

Software Requirement Specification Document for Fungicide Utilizer

Kirrollos Amir, Mahmoud Heidar, Walid Wagdy, Yara Negm
Supervised by: Dr. Taraggy, Eng. Randa

December 28, 2020

Table 1: Document version history

Version	Date	Reason for Change
1.0	28-Oct-2020	SRS First version's specifications are defined.

GitHub: <https://github.com/KirrollosAmir/Graduation-Project.git>

Abstract

Agriculture is one of the most important sectors in Egypt , although environmental factors influence the life cycle of crops. Many diseases depend on climatic changes in the weather. Integrating technology with agriculture is a major challenge but an important approach to the evolution of the agriculture sector. We aim to build a system that collects data from the weather station , this data helps measure the effects of weather on crops and fungicides. This data is used in three models: The first is the IOT model which recommends the application of the fungicide to crops with an accurate percentage dose taking into account any corrective action as well as when to use it based on the weather API. The second model is the fuzzy inference model that predicts disease severity in crops based on the weather station. The third model that will be built at the end of the project is the ANN model ,that predicts some diseases in some crops based on the data set collected from the weather station and used during the project period.

1 Introduction

1.1 Purpose of this document

The purpose of this Program Requirement Specification document is to define both functional and non-functional requirements for our "Fungicide Use" application [1] , [2], as well as the inter-

faces and restrictions that the system must meet and operate under. The main requirement for this program is to improve the field of agriculture [3], [4]. This document will be provided with a complete description of each case of the project, along with a full explanation of all the requirements of the stage and the development process. This document is for both stakeholders and the system development team[5].

1.2 Scope of this document

This SRS document targets farm owners and agricultural experts. Our application will help them in improving crop health through calculation of fungicide dose and application time [6], [7] It will also predict some diseases based on the collected data from the weather station [3], [8]. our system explains the purpose of the program in a simple and understandable way. This document assist future developers and designers to understand the system and its functions[9]

1.3 System Overview

As shown in figure 1, the proposed system is presented in 4 stages: The first stage is data entry which consists of the DHT11 sensor responsible for obtaining temperature and humidity data, the soil moisture sensor, the wind sensor, and the rain sensor associated with the Arduino to collect information about the yield and climate. The next stage is the controller consisting of an Arduino, 1sheeld and a smartphone. The Arduino receives data from the sensors and sends it to the server through 1sheeld and smartphone. The third stage is the server which consists of the database that stores the data received from the console along with the APIs that process the data through the fuzzy model that predicts some diseases and the IOT model which recommends the use of fungicides along with the recommended concentration and time to add it. The fourth stage is the mobile application that enables users to view the status of their crops and their recommendations, as well as enter data on crops and fungicides such as fungicides used and dates of addition, as well as their notes about crops such as symptoms that appear on crops.

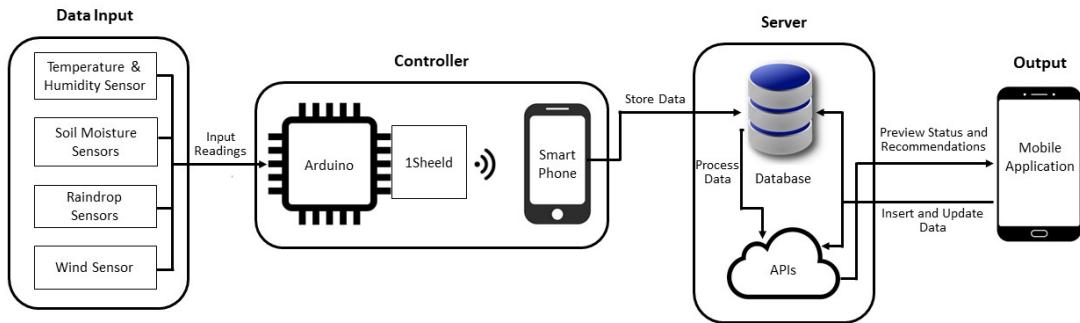


Figure 1: System Overview

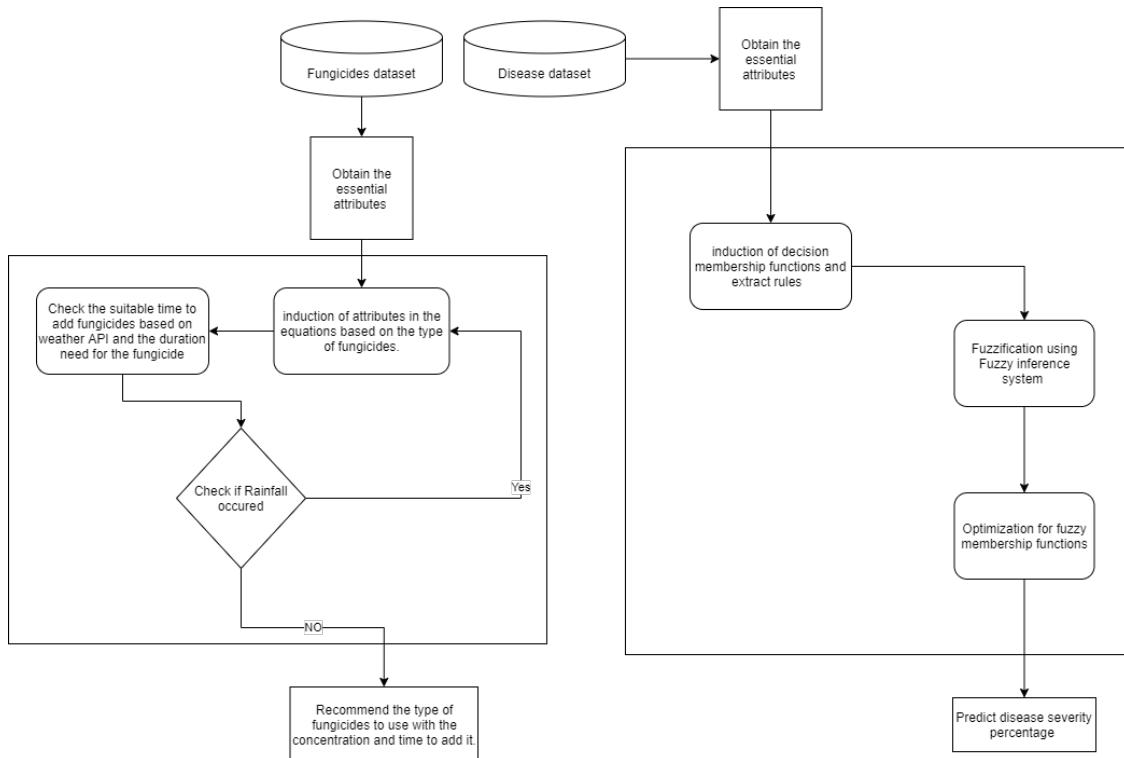


Figure 2: Software Flowchart

1.4 System Scope

The proposed system is designed to:

- Build a weather station that collects data for analysis for many uses, with an Internet of Things system which tells the farmer when to apply fungicides and the dose percentage.
- Build a fuzzy model to help detect rust disease in the wheat crop early.
- Build an artificial intelligence model to predict some diseases that affect crops by analyzing data from the weather station.

1.4.1 Expected outcome

The expected outcome of the project is an application fully operated to improve and ensure crop quality.

- The application should monitor the climate changes [10] to alert the farmer whether to apply the fungicide or to avoid its use, and the system will recommend the appropriate fungicide for the selected crop and its proportion to use a corrective action in case of incorrect reading.
- The application should predict the wheat rust disease [11] [12] [13] that threatens wheat crops in Egypt using data collected from the weather station.
- The application will collect some observations from agricultural users and experts about crop health and build assumptions based on weather [3] and those observations, to help researchers track some crop diseases and build an AI model.

1.5 Business Context

The goal of our system is to guarantee the quality of crops. Some farmers waste [14], [15] while using them. Most agricultural experts currently have limited time and have to make a quick decision without a good diagnosis of the problem. The process of manually detecting crop diseases is time consuming. This automated system will save time for experts and provide more accurate results. Moreover, reduce system operating costs by eliminating the need for continuous human monitoring.

