs

Name of API	Description of API	Purpose of Usage	List down the function/class name in which it is used
http://api.openweathermap.org/	Provides the weather data of a particular location	To let our user know what he should expect from the weather, so that they anticipate and prepare accordingly to protect their fruit crops.	Used in the _WeatherState class in the weather screen of the Desktop app.

5.1.8 User Interface

Details about user interface with descriptions are presented in this section.

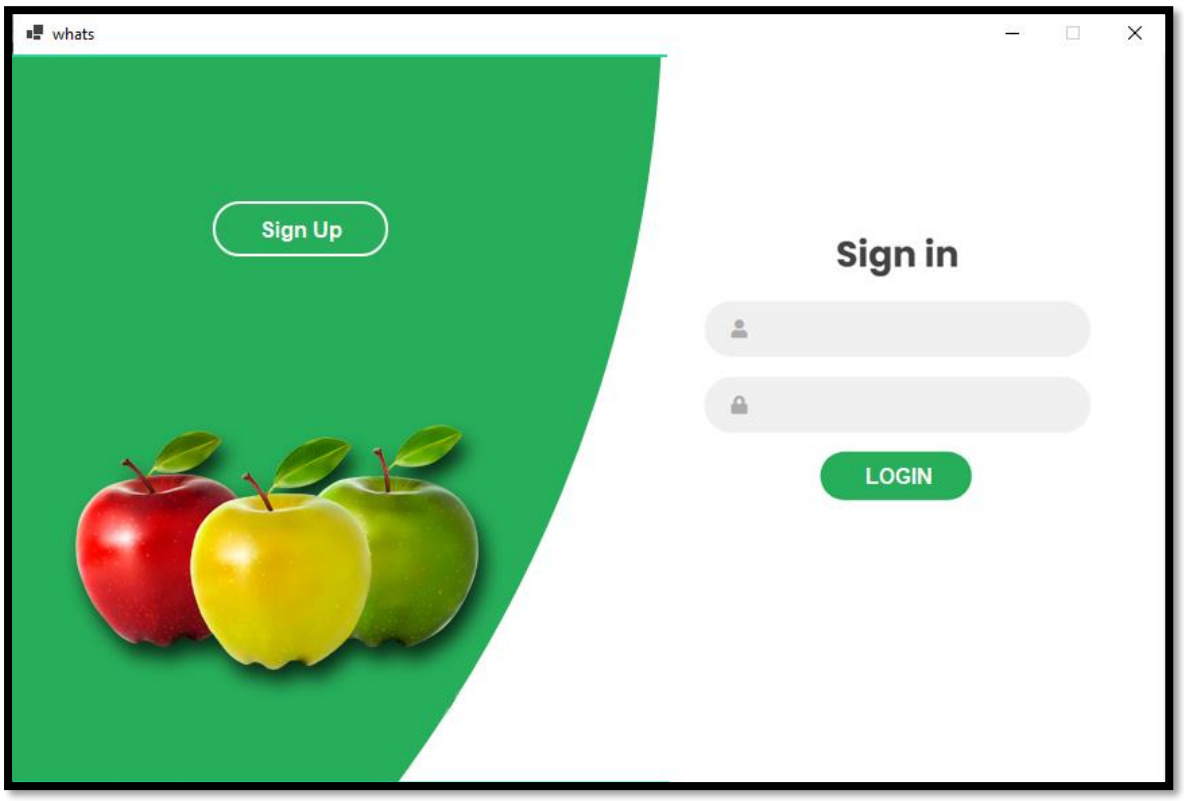

Figure 5.1: Login Page

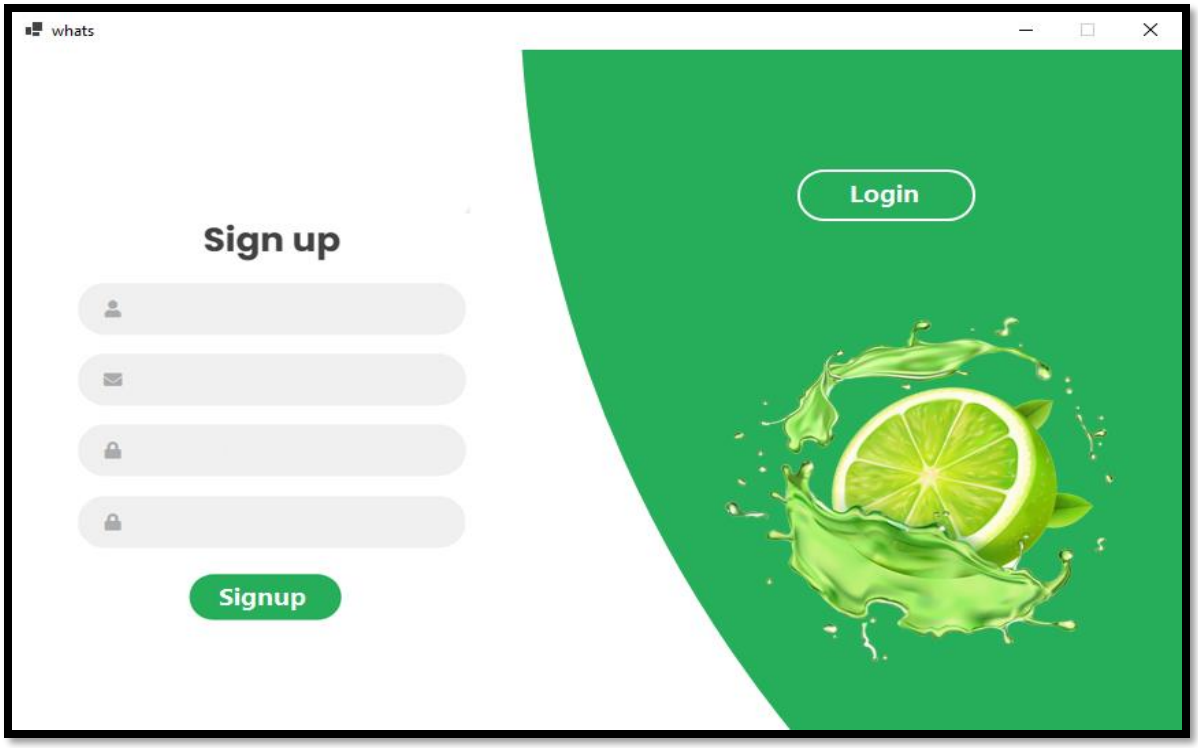

Figure 5.2: Signup Page