2 Similar Systems

2.1 Academic

Richmond et.al.[16] presented the end result of some fungicides [17] on plant microarchitecture. They found that benomyl and carbendazim affect the cleavage of the botrytis nuclei of cineria in 5 minutes and detect unusual behavior in cell division in 3% of the cells in the onion root[18]. In their description they found that the compounds can be transported in plants [19] either in the apoplast (the non-living parts of the cell) or in a symplast (the living parts of cells).

Roelfs et.al. [20] This book is talking about Rust disease [21] [13]and the types of it and it has models that we take to help in our system , the Rust [3] have many types such as leaf rust and stripe rust . Rust can cause a 100 percent of loss in the wheat crop and it is because the change of the temperature . here is some of the Environmental conditions the help Rust disease in wheat spread [11] .

Stripe or yellow rust is principally a disease of wheat grown in cooler climates (2-15°C), which are generally associated with higher elevations, northern latitudes or cooler years. It takes its name from the characteristic stripe of uredinia that produce yellow colored urediniospores. Because of the disease's early attack, stunted and weakened plants often occur. Losses can be severe (50%) due to shriveled grain and damaged tillers. In extreme situations, stripe rust can cause 100% losses.

Tables 1 and 2 summarize primary hosts, alternate hosts, symptoms, and generally accepted environmental conditions needed by the three rust diseases.

Table 1. The rust diseases of wheat, their primary and alternate hosts, and symptoms.

Disease	Pathogen	Primary hosts	Alternate hosts	Symptoms
Leaf rust	<i>Puccinia recondita</i> f.sp. <i>tritici</i>	Bread & durum wheats and triticale	<i>Thalictrum</i> , <i>Anchusa</i> <i>Isopyrum</i> , and <i>Clematis</i>	Isolated uredinia on upper leaf surface and rarely on leaf sheaths
Stem rust	<i>Puccinia graminis</i> f.sp. <i>tritici</i>	Bread & durum wheats, barley, and triticale	<i>Berberis vulgaris</i>	Isolated uredinia on upper and lower leaf surfaces, stem, and spikes
Stripe rust	<i>Puccinia striiformis</i> f.sp. <i>tritici</i>	Bread & durum wheats, triticale, and a few barley cultivars	unknown	Systemic uredinia on leaves and spikes and rarely on leaf sheaths

Figure 3: Symbols of Rust disease

EPIDEMIOLOGY

There are several areas worldwide in which each of the rusts can cause severe losses (326). In other areas the environment is marginally suited for the diseases. In such areas, the disease is severe only in years when:

- Conditions are unusually favorable.
- Susceptible cultivars are grown.
- Cultural practices are altered.
- The above factors occur in combination.

Table 3 provides a general summary of the current and historical importance of the rust diseases worldwide.

Urediniospores of the wheat rusts initiate germination within 1 to 3 hours of contact with free moisture over a range of temperatures depending on the rust. Urediniospores are produced in large numbers and can be blown considerable distances by the wind (149, 392). However, most urediniospores are deposited close to their source (309) under the influence of

Table 2. Environmental conditions required for the wheat rusts.

Stage	Temperature (°C)				Light	Free water
	Minimum	Optimum	Maximum			
Leaf rust						
Germination	2	20	30		Low	Essential
Germling	5	15-20	30		Low	Essential
Appressorium		15-20			None	Essential
Penetration	10	20	30		No effect	Essential
Growth	2	25	35		High	None
Sporulation	10	25	35		High	None
Stem rust						
Germination	2	15-24	30		Low	Essential
Germling		20			Low	Essential
Appressorium		16-27			None	Essential
Penetration	15	29	35		High	Essential
Growth	5	30	40		High	None
Sporulation	15	30	40		High	None
Stripe rust						
Germination	0	9-13	23		Low	Essential
Germling		10-15			Low	Essential
Appressorium			(not formed)			
Penetration	2	8-13	23		Low	Essential
Growth	3	12-15	20		High	None
Sporulation	5	12-15	20		High	None

Figure 4: Environmental conditions required for the wheat Rust

David H. Gent et.al. [22] This paper is talking about Disease forecasts [23]from regional or remotely sensed meteorological data free growers from in- field weather data monitoring and may improve disease forecast implementation . they monitor weather by an automated re- gional network of 40 Campbell Scientific, Inc. (CSI) CR-10 weather stations (Camp- bell Scientific, Inc., Logan, UT) in the Colorado Agricultural Meteorological Network (COAGMET), The results was Comparison of temperature observations [10], [24]. In-field and mPOWER3/EMERGE mean hourly temperature deviations from COAGMET observations varied from 1.11oC to -0.93oC, de- pending on location and source of meteorological data

Emperical models	Fundamental Models
Dutch Rules (1926)	BLIGHTCAST (1975)
Beaumont's Rules (1938)	Mac Hardy (1979)
Cook's System (1947, 49)	Fry and Apple (1983)
Bourke's System (1953)	Hijmans, <i>et al.</i> (2000)
Hyre's System (1954)	Grunwald, <i>et al.</i> (2000)
Hyre and Bonds's System (1955)	Hansen, <i>et al.</i> (2000)
Smith's Sytem (195)	
Wallin's system (1962)	Runno and Koppel (2002)

Figure 5: Fundamental Models

Table 1: Adjustable Matrix developed by Krause *et al.* (1975) using Hyre's *blight favourable days* and Wallin's *severity values*.

Total rain favourable days during last 7 days	Severity values during last 7 days						
	<3	3	4	5	6	<6	
	<5	-1	-1	0	1	1	2
	>4	-1	0	1	2	2	2
Message number							

Message numbers	Recommendation
-1	No spray
0	Late blight warning
1	7-day schedule
2	5-day schedule

Figure 6: Environmental conditions required for the wheat Rust

Dammer et.al. [25] introduced a real time technique in the practice of spraying some fungicide [26] modifiers in some areas. Plant parameters can be detected to indirectly characterize heterogeneous plant growth such as biomass or plant surface area by CROP-Meter. The sensor technologies in the market for automatic detection of diseases for which the plant area index will

be developed, and in the end, it was possible to take into account the different dispersion dynamics of different fungal diseases in a field, and more than one disease[13] could occur simultaneously or sequentially in the grain field and with different infections.

Halah et.al. [27] proposed a system explaining Chocolate spots disease which affect Faba beans and how the equations they find to help the crop to get rid of this disease by scheduling the Fungicides [2] , [16]based on the study was carried out during two successive growing in winter peaks seasons (2010/2011 and 2011/2012) on faba bean .

Disease severity was calculated using the equation developed by Townsend and Heuberger (1943).

$$P = [\Sigma(n \times v)/4N] \times 100$$

where :

P = Disease severity

n = Number of leaves within infection category.

v = Numerical value of each category.

N = Total number of leaves.

Fungicide efficacy was calculated using Abbott equation (Frölich, 1979)

$$Fe = \frac{c - t}{c} * 100$$

Where :

Fe = Fungicide efficacy

c = Disease severity in Control

t = Disease severity in treatment.

Fungicide efficacy reduction (as a result of rainfall) was calculated as follows:

$$Fer = (Fec - Fct) / Fec \times 100$$

Where;

Fre = Fungicide efficacy reduction %

Fec = Fungicide efficacy of the tested fungicide without rainfall (control)

Fet = Fungicide efficacy of the tested fungicide that received certain amount of rainfall after certain time (treatment).

Figure 7: Equations of disease severity

Table 1: Disease severity, fungicides efficacy and fungicide efficacy reduction as a result of different amounts of artificial rainfall after different periods of fungicides application in the agricultural year 2010/2011 under field conditions.

Treatments *		Disease and fungicide parameters and time of application of the artificial rainfall after the second fungicides application in hours														
Fungicides	Time in hr.	Disease severity					fungicide efficacy					fungicide efficacy reduction %				
		6	12	24	48	Mean	6	12	24	48	Mean	6	12	24	48	Mean
Amistar Top 32.5% SC	0.0	10.0	10.0	10.0	10.0	10.0	83.0	83.0	83.0	83.0	83.0	0.0	0.0	0.0	0.0	0.0
	2.5	16.3	15.0	12.7	12.0	14.0	74.2	76.3	81.0	81.0	78.1	13.1	11.0	3.1	2.0	7.3
	5.0	20.3	17.3	14.7	11.7	16.0	67.9	72.6	77.8	81.6	75.0	19.4	14.7	6.3	1.4	10.5
	7.5	24.3	20.3	18.3	12.7	18.9	61.6	67.9	69.9	80	69.9	25.7	19.4	12.1	3	15.1
	10	30.7	25.7	22.7	16.7	24.0	51.6	59.5	66.8	73.7	62.9	35.7	27.8	18.9	9.3	22.9
	Mean	20.3	17.7	15.7	12.6	16.6	67.7	71.9	75.7	79.9	73.8	18.8	14.6	8.1	3.1	11.2
Topsin M70 WP	0.0	14.3	14.3	14.3	14.3	14.3	77.4	77.4	77.4	77.4	77.4	0	0	0	0	0.0
	2.5	24.0	23.3	18.3	15.0	20.2	62.1	63.1	71.3	76.3	68.2	15.7	14.6	6.7	1.5	9.6
	5.0	29.0	31.7	25.0	16.9	25.7	54.2	50.0	55.7	73.3	58.3	23.6	27.8	17.3	4.5	18.3
	7.5	34.7	36.4	31.6	20.0	30.7	45.2	42.4	51.4	68.4	51.9	32.5	35.3	27.8	9.4	26.3
	10	44.3	43.1	38.6	26.0	38.0	30.0	32.0	40.2	58.9	40.3	44.5	45.8	38.9	18.9	37.0
	Mean	29.3	29.8	25.6	18.4	25.8	53.8	53.0	59.2	70.9	59.2	23.3	24.7	18.1	6.9	18.2
Rovral 50% WP	0.0	17.0	17.0	17.0	17.0	17.0	73.2	73.2	73.2	73.2	73.2	0.0	0.0	0.0	0.0	0.0
	2.5	32.0	29.2	21.7	21.0	26.0	49.4	53.9	65.8	66.8	59.0	23.7	15.9	4.1	3.0	11.7
	5.0	37.3	36.0	28.3	24.0	31.4	41.0	43.1	55.2	62.1	50.4	32.1	26.7	14.6	7.8	20.3
	7.5	44.6	41.7	36.7	27.3	37.6	29.6	34.2	42.1	56.8	40.7	43.5	35.7	27.8	13.0	30.0
	10	47.7	45.8	43.3	33.3	42.5	24.7	27.7	31.5	47.3	32.8	48.4	42.1	38.3	22.5	37.8
	Mean	35.72	33.9	29.4	24.5	30.9	43.6	46.4	53.6	61.2	51.2	29.5	24.1	16.9	9.3	20.0
Tridex 80% WP	0.0	27.7	27.7	27.7	27.7	27.7	56.3	56.3	56.3	56.3	56.3	0.0	0.0	0.0	0.0	0.0
	2.5	43.9	40.0	38.3	34.0	39.1	30.6	36.8	39.4	46.3	40.3	23.0	17.2	14.5	7.7	15.6
	5.0	48.3	44.0	43.3	36.7	43.1	23.6	30.5	31.5	42.1	33.6	30.0	23.5	22.4	11.9	22.0
	7.5	55.4	52.1	48.1	39.3	48.7	12.5	17.7	24.0	37.9	26.0	41.2	36.3	30.0	16.1	30.9
	10	58.9	56.7	56.2	52.1	56.0	6.8	10.4	11.2	17.6	12.6	46.8	43.5	42.8	36.3	42.4
	Mean	46.8	44.1	42.7	38.0	42.9	26.0	30.3	32.5	40.0	33.8	28.2	24.1	21.9	14.4	22.2

*= Mean of disease severity in general control (plots treated only with water without any fungicide) was 63.33.

LSD 0.05 for disease severity: time	rain amount	time X rain intensities	Correlation Coefficients	Time	Rain amount
Amistar Top	1.4	2.9	5.8	for fungicide efficacy reduction: -65.6%	69.4%
Topsin- M70	2.8	2.9	5.9	-61.7%	72.7%
Rovral	2.7	3.6	7.2	-62.4%	73.7%
Tridex	0.7	2.7	5.5	-49.0%	80.2%

Figure 8: Fungicides efficacy and fungicides efficacy reduction

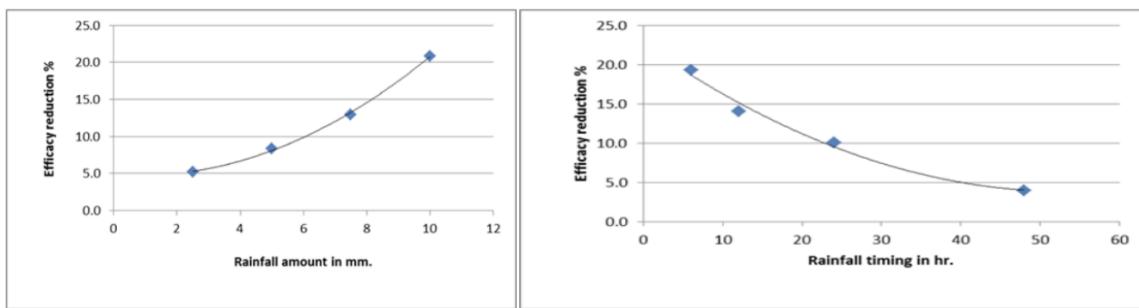


Figure 1: Amistar Top efficacy reduction affected with different . Figure 2: Amistar Top efficacy reduction affected with

Figure 9: Amistar of efficacy reduction

2.2 Business Applications

Sirrus: A mobile application exists from any computer or any mobile device, whether the user is working on the road, farm or office. User can easily extract insights from field data, create recommendations and point of view, and make sound agricultural decisions, and it comes with many features from these:

- Explore crops to compress pests, record and collect monitoring images with ease.
- Create and share recommendation from the field.
- Create PDF reports and share them via email or text message.
- A sample of the soil using a network, areas, or previous points for soil sampling.
- View fertilizer and nutrient recommendations.
- Send wireless recommendations using raven slingshot.
- Export data as a format file.
- Receive updated hourly rainfall estimates for each field.
- Leading or drawing new boundaries of the field.
- Securely store and share data via the agX platform.
- Advanced fertilizer recommendation editor: Edit variable rate recommendations by product total or cost.
- Product Labels / SDS: View and store product information, labels and updated SDS for offline use.
- Variety Tech Papers: View, share, and store various attributes / information for offline use.
- Premium Reports: Beautiful PDF reports that cover scouting and recommendation communication needs.
- Record fields to receive in-season photos to explore oriented crops and create recommendations.