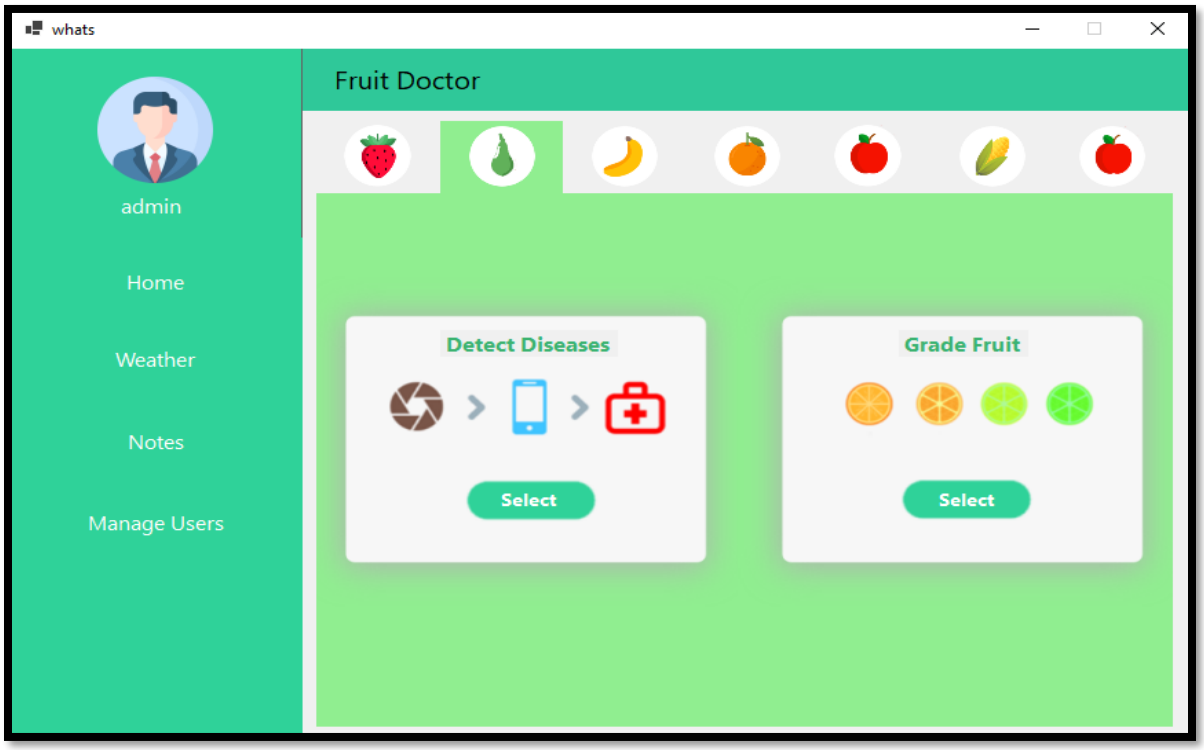

Figure 5.3: Home Page

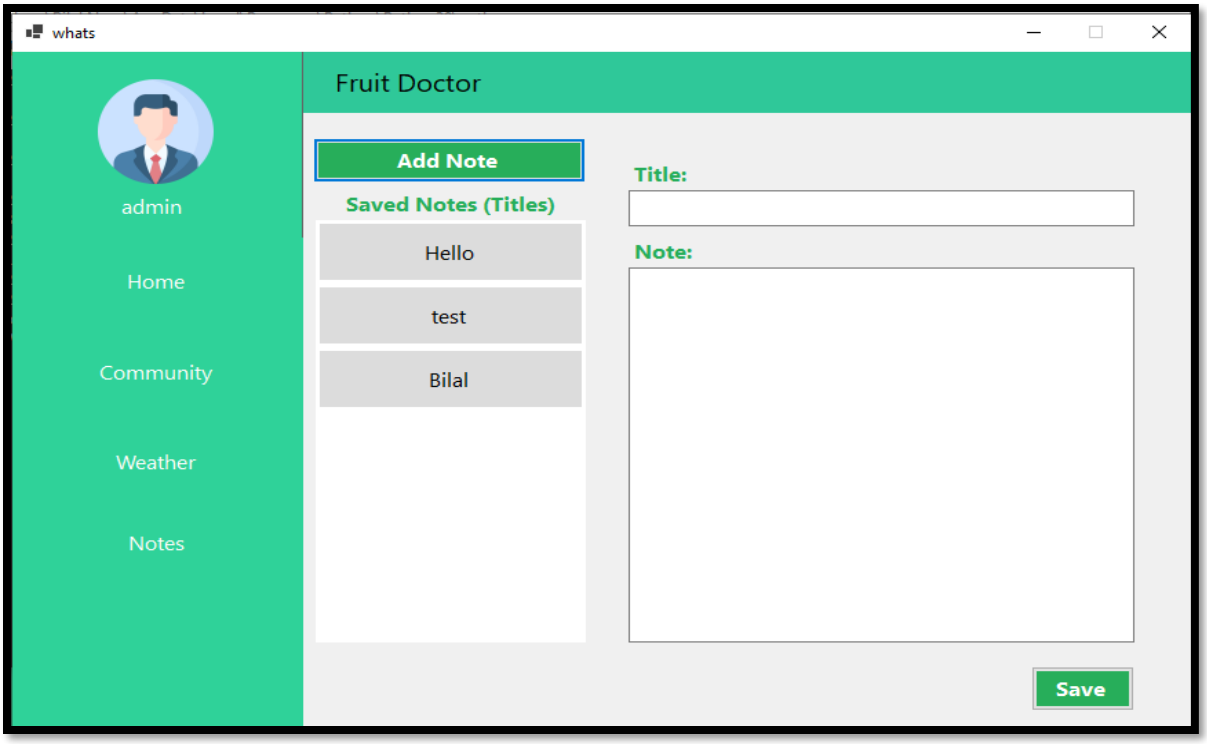

Figure 5.4: Notes

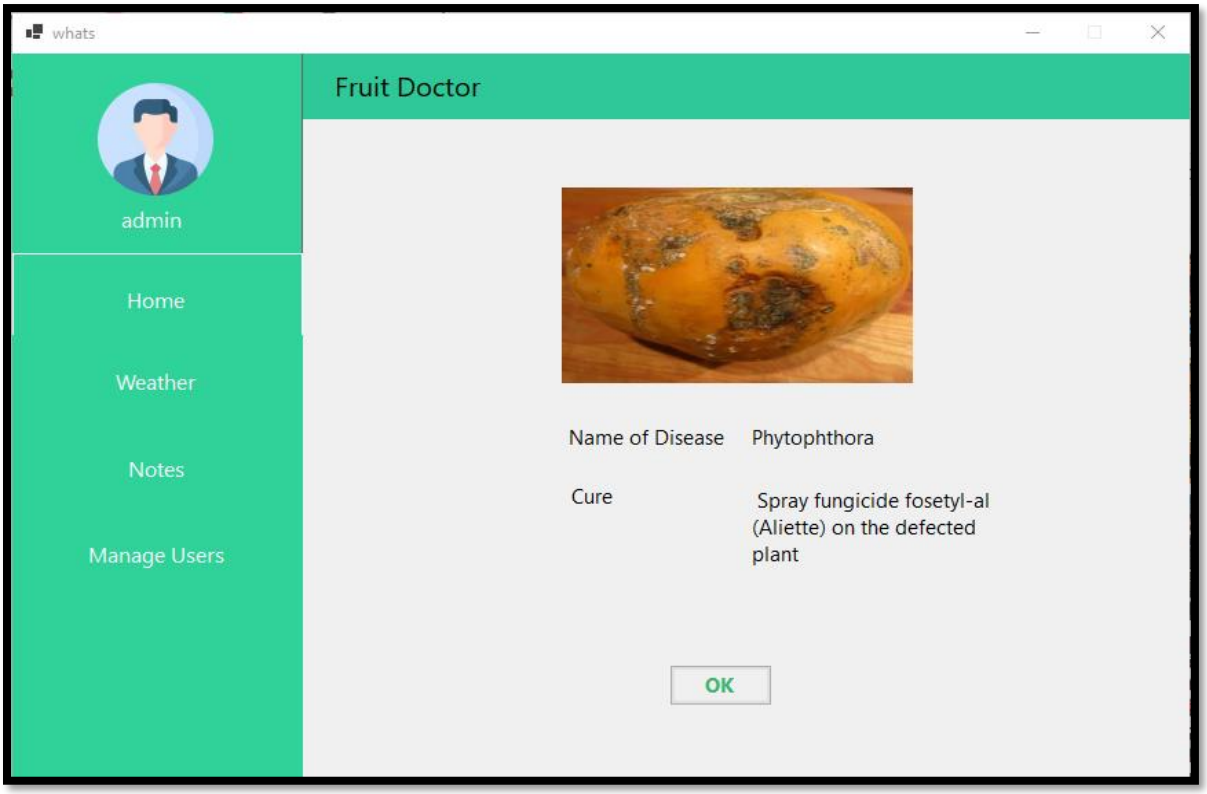

Figure 5.5: Disease Prediction

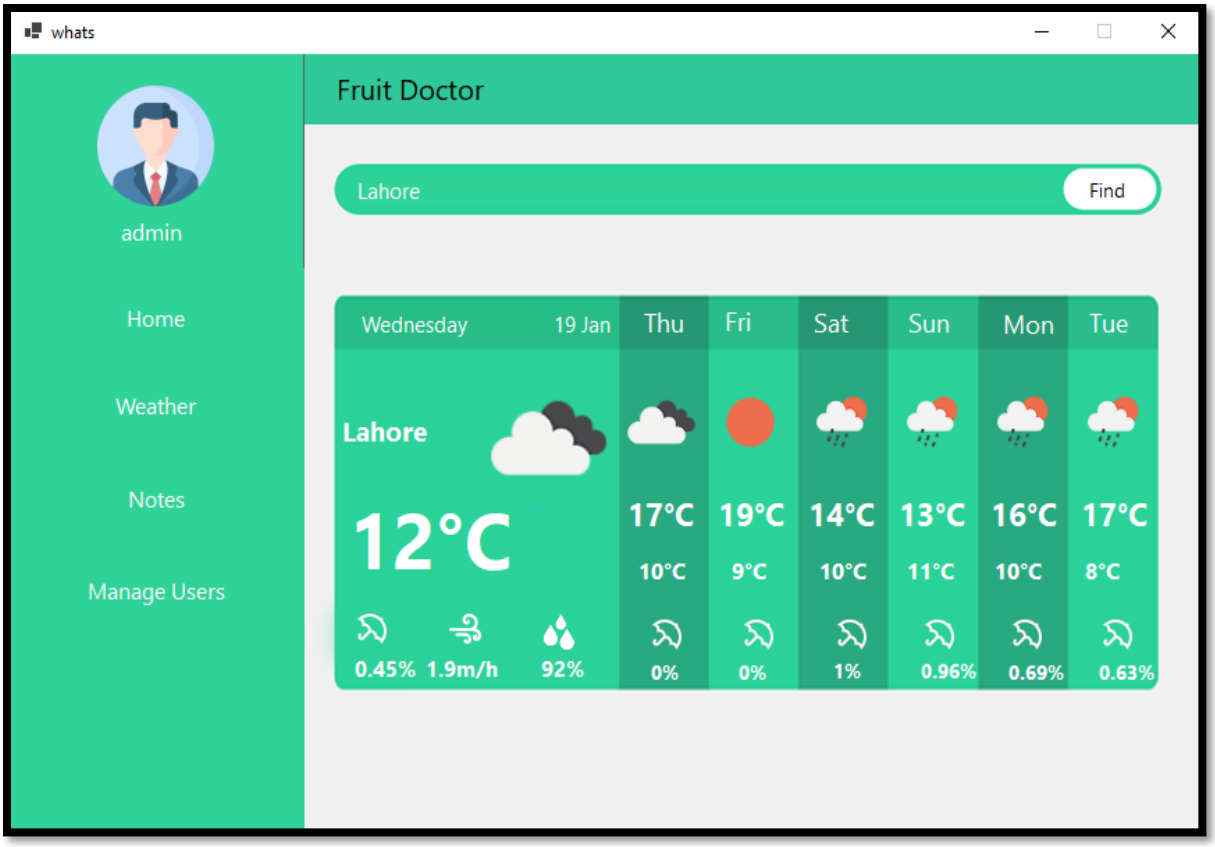

Figure 5.6: Weather

Chapter 6

System Testing

6.1 Manual Testing

6.1.1 Unit Testing

Unit Testing 1: Add New User

Testing Objective: To ensure that a new user can register in the application

Test Case Id: BU_001

Test Case Description: Test the signup functionality

Test Scenario: Verify that on entering correct Name, email, password and confirm password, and the Admin verify the user so it can sign up.