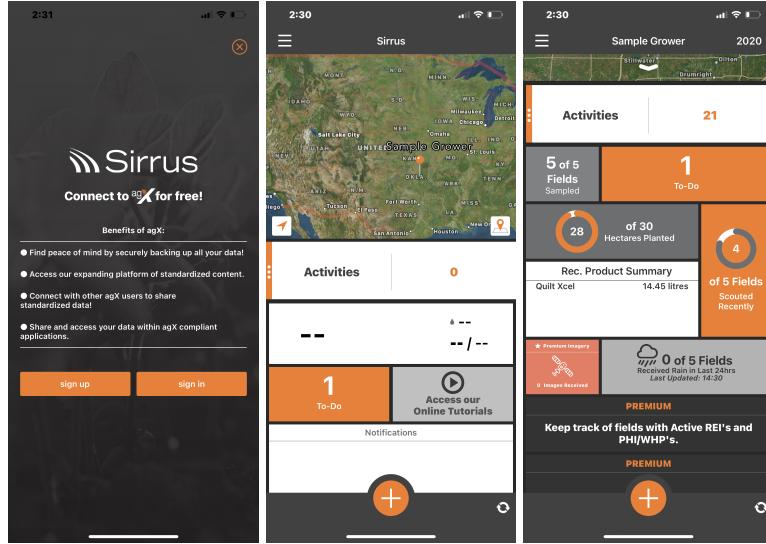


Figure 10: Application interface

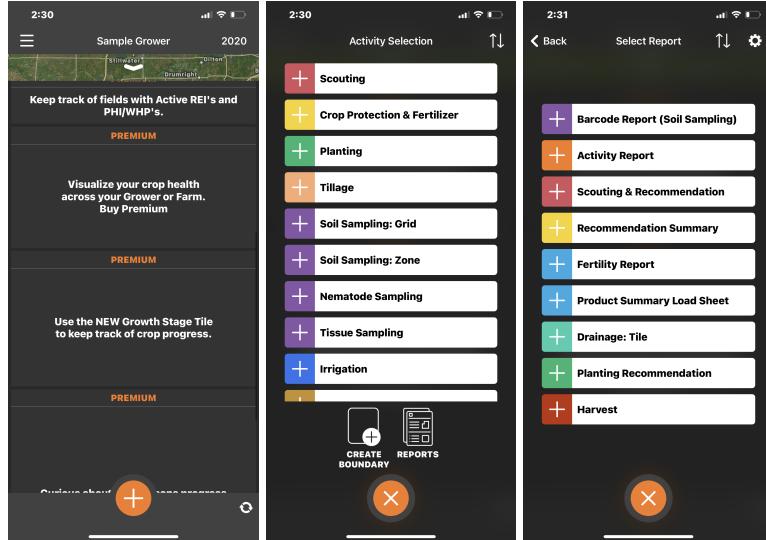


Figure 11: Application interface

›: The Android and iOS compatible web app gives farmers access to their business identification feature, this app also provides market description, agriculture news, weather information, local rain report, commodity prices, everything to help farmers.

The use of the smart platform uses data science, such as public and private data sets, and deep learning algorithms to design guarantee-backed crop management plans, minimizing the risks farmers take to help them more efficiently and to have a healthy plant that grows, nourishes and harvests crops.

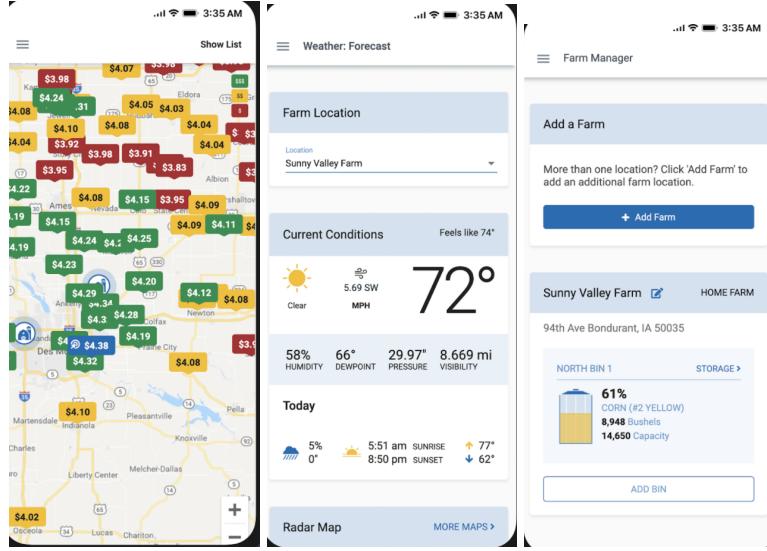


Figure 12: Application interface

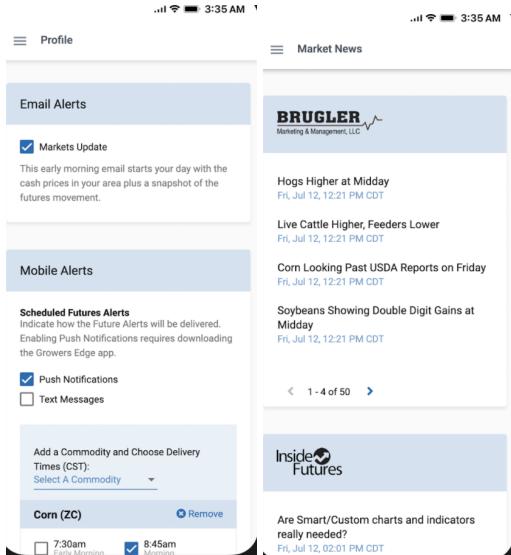


Figure 13: Application interface

3 System Description

3.1 User Problem Statement

Farmers and crop disease experts have some restrictions that they have to track before it occurs, such as rust disease in wheat crops [13] [21], [11] and its application time, fungicides and their concentration accuracy. Diagnoses vary according to the weather climate [8], [23], which is difficult to track, and experts and farmers may face some problems in this regard, whether they have to use a fungicide or avoid using it with a corrective approach in the event of rain with an accurate corrective ratio. They also need to check crops every day for some symptoms of disease and this is very difficult because some diseases spread quickly, so they must track it before it is too late. So

we proposed a disease predictor system and automatically recommend related treatments [28].

3.2 User Objectives

With the use of the Internet of Things, the field of agriculture can be improved for artificial intelligence and fuzzy models [29][30] and tracking crop health is made easier through disease prediction. The user will only receive a notification in his cell phone, to see if there are any changes in the climate that could affect the crops with any disease or fungicides applied, the system must also recommend which fungicide to be applied, the percentage and time to apply. The user will choose the crop he is growing from the crop list to start and will fill in some data that will be useful to the system, and he must also record all his observations about the grown crop that will help in some AI models.

3.3 User Characteristics

There are 3 types of users that interact with the system:

- Farms owners, agricultural experts and Admin:
 - Must have basic knowledge in using Android or Ios mobile devices, how to use the device and apps.

3.4 System Context

There is no system that has no boundaries, our system will have some constraints :

- Must have sensors to collect data from the farms .
- Must have a server to store the data that sensors are collecting .
- Must have an Arduino to take the input reading from sensors .
- Must have connection to the internet as it should deal with the real data transfer to the server.

4 Functional Requirements

4.1 System Functions

- The admin shall be able to add , edit , delete and view user.
- The admin shall be able to add , edit , delete and view crop.
- The admin shall be able to add , edit , delete and view fungicide.
- The admin shall be able to add , edit , delete and view disease.
- The admin shall be able to add , edit , delete and view notification.
- The admin shall be able to add , edit , delete and view report.
- The admin shall be able to add , edit , delete and view statistics.
- The admin shall be able to add , edit , delete and view news.
- The farmer shall be able to choose crop.
- The farmer shall be able to choose fungicides.
- The farmer shall be able to add and delete land.
- The farmer shall be able to view reports , statistics and news.
- The expert shall be able to view reports , statistics and news.
- The system shall be able to do encrypt and decrypt password.
- The system shall be able to predict diseases.
- The system shall be able to calculate fungicide dose and when to apply it.
- The system shall be able to predict diseases of crops.

4.2 Detailed Functional Specification

Table 2: Add User

Name	Accept User
Code	FR01
Priority	Extreme
Critical	it is very critical step to our project to create accept account request
Description	it is an important and essential step complete creation of a user account to can use our project.
Input	Not needed
Output	User or Expert accepted
Pre-condition	User must register.
Post-condition	the account will be accepted and stored in the database to can exist the application
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 3: Accept Expert

Name	Accept Expert
Code	FR02
Priority	Extreme
Critical	it is very critical step to our project to create accept account request
Description	it is an important and essential step complete creation of a user account to can use our project.
Input	—
Output	Expert accepted
Pre-condition	Expert must register.
Post-condition	the account will be accepted and stored in the database to can exist the application
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 4: Create Crop

Name	Create Crop
Code	FR03
Priority	High
Critical	it is very critical step to our project to get the higher performance.
Description	it is an important step to get the higher efficiency and best results.
Input	Crop name and details
Output	Crop added
Pre-condition	-
Post-condition	the crop added on the database
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 5: view Crop

Name	View Crop
Code	FR04
Priority	medium
Critical	it is important to users , experts and admin to view the crops to check anything.
Description	it is an important step to make check on crops details.
Input	-
Output	-
Pre-condition	there are already crops saved on the database.
Post-condition	-
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 6: Update Crop

Name	Update Crop
Code	FR05
Priority	high
Critical	it is important to users , experts and admin to view the crops to check anything.
Description	it is an important for user if need to make an edit on the crop details.
Input	crop details.
Output	crop updated.
Pre-condition	the crop has no updates.
Post-condition	the updated details replaced.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 7: Delete Crop

Name	Delete Crop
Code	FR06
Priority	high
Critical	it is important to admin if need to delete crops.
Description	it is an important for user if need to make an edit on the crop details.
Input	crop details.
Output	crop updated.
Pre-condition	the crop has no updates.
Post-condition	the updated details replaced.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 8: Disease Add

Name	Add Disease
Code	FR07
Priority	high
Critical	it is important to admin if need to add disease.
Description	it is an important for admin if need to add a diseases.
Input	diseases details.
Output	disease added.
Pre-condition	-
Post-condition	the disease is added in the database.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 9: Disease view

Name	Add Disease
Code	FR08
Priority	high
Critical	it is important to admin if need to view the diseases.
Description	it is an important for user if need to look on diseases details.
Input	-
Output	-
Pre-condition	must be at least 1 disease stored in the database.
Post-condition	-
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 10: Disease update

Name	update Disease
Code	FR09
Priority	high
Critical	it is important to admin if need to make an update.
Description	it is an important for user if need to make an edit on the disease details.
Input	disease details.
Output	disease information updated.
Pre-condition	the disease has no updates.
Post-condition	the updated details replaced.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 11: Disease delete

Name	Delete Disease
Code	FR10
Priority	high
Critical	it is important to admin if need to add crops.
Description	it is an important for admin if he want to remove disease.
Input	-
Output	crop updated.
Pre-condition	-
Post-condition	Disease deleted.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 12: Fungicide Add

Name	Add Fungicide
Code	FR11
Priority	high
Critical	it is important to admin if need to add fungicide.
Description	it is an important for admin if need to add a fungicide.
Input	fungicide details.
Output	fungicide added.
Pre-condition	-
Post-condition	the fungicide is added in the database.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 13: fungicide view

Name	Add Disease
Code	FR12
Priority	high
Critical	it is important to admin if need to view the fungicide.
Description	it is an important for user if need to look on fungicide details.
Input	-
Output	-
Pre-condition	must be at least 1 fungicide stored in the database.
Post-condition	-
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 14: fungicide update

Name	update fungicide
Code	FR13
Priority	high
Critical	it is important to admin if need to make an update.
Description	it is an important for user if need to make an edit on the fungicide details.
Input	fungicide details.
Output	fungicide information updated.
Pre-condition	the fungicide has no updates.
Post-condition	the updated details replaced.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 15: Disease delete

Name	Delete Disease
Code	FR14
Priority	high
Critical	it is important to admin if need to add crops.
Description	it is an important for admin if he want to remove disease.
Input	-
Output	crop updated.
Pre-condition	-
Post-condition	Disease deleted.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 16: notification Add

Name	Add notification
Code	FR15
Priority	high
Critical	it is important to admin if need to add notification.
Description	it is an important for admin if need to add a notification.
Input	notification details.
Output	notification added.
Pre-condition	-
Post-condition	the notification is added in the database.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 17: notification view

Name	view notification
Code	FR16
Priority	high
Critical	it is important to admin if need to view the notification.
Description	it is an important for user if need to look on notification details.
Input	-
Output	-
Pre-condition	must be at least 1 notification stored in the database.
Post-condition	-
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 18: notification delete

Name	Delete notification
Code	FR17
Priority	high
Critical	it is important to admin if want to make a notification remove.
Description	it is an important for admin if he want to remove notification.
Input	-
Output	notification deleted.
Pre-condition	-
Post-condition	the notification will removed from the database.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 19: Report Add

Name	Add Report
Code	FR18
Priority	high
Critical	it is important to admin if need to add Report.
Description	it is an important for admin if need to add a Report.
Input	Report details.
Output	Report added.
Pre-condition	none
Post-condition	the Report is added in the database.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 20: Report view

Name	View Report
Code	FR19
Priority	high
Critical	it is important to admin if need to view the Report.
Description	it is an important for user if need to look on Report details.
Input	none
Output	none
Pre-condition	must be at least 1 Report stored in the database.
Post-condition	none
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 21: Report update

Name	update Report
Code	FR20
Priority	high
Critical	it is important to admin if need to make an update.
Description	it is an important for user if need to make an edit on the Report details.
Input	Report details.
Output	Report information updated.
Pre-condition	the Report has no updates.
Post-condition	the updated details replaced.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 22: Report delete

Name	Delete Disease
Code	FR21
Priority	high
Critical	it is important to admin if need to delete Reports.
Description	it is an important for admin if he want to remove Report.
Input	-
Output	Report deleted.
Pre-condition	the report still stored in the database.
Post-condition	Report deleted from the database.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 23: Report Add

Name	Add Report
Code	FR22
Priority	high
Critical	it is important to admin if need to add Report.
Description	it is an important for admin if need to add a Report.
Input	Report details.
Output	Report added.
Pre-condition	-
Post-condition	the Report is added in the database.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 24: Report view

Name	View Report
Code	FR23
Priority	high
Critical	it is important to admin if need to view the Report.
Description	it is an important for user if need to look on Report details.
Input	-
Output	-
Pre-condition	must be at least 1 Report stored in the database.
Post-condition	-
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 25: Report update

Name	update Report
Code	FR24
Priority	high
Critical	it is important to admin if need to make an update.
Description	it is an important for user if need to make an edit on the Report details.
Input	Report details.
Output	Report information updated.
Pre-condition	the Report has no updates.
Post-condition	the updated details replaced.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 26: Report delete

Name	Delete Disease
Code	FR25
Priority	high
Critical	it is important to admin if need to delete Reports.
Description	it is an important for admin if he want to remove Report.
Input	-
Output	Report deleted.
Pre-condition	the report still stored in the database.
Post-condition	Report deleted from the database.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 27: statistics Add

Name	Add statistics
Code	FR26
Priority	high
Critical	it is important to admin if need to add statistics.
Description	it is an important for admin if need to add a statistics.
Input	statistics details.
Output	statistics added.
Pre-condition	-
Post-condition	the statistics is added in the database.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 28: statistics view

Name	View statistics
Code	FR27
Priority	high
Critical	it is important to admin if need to view the statistics.
Description	it is an important for user if need to look on statistics details.
Input	-
Output	-
Pre-condition	must be at least 1 statistics stored in the database.
Post-condition	-
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 29: statistics update

Name	update statistics
Code	FR28
Priority	high
Critical	it is important to admin if need to make an update.
Description	it is an important for user if need to make an edit on the statistics details.
Input	statistics details.
Output	statistics information updated.
Pre-condition	the statistics has no updates.
Post-condition	the updated details replaced.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 30: Delete statistics

Name	Delete statistics
Code	FR29
Priority	high
Critical	it is important to admin if need to delete statistics.
Description	it is an important for admin if he want to remove statistics.
Input	-
Output	statistics deleted.
Pre-condition	the statistics still stored in the database.
Post-condition	statistics deleted from the database.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 31: News Add

Name	Add News
Code	FR30
Priority	high
Critical	it is important to admin if need to add News.
Description	it is an important for admin if need to add a News.
Input	statistics details.
Output	statistics added.
Pre-condition	-
Post-condition	The News is added in the database.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 32: News view

Name	View news
Code	FR31
Priority	high
Critical	it is important to admin if need to view the news.
Description	it is an important for user if need to look on news details.
Input	-
Output	-
Pre-condition	must be at least 1 new stored in the database.
Post-condition	-
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 33: news update

Name	update news
Code	FR32
Priority	high
Critical	it is important to admin if need to make an update.
Description	it is an important for user if need to make an edit on the statistics details.
Input	news details.
Output	news information updated.
Pre-condition	the news has no updates.
Post-condition	the updated details replaced.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 34: Delete news

Name	Delete news
Code	FR33
Priority	high
Critical	it is important to admin if need to delete news.
Description	it is an important for admin if he want to remove news.
Input	-
Output	news deleted.
Pre-condition	the news still stored in the database.
Post-condition	news deleted from the database.
Dependency	the admin account must be created.
Risk	this process might not able to be satisfied when there is no admin account, and to reduce the probability of this occurrence, the admin account must be created firstly

Table 35: choose crop

Name	crop choose
Code	FR34
Priority	high
Critical	it is important to farmer if need to choose a crop.
Description	it is important to farmer if need to make operations on a specific crop.
Input	crop name
Output	crop details.
Pre-condition	there is no selected crop else to make operations on it.
Post-condition	the operation will start on the selected crop.
Dependency	the farmer account must be created.
Risk	this process might not able to be satisfied when there is no farmer account, and to reduce the probability of this occurrence, the farmer account must be created.