Table 6.1: Successful Signup

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user entered valid password i.e combination of lowercase letters, uppercase letters and special characters.	**Name**: User's full Name **Email**: any valid email that is not already registered (user@example.com) **Password**: Must be 8 digits long with at least one uppercase, one lower case and one special character.	Password is valid.	As Expected	Pass
2.	Verify that password and confirm password are same	Same password as entered previously.	Password Matched	As Expected	Pass

3	Admin verify so the user can signup	Request is send to the admin for verification. If Admin approves.	Signup Successful	As Expected	Pass

Table 6.2: Unsuccessful Signup

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user does not enter a valid email.	**Name**: User's full Name **Email**: anyinvalid email (userexample.com) **Password**: Must be 8 digits long with at least one uppercase, one lower case and one special character.	Informs the user that the email is invalid and they must provide a valid one	As Expected	Pass

Table 6.3: Unsuccessful Signup

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user does not entered an email that is already registered	**Name**: User's full Name **Email**: any valid email (user@example.com) **Password**: Must be 8 digits long with at least one uppercase, one lower case and one special character.	Informs the user that the Email is already registered.	As Expected	Pass

Table 6.4: Unsuccessful Signup

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user does not entered a valid password.	**Name**: User's full Name **Email**: any valid email (user@example.com) **Password**: 8 digits long but no special character abcd1234	Informs the user that the password is not valid.	As Expected	Pass

Table 6.5: Unsuccessful Signup

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user does not enter same password.	**Name:** User's full Name **Email**: any valid email that is not already registered (user@example.com) **Password**: Must be 8 digits long with at least one uppercase, one lower case and one special character. Confirm Password does not match with the password	Informs the user that the password does not match.	As Expected	Pass

Unit Testing 2: Login to the system

Testing Objective: To ensure that a user can login to the system

Test Case Id: BU_002

Test Case Description: Test the login functionality

Test Scenario: Verify that on entering a registered email and correct password, the user can login.

Table 6.6: Login Successful

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify user login after click on the 'Login' button on login form with correct input data	Username: correct Password: correct	Login Successful, the user is taken to the home page of the application	As Expected	Pass

Table 6.7: Login Unsuccessful

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify user does not login after click on the 'Login' button on login form with incorrect input data	Username: correct Password: incorrect	Login Unsuccessful, the user is taken back to the login page again	As Expected	Pass

Table 6.8 Login Unsuccessful

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify user does not login after click on the 'Login' button on login form with incorrect input data	Username: incorrect Password: correct	Login Unsuccessful, the user is taken back to the login page again	As Expected	Pass

Unit Testing 3: Forgot Password

Testing Objective: To ensure reset the password if they forget it, from the login screen

Test Case Id: BU_003

Test Case Description: Test the forgot password functionality

Test Scenario: Verify that on entering a registered email, the user gets an OTP code, and the user is taken to the update password page and on providing the correct OTP code the user enter valid password and correct confirm password fields, the user's password is successfully updated.

Table 6.9: Successful Forgot Password

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user gets an OTP code on the entered email, if the email is registered	Email: any valid email that is registered (user@example.com)	Successfully sends OTP code to the provided email	As Expected	Pass
2.	Verify that the user entered valid password i.e combination of lowercase letters, uppercase letters and special characters.	**Name**: User's full Name **Email**: any valid email that is not already registered (user@example.com) **Password**: Must be 8 digits long with at least one uppercase, one lower case and one special character.	Password is valid.	As Expected	Pass
3	Verify that password and confirm password are same	Same password as entered previously.	Signup Successful	As Expected	Pass

Table 6.10: Unsuccessful Forgot Password

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user does not entered a valid email.	**Name**: User's full Name **Email**: anyinvalid email (userexample.com) **Password**: Must be 8 digits long with at least one uppercase, one lower case and one special character.	Informs the user that the email is invalid and they must provide a valid one	As Expected	Pass

Table 6.11: Unsuccessful Forgot Password

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user does not entered an email that is already registered	**Name**: User's full Name **Email**: any valid email (user@example.com) **Password**: Must be 8 digits long with at least one uppercase, one lower case and one special character.	Informs the user that the Email is already registered.	As Expected	Pass

Table 6.12: Unsuccessful Forgot Password

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user gets an OTP code on the entered email, if the email is registered	Email: any valid email that is registered (user@example.com)	Successfully sends an OTP code to the provided email	As Expected	Pass
2.	Verify that the user does not entered a valid password.	**Name**: User's full Name **Email**: any valid email (user@example.com) **Password**: 8 digits long but no special character abcd1234	Informs the user that the password is not valid.	As Expected	Pass