Table 36: choose fungicide

Name	fungicide choose
Code	FR35
Priority	high
Critical	it is important to farmer if need to choose a fungicide.
Description	it is important to farmer if need to use a specific fungicide.
Input	fungicide name
Output	fungicide details.
Pre-condition	there is no selected fungicide else to use it.
Post-condition	the operation will start on the selected fungicide.
Dependency	the farmer account must be created.
Risk	this process might not able to be satisfied when there is no farmer account, and to reduce the probability of this occurrence, the farmer account must be created.

Table 37: Add Farm

Name	Add Farm
Code	FR36
Priority	high
Critical	it is important to Farmer if need to add Farm.
Description	it is an important for Farmer if need to add a Farm to distribute it into lands.
Input	Farm details.
Output	Farm added.
Pre-condition	-
Post-condition	the Farm is added in the database.
Dependency	the Farmer account must be created.
Risk	this process might not able to be satisfied when there is no Farmer account, and to reduce the probability of this occurrence, the Farmer account must be created firstly

Table 38: Farm view

Name	View Farm
Code	FR37
Priority	high
Critical	it is important to Farmer if need to view the Farm.
Description	it is an important for user if need to look on Farm details.
Input	-
Output	-
Pre-condition	must be at least 1 Farm stored in the database.
Post-condition	-
Dependency	the Farmer account must be created.
Risk	this process might not able to be satisfied when there is no Farmer account, and to reduce the probability of this occurrence, the Farmer account must be created firstly

Table 39: Farm update

Name	Update Farm
Code	FR38
Priority	high
Critical	it is important to farmer if need to make an update on the Farm.
Description	it is an important for farmer if he need to make an edit on the Farm details.
Input	Farm details.
Output	Farm information updated.
Pre-condition	the Farm has no updates.
Post-condition	the updated details replaced.
Dependency	Farmer account must be created.
Risk	this process might not able to be satisfied when there is no Farmer account, and to reduce the probability of this occurrence, the Farmer account must be created firstly

Table 40: Delete Farm

Name	Delete Farm
Code	FR39
Priority	high
Critical	it is important to Farmer if need to delete Farm.
Description	it is an important for Farmer if he want to remove Farm.
Input	-
Output	Farm deleted.
Pre-condition	the Farm still stored in the database.
Post-condition	Farm deleted from the database.
Dependency	the Farmer account must be created.
Risk	this process might not able to be satisfied when there is no Farmer account, and to reduce the probability of this occurrence, the Farmer account must be created.

Table 41: Add Land

Name	Add Land
Code	FR40
Priority	high
Critical	it is important to Farmer if need to add Land .
Description	it is an important for Farmer if need to add a Lands on the farm.
Input	Land details.
Output	Land added.
Pre-condition	-
Post-condition	the Land is added in the database.
Dependency	the Farmer account must be created.
Risk	this process might not able to be satisfied when there is no Farmer account, and to reduce the probability of this occurrence, the Farmer account must be created firstly

Table 42: Delete Land

Name	Delete Land
Code	FR41
Priority	high
Critical	it is important to Farmer if need to delete Land.
Description	it is an important for Farmer if he want to remove Land.
Input	-
Output	Land deleted.
Pre-condition	the Land still stored in the database.
Post-condition	Land deleted from the database.
Dependency	the Farmer account must be created.
Risk	this process might not able to be satisfied when there is no Farmer account, and to reduce the probability of this occurrence, the Farmer account must be created.

Table 43: view reports

Name	View reports
Code	FR42
Priority	high
Critical	it is important to Farmer if need to view the reports.
Description	it is an important for user if need to look on report details.
Input	-
Output	-
Pre-condition	must be at least 1 report stored in the database.
Post-condition	-
Dependency	the Farmer account must be created.
Risk	this process might not able to be satisfied when there is no Farmer account, and to reduce the probability of this occurrence, the Farmer account must be created firstly

Table 44: view statistics

Name	View statistics
Code	FR43
Priority	high
Critical	it is important to Farmer if need to view the statistics.
Description	it is an important for user if need to look on statistics details.
Input	-
Output	-
Pre-condition	must be at least 1 statistics stored in the database.
Post-condition	-
Dependency	the Farmer account must be created.
Risk	this process might not able to be satisfied when there is no Farmer account, and to reduce the probability of this occurrence, the Farmer account must be created firstly

Table 45: view news

Name	View news
Code	FR44
Priority	high
Critical	it is important to Farmer if need to view the news.
Description	it is an important for user if need to look on news.
Input	-
Output	-
Pre-condition	must be at least 1 new stored in the database.
Post-condition	-
Dependency	the Farmer account must be created.
Risk	this process might not able to be satisfied when there is no Farmer account, and to reduce the probability of this occurrence, the Farmer account must be created firstly

Table 46: view reports

Name	View reports
Code	FR45
Priority	high
Critical	it is important to expert if need to view the reports.
Description	it is an important for expert if need to look on report details.
Input	-
Output	-
Pre-condition	must be at least 1 report stored in the database.
Post-condition	-
Dependency	the expert account must be created.
Risk	this process might not able to be satisfied when there is no expert account, and to reduce the probability of this occurrence, the expert account must be created firstly

Table 47: view statistics

Name	View statistics
Code	FR46
Priority	high
Critical	it is important to expert if need to view the statistics.
Description	it is an important for expert if need to look on statistics details.
Input	-
Output	-
Pre-condition	must be at least 1 statistics stored in the database.
Post-condition	-
Dependency	the expert account must be created.
Risk	this process might not able to be satisfied when there is no expert account, and to reduce the probability of this occurrence, the expert account must be created firstly

Table 48: view news

Name	View news
Code	FR47
Priority	high
Critical	it is important to expert if need to view the news.
Description	it is an important for expert if need to look on news.
Input	-
Output	-
Pre-condition	must be at least 1 new stored in the database.
Post-condition	-
Dependency	the expert account must be created.
Risk	this process might not able to be satisfied when there is no expert account, and to reduce the probability of this occurrence, the expert account must be created firstly

Table 49: Encrypt password

Name	Encrypt password
Code	FR48
Priority	Extreme
Critical	it is very important to improve the security in our project.
Description	the function is used to translate password into another firm to keep it secured.
Input	password
Output	encrypted password
Pre-condition	none
Post-condition	password is added into the database in the hashed form successfully
Dependency	none the expert account must be created.
Risk	if the process might not able to be satisfied when there is no security for accounts, and to reduce the probability of this occurrence, the encrypted password must be created.

Table 50: Decrypt password

Name	Decrypt password
Code	FR49
Priority	Extreme
Critical	it is very important to improve the security in our project.
Description	the function is used to translate password back into its original form.
Input	hashed password
Output	none
Pre-condition	decrypted password
Post-condition	password returned to its normal state.
Dependency	none the expert account must be created.
Risk	if the process might not able to login , and to avoid this to occurrence, decrypted password must be allowed.

Table 51: Add time interval

Name	Add time interval
Code	FR50
Priority	Extreme
Critical	none.
Description	admin will able to set a time interval of a LED color to be turned on in a specific land.
Input	LED ID, Land ID and the time in hours.
Output	confirmation that the data is added successfully.
Pre-condition	Land ID and LED ID already exists. validate data entered. admin logged in.
Post-condition	the data will stored in the database.
Dependency	none the expert account must be created.
Risk	if the process might not able to login , and to avoid this to occurrence, decrypted password must be allowed.

Table 52: insert statuses

Name	insert statuses
Code	FR
Priority	high
Critical	
Description	
Input	date, weather station id, max temp , min temp , humidity, wind speed , wind direction , daylight, rain and moister.
Output	insert successfully.
Pre-condition	
Post-condition	
Dependency	FR
Risk	if the process might not able to predict disease , there is no idea for the project, to avoid this occurrence, the system must predict the diseases.

Table 53: predict disease

Name	predict disease
Code	FR51
Priority	Extreme
Critical	this function is extremely critical as its the main function that the system is built on.
Description	predict the diseases that will come to the crop.
Input	weather station data.
Output	probability.
Pre-condition	the system don't predict any disease yet.
Post-condition	the system make its calculation and get the best decision to improve the crop performance.
Dependency	FR
Risk	if the process might not able to predict disease , there is no idea for the project, to avoid this occurrence, the system must predict the diseases.

Table 54: fungicider

Name	fungicider
Table 55: insert statuses	
Name	insert statuses
Code	FR
Priority	high
Critical	
Description	
Input	date, weather station id, max temp , min temp , humidity, wind speed , wind direction , daylight, rain and moister.
Output	insert successfully.
Pre-condition	
Post-condition	
Dependency	FR
Risk	if the process might not able to predict disease , there is no idea for the project, to avoid this occurrence, the system must predict the diseases.
Code	FR
Priority	high
Critical	
Description	
Input	date, weather station id, max temp , min temp , humidity, wind speed , wind direction , daylight, rain and moister.
Output	insert successfully.
Pre-condition	
Post-condition	
Dependency	FR
Risk	if the process might not able to predict disease , there is no idea for the project, to avoid this occurrence, the system must predict the diseases.

5 Interface Requirements

5.1 User Interfaces

The system user interface is designed to be simple enough and allow minimal interaction. The user must first create an account on the application from the registration page, then log in to the application, when the user logs in, the home page will appear. The homepage consists of several items (Profile, Profile Modification, Crops, and Logout). The user can see the profile page to view his information only if it is necessary to make an update, and he must edit the profile from the previous page (the main page). User can select the crop he is working on from (crop page).

When the user selects the desired crop, the app will transfer it to 4 pages. First, the diseases page contains diseases that afflicted the selected crop based on climate change. Second, the report page that contains today's weather news. When the user next clicked, it would switch to the climate report. Third: the climate report that contains data on climate change on this day. Next, when the user clicks, they will be taken to the Resolutions page. Fourth: The decisions page that contains the best decisions for crop performance, which depend on accurate calculations.

5.1.1 GUI



Figure 14: Sign-in Page



Figure 15: Sign-up Page



Figure 16: Home Page



Figure 17: Profile page



Figure 18: Edit Profile Page

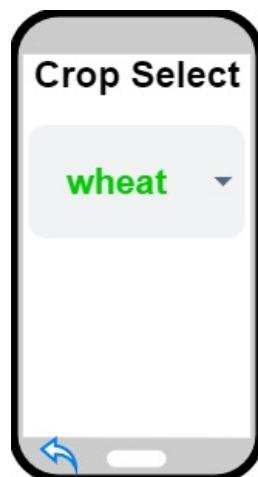


Figure 19: Crop Select Page

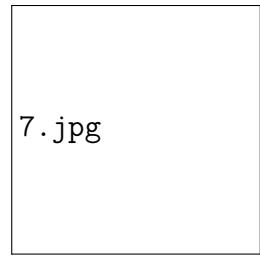


Figure 20: Diseases Page

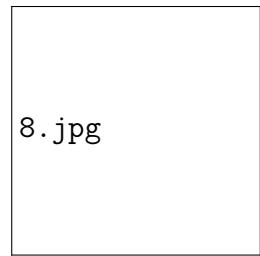


Figure 21: Report Page

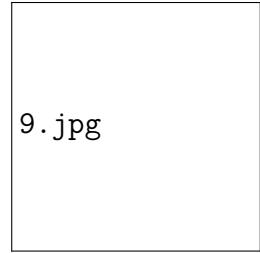


Figure 22: Climate report Page



Figure 23: Decisions page

5.2 Hardware Interfaces

N/A

5.3 Communications Interfaces

The system only needs internet connection.

5.4 API

- Weather forecast API
- Fungicides recommendation API
- Diseases forecast API

6 Design Constraints

- the system should be user friendly.
- any smart phone running either android or IOS
- Must have connection to the internet as it should deal with the real data transfer to the server.

6.1 Hardware Limitations

This system is in need of powerful hardware to be able to deal with the huge datasets proposed in this system

7 Non-functional Requirements

7.1 usability:

The system functionalities doesn't need time to be learned by the user.

7.2 portability:

The system is portable as it's a mobile based application, it is developed using flutter as it runs on Android and IOS devices.

7.3 Maintainability:

The system could be improved by different developers so ease of system maintainability is important, it should be easy to extend thought the implementation of MVC design pattern and using naming convention which ease the use of functions and understanding their purpose. MVC design pattern divides the system into three modules which are Model, View and controller, it simply separates handling of the data from the how the interface appears to the user and the intermediate communicator between both of them.

7.4 security:

Users passwords []must be hashed in the database. Users of our system shall authenticate themselves using their username and password. Also personal information about the users such as mobile numbers and passwords for instance must be protected.

7.5 performance and speed:

Every thing is being processed and computed on cloud to make the performance faster.

8 Data Design

8.1 Data Description

- The diseases models used in building the diseases prediction fuzzy models are collected from different research papers and books.
- Weather stations collect data about the land such as maximum and minimum temperature, humidity, rain, soil moisture, wind speed, wind direction and day light duration per day to insert them to the disease prediction model beside using the data in building a dataset to be used in the future to predict other diseases.
- The dataset that will be collected from the weather stations is expected to reach more than 500 rows by the end of the project.
- The expected number of users is about 50 users who are farmer and agriculture experts.
- The primary keys are integers that are auto incremented when the entries are inserted to the database.
- The Date format is stored using this format dd/mm/yyyy.

8.2 Database design description

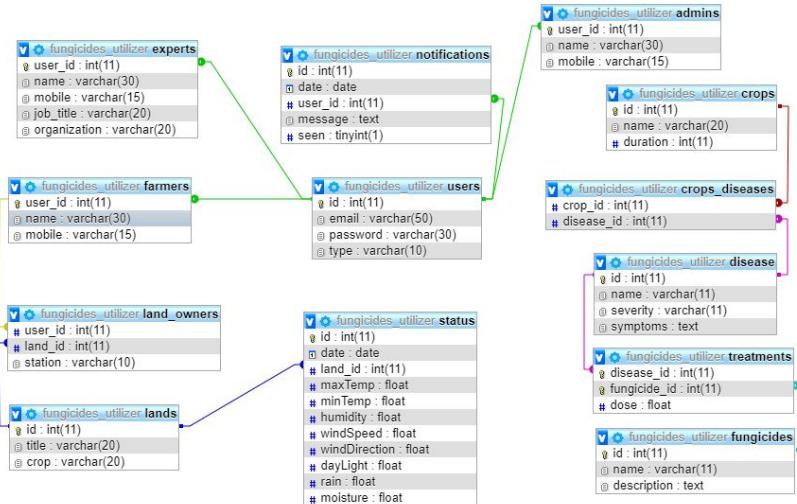


Figure 24: Database Schema

9 Preliminary Object-Oriented Domain Analysis

9.1 Inheritance Relationships

25.