Table 6.13: Unsuccessful Forgot Password

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user gets an OTP code on the entered email, if the email is registered	Email: any valid email that is registered (user@example.com)	Successfully sends an OTP code to the provided email	As Expected	Pass
2.	Verify that the user entered a valid password.	**Name**: User's full Name **Email**: any valid email (user@example.com) **Password**: 8 digits long but no special character abcd1234	Password is valid.	As Expected	Pass
3.	Verify that the user does not enter same password.	**Name:** User's full Name **Email**: any valid email that is not already registered (user@example.com) **Password**: Must be 8 digits long with at least one uppercase, one lower case and one special character. Confirm Password does not match with the password	Informs the user that the password does not match.	As Expected	Pass

Unit Testing 4: Get Cure

Testing Objective: To ensure that all the info about a fruit's particular disease is shown to the user once a disease is detected

Test Case Id: BU_004

Test Case Description: Test the get cure functionality

Test Scenario: Verify that on clicking the Get Cure button, the user is taken to the complete info and cure of the disease.

Table 6.14: Successfully shown the cure of a detected disease

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user is shown all the relevant information about a detected disease including the pest info which causes a particular disease all the possible cures that they can apply	The user detects a disease and clicks on Get Cure button	The user is shown all the relevant information about a detected disease including the pest info which causes a particular disease	As Expected	Pass

Unit Testing 5: Predict Disease/ Grade Fruit

Testing Objective: To ensure that a user can predict disease/grade a fruit, if a picture is provided and the relevant button is clicked

Test Case Id: BU_005

Test Case Description: Test the predict disease/grade fruit functionality

Test Scenario: Verify that on providing an image, and clicking the relevant button, the user is shown prediction results on a pie chart, showing the percentage probabilities of all the possible diseases.

Table 6.15: Successfully shown the replies to a query in the forum

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user can detect diseases and can get accurate predictions	The user taps on the take a picture button on either the detect a disease card or grade fruit card and goes on to providing a picture, provides a picture and clicks on the relevant button under the picture, that is either detect disease or grade fruit.	The user gets the prediction results in the form of a text, i.e. the maximum predicted probability and a pie chart which shows all the predicted probabilities.	As Expected	Pass

Table 6.16: Successfully shown replies to a query in the forum

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user can not detect diseases/grade fruit and can not get predictions	The user does not select a picture and taps on the predict disease/grade fruit button.	The user is shown a text in the prediction bar that you have to select an image to make a prediction from the application.	As Expected	Pass

6.1.2 Functional Testing

Functional Testing 1: Add New User

Testing Objective: To ensure that a new user can register in the application

Test Case Id: BU_001

Test Case Description: Test the signup functionality

Test Scenario: Verify that on entering correct Name, email, password and confirm password, the user can sign up.

Table 6.17: Successful Signup

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user entered valid password i.e combination of lowercase letters, uppercase letters and special characters.	**Name**: User's full Name **Email**: any valid email that is not already registered (user@example.com) **Password**: Must be 8 digits long with at least one uppercase, one lower case and one special character.	Password is valid.	As Expected	Pass
2.	Verify that password and confirm password are same	Same password as entered previously.	Signup Successful	As Expected	Pass
3	Admin verify so the user can signup	Request is send to the admin for verification. If Admin approves.	Signup Successful	As Expected	Pass

Table 6.18: Unsuccessful Signup

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user does not entered a valid email.	**Name**: User's full Name **Email**: anyinvalid email (userexample.com) **Password**: Must be 8 digits long with at least one uppercase, one lower case and one special character.	Informs the user that the email is invalid and they must provide a valid one	As Expected	Pass

Table 6.19: Unsuccessful Signup

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user does not entered an email that is already registered	**Name**: User's full Name **Email**: any valid email (user@example.com) **Password**: Must be 8 digits long with at least one uppercase, one lower case and one special character.	Informs theuser that the Email is already registered.	As Expected	Pass

Table 6.20: Unsuccessful

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user does not entered a valid password.	Name: User's full Name Email: any valid email (user@example.com) Password: 8 digits long but no special character abcd1234 Confirm Password has not match with the current password	Informs the user that the password is not valid.	As Expected	Pass

Table 6.21: Unsuccessful Signup

No.	**Test Case/Test Script**	**Test Data**	**Expected Result**	**Actual Result**	**Pass/Fail/Not Executed/ Suspended**
1.	Verify that the user does not enter same password.	**Name:** User's full Name **Email**: any valid email that is not already registered (user@example.com) **Password**: Must be 8 digits long with at least one uppercase, one lower case and one special character. Confirm Password does not match with the password	Informs the user that the password does not match.	As Expected	Pass

Functional Testing 2: Login to the system

Testing Objective: To ensure that a user can login to the system

Test Case Id: BU_002

Test Case Description: Test the login functionality

Test Scenario: Verify that on entering a registered email and correct password, the user can login

Table 6.22: Login Successful

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify user login after click on the 'Login' button on login form with correct input data	Username: correct Password: correct	Login Successful, the user is taken to the home page of the application	As Expected	Pass

Table 6.23: Login Unsuccessful

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify user does not login after click on the 'Login' button on login form with incorrect input data	Username: correct Password: incorrect	Login Unsuccessful, the user is taken back to the login page again	As Expected	Pass

Table 6.24: Login Unsuccessful

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify user does not login after click on the 'Login' button on login form with incorrect input data	Username: incorrect Password: correct	Login Unsuccessful, the user is taken back to the login page again	As Expected	Pass

Functional Testing 3: Forgot Password

Testing Objective: To ensure reset the password if they forget it, from the login screen

Test Case Id: BU_003

Test Case Description: Test the forgot password functionality

Test Scenario: Verify that on entering a registered email, the user gets the a link, and from the following link, the user is taken to the update password page and on providing the valid password and correct confirm password fields, the user's password is successfully updated.

Table 6.25: Successful Forgot Password

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user gets an OTP code on the entered email, if the email is registered	Email: any valid email that is registered (user@example.com)	Successfully sends OTP code to the provided email	As Expected	Pass
2.	Verify that the user entered valid password i.e combination of lowercase letters, uppercase letters and special characters.	**Name**: User's full Name **Email**: any valid email that is not already registered (user@example.com) **Password**: Must be 8 digits long with at least one uppercase, one lower case and one special character.	Password is valid.	As Expected	Pass
3	Verify that password and confirm password are same	Same password as entered previously.	Signup Successful	As Expected	Pass

Table 6.26: Unsuccessful For Password

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user does not entered a valid email.	**Name**: User's full Name **Email**: anyinvalid email (userexample.com) **Password**: Must be 8 digits long with at least one uppercase, one lower case and one special character.	Informs the user that the email is invalid and they must provide a valid one	As Expected	Pass

Table 6.27: Unsuccessful Forgot Password

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user does not entered an email that is already registered	**Name**: User's full Name **Email**: any valid email (user@example.com) **Password**: Must be 8 digits long with at least one uppercase, one lower case and one special character.	Informs the user that the Email is already registered.	As Expected	Pass

Table 6.28: Unsuccessful Forgot Password

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user gets an OTP code on the entered email, if the email is registered	Email: any valid email that is registered (user@example.com)	Successfully sends an OTP code to the provided email	As Expected	Pass
2.	Verify that the user does not entered a valid password.	**Name**: User's full Name **Email**: any valid email (user@example.com) **Password**: 8 digits long but no special character abcd1234	Informs the user that the password is not valid.	As Expected	Pass

Table 6.29: Unsuccessful Forgot Password

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user gets an OTP code on the entered email, if the email is registered	Email: any valid email that is registered (user@example.com)	Successfully sends an OTP code to the provided email	As Expected	Pass
2.	Verify that the user entered a valid password.	**Name**: User's full Name **Email**: any valid email (user@example.com) **Password**: 8 digits long but no special character abcd1234	Password is valid.	As Expected	Pass
3.	Verify that the user does not enter same password.	**Name:** User's full Name **Email**: any valid email that is not already registered (user@example.com) **Password**: Must be 8 digits long with at least one uppercase, one lower case and one special character. Confirm Password does not match with the password	Informs the user that the password does not match.	As Expected	Pass

Functional Testing 4: Get Cure

Testing Objective: To ensure that all the info about a fruit's particular disease is shown to the user once a disease is detected

Test Case Id: BU_004

Test Case Description: Test the get cure functionality

Test Scenario: Verify that on clicking the Get Cure button, the user is taken to the complete info and cure of the disease.

Table 6.30: Successfully shown the cure of a detected disease

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Verify that the user is shown all the relevant information about a detected disease including the pest info which causes a particular disease , and all the possible cures that they ca n apply	The user detects a disease and clicks on Get Cure button	The user is shown all the relevant information about a detected disease including the pest info which causes a particular disease and all the possible cure that can apply	As Expected	Pass

6.1.3 Integration Testing

Table 6.31: Test Cases for Integration Testing of Authentication

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	Login without have account register	email: ali@gmail.com Password: Ali@123	Request sent to Data Access Layer. No such record present on database. No account registered against this email, so login failed.	As Expected	Pass
2.	Sign up	email: ali@gmail.compass word: Ali@123 confirm password: Ali@123	Successfully Registered account.	As Expected	Pass
3.	Login with wrong email or password	Email: Ali@gmail.com Password: os@222123	The request sent to the server the password or email is incorrect. Response sent to the interface about login failed	As Expected	Pass
4.	Forget Password	Email: Ali@gmial.com	User enters an email, and an OTP code is send to the user on the given email from the server. On right entry the update password screen is shown to user	As Expected	Pass

Table 6.32: Test Cases for Integration Testing of Disease and Grading Prediction

No.	Test Case/Test Script	Test Data	Expected Result	Actual Result	Pass/Fail/Not Executed/ Suspended
1.	User selects from the system for disease prediction	Picture	The picture is sent to model integrated and the prediction data that model sent shows to the user in the form of pie graph	As Expected	Pass
2.	User select an image from the system for grading that fruit.	Picture	The picture is sent to model integrated and the prediction data that model sent shows to the user.	As Expected	Pass
3.	Disease Cure	The disease that is predicted by the system	The request is sent to the Data Access layer about the cure of that disease predicted and the layer sent data of that particular disease and Similar diseases.	As Expected	Pass

6.1.4 System Testing

All sorts of testing including unit testing, functional testing and integration testing were performed to ensure that the system performs as specified in the requirements document and as well as specified in the software design specification. As the overall system is also passed through intensive testing consists of load testing and usability testing. All these testing fragments are passed by our Desktop application.

Chapter 7

Conclusion

7.1 Conclusion

Fruit Doctor, a complete system that provides modern technology to the farmers for analyzing their fruit crop and helps farmers in increasing the yield. Fruit doctor diagnoses diseases in different fruits and provides the best way to deal with a particular disease problem by providing all the possible cures to the end user and by giving all the relevant info about the detected disease like pests that cause it, chemical and biological cures. As farmers have to do a lot of farming related tasks, to organize those tasks, the system provides an interface that allows the user to write down their work.

7.2 Future Work

The following things can be considered worth adding to the project in the future:

- More fruits can be added to enhance the application's scope by doing some research on disease datasets.
- For video or stream processing, the application, model server and the model's architecture need to be revamped little more and some more resources are required to make it possible. But if it can be achieved somehow, it can make disease detection way easier for the end user.
- A handsome application can also be thought of to widen the user circle of Fruit Doctor.

References

1. Dubey, S.R. Detection and Classification of Apple Fruit Diseases using Complete Local Binary Patterns, 2012
2. Gavhale, M.K.R. Unhealthy Region of Citrus Leaf Detection Using Image Processing Techniques,2014,pp.2–7.
3. Jhuria, M. & Kum, A., g for Smart Farming : Detection of Disease and Fruit Grading,2013, pp.521– 526.
4. Khirade, S.D. & Patil, A.B., Plant Disease Detection Using Image Processing. 2015 International Conference on Computing Communication Control and Automation,2015, pp.768–771.
5. Samajpati, B.J. & Degadwala, S.D., 2016. Hybrid Approach for Apple Fruit Diseases Detection and Classification Using Random Forest Classifier, (2013), pp.1015–1019.
6. Singh, V., Detection of unhealthy region of plant leaves using Image Processing and Genetic Algorithm, 2015, pp.1028–1032.
7. Dhakate, M., Ingole, A.B.: Diagnosis of pomegranate plant diseases using neural network. In Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), Dec 2015, pp. 1–4, 16–19.
8. Dhakate, M., Ingole, A.B.: Diagnosis of pomegranate plant diseases using neural network. In:
9. Revathi, P., Hemalatha, M.: Classification of Cotton Leaf Spot Diseases Using Image Processing Edge Detection Techniques. In: International Conference on Emerging Trends in Science, Engineering and Technology (INCOSET), Dec 2012, pp. 169–173, 13–14.
10. Lu, J., Wu, P., Xue, J., Qiu, M., Peng, F.: Detecting defects on citrus surface based on circularity threshold segmentation. In: 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD),(2015) , pp. 1543–1547.
11. Dewliya, S., Singh, P.: Detection and classification for apple fruit diseases using support vector machine and chain code. Int. Res.J. Eng. Technol. (IRJET) 02, 04 Aug 2015.
12. Awate, A., Deshmankar, D., Amrutkar, G., Bagul, U., Sonavane, S.: Fruit disease detection usingcolor, texture analysis and ANN. International Conference on Green Computing and Internet of Things (ICGCIoT),Oct 2015, pp. 970–975, 8–10.
13. Bin Mohamad Azmi, M.T., Isa, N.M.: Orchid disease detection using image processing andfuzzy logic. In: International Conference on Electrical, Electronics and System Engineering (ICEESE), Dec 2013, pp. 37–42, 4–5.
14. Bhange, M., Hingoliwala, H.A.: Smart farming: pomegranate disease detection using image processing. In: Second International Symposium on Computer Vision and the Internet (VisionNet'15),Aug. 2015, pp. 280-288, 22.