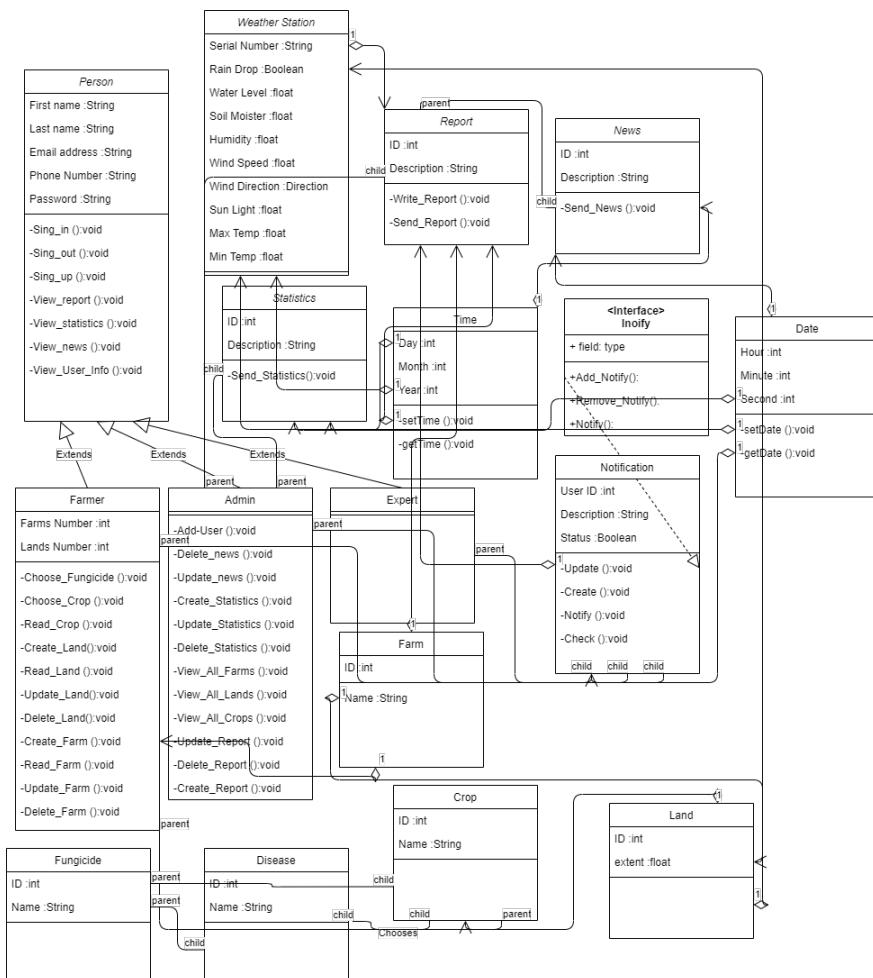


Figure 25: Class Diagram

9.2 Class descriptions

Table 56: Class Name - Person

Abstract or Concrete:	Concrete class.
List of Superclasses	None.
List of Subclasses	Admin, Farmer, Expert.
Purpose	Main class used to encapsulate different user types with their common attributes.
Collaborations	Admin, Farmer, Expert classes inherit from it .
Attributes	ID, UserID, FirstName, LastName, Mobile, Email, Password.
Operations	Login(String email, String password) Logout() Encrypt Pass(String pass) Decrypt Pass(string Hpass) View reports() View statistics() View news() view userInfo()
Constraints	None.

Table 57: Class Name - Admin

Abstract or Concrete:	Concrete class.
List of Superclasses	Person.
List of Subclasses	None.
Purpose	The user type admin which controls all system functionalities.
Collaborations	<p>This class inherit from class Person.</p> <p>Class Report associate class Admin, as admin can create, send and manipulate reports.</p> <p>Class Statistics associate Admin class, as he can create, send and manipulate statistics.</p> <p>Class Statistics associate News class, as he can create, send and manipulate news. Class Statistics associate Notification class, as he can create, send and manipulate Notification.</p>
Attributes	
Operations	<p>Add User()</p> <p>Delete News()</p> <p>Update News()</p> <p>Create News()</p> <p>Send News()</p> <p>create Statistics()</p> <p>Update Statistics()</p> <p>Delete Statistics()</p> <p>View ALLFarms()</p> <p>View AllLands()</p> <p>View AllCrops()</p> <p>Create Report()</p> <p>Update Report()</p> <p>Delete Report()</p> <p>Send Report()</p> <p>Create Notification()</p> <p>Update Notification()</p> <p>Check Notification()</p>
Constraints	None.

Table 58: Class Name - Farmer

Abstract or Concrete:	Concrete class.
List of Superclasses	Person.
List of Subclasses	None.
Purpose	The user type which interact and benefit with the system in his farm.
Collaborations	Inherit from Person class to obtain all its attributes and functions. aggregate from Land, Crop, Farm, classes to create objects. associate from Fungicide and Disease classes to choose which crop to apply for which disease.
Attributes	All Person class attributes, NumberOfFarms, NumberOfLand, object Land, Object Crop, Object Farm.
Operations	Choose Fungicide() Choose Crop() Read Crop() Create Land() Read Land() Update Land() Delete Land() Create Farm() Read Farm() Update Farm() Delete Farm()
Constraints	None.

Table 59: Class Name - Expert

Abstract or Concrete:	Concrete class.
List of Superclasses	Person class.
List of Subclasses	None.
Purpose	The user type which review the news, reports, statistics and data sets about crops and diseases.
Collaborations	Inherit from person class. associate with class news and statistics to view statistics and latest news about crops.
Attributes	All parent class attributes.
Operations	View News() View Statistics()
Constraints	None.

Table 60: Class Name - Crop

Abstract or Concrete:	Concrete class.
List of Superclasses	None.
List of Subclasses	None.
Purpose	object that is monitored by the system to apply fungicide and predict for disease.
Collaborations	Association with class Farmer, Admin and Fungicide classes. Aggregation with Land, Disease, Reports classes.
Attributes	CropID, CropName, ObjectDisease, ObjectReport.
Operations	Choose Crop() Add Crop() Choose Fungicide() Apply Fungicide() Disease Severity()
Constraints	None.

Table 61: Class Name - Disease

Abstract or Concrete:	Concrete class.
List of Superclasses	None.
List of Subclasses	None.
Purpose	Class that contains all data and functions about disease that threats the crops.
Collaborations	Aggregation with crop and Fungicide classes,
Attributes	Disease ID, Disease Name, severity, symptoms, Object Fungicide.
Operations	setters and getters.
Constraints	None.

Table 62: Class Name - Fungicide

Abstract or Concrete:	Concrete class.
List of Superclasses	None.
List of Subclasses	None.
Purpose	This class contains data about fungicides that will be applied on the crops to treat diseases
Collaborations	Aggregation with Disease class. association with crop class to apply on crop.
Attributes	Fungicide ID, Fungicide Name, Fungicide Dose.
Operations	Calculate Dose(crop x)
Constraints	None.

Table 63: Class Name - Land

Abstract or Concrete:	Concrete.
List of Superclasses	None.
List of Subclasses	None.
Purpose	Class land is a list of objects in class farm and it contains crops and weather station.
Collaborations	Aggregation with Farm, Crop, Farmer and Weather Station classes.
Attributes	Land ID, Farmer ID, Object Crop, Object Weather Station.
Operations	setter and getters.
Constraints	None.

Table 64: Class Name - Farm

Abstract or Concrete:	Concrete class.
List of Superclasses	None.
List of Subclasses	None.
Purpose	farm contains many lands and is owned by the Farmer.
Collaborations	Aggregation with class land as farm could contain many lands.
Attributes	Farm ID, Farmer ID, Farm Title, Lands number, Object land.
Operations	setters and getters
Constraints	None.

Table 65: Class Name - News

Abstract or Concrete:	Concrete class.
List of Superclasses	None.
List of Subclasses	None.
Purpose	class news contains data about the released news about crops, diseases, fungicides and agriculture that is viewed by Farmer and Expert . The admin is responsible for creating updating and deleting them.
Collaborations	association with classes statistics and Admin. aggregation with classes Date and Time.
Attributes	News ID, Description, Title, Object Date, Object Time.
Operations	Create News() Search News() Delete News() Send News()
Constraints	None.

Table 66: Class Name - Reports

Abstract or Concrete:	Concrete class.
List of Superclasses	None.
List of Subclasses	None.
Purpose	This class contains data and functions about reports of crops, lands, diseases and fungicides.
Collaborations	Aggregation with class Weather Station, Land. Association with classes crop.
Attributes	Report ID, Title Object Weather Station, Object Land,
Operations	Lists each operation that can be invoked upon instances of this class. For each operation, the arguments (and their type), the return value (and its type), and any side effects of the operation should be specified.
Constraints	Lists any restrictions upon the general state or behavior of instances of this class.

Table 67: Class Name - xxx

Abstract or Concrete:	Indicates whether this class is abstract or concrete.
List of Superclasses	Names all immediate superclasses.
List of Subclasses	List of Subclasses
Purpose	Purpose
Collaborations	Names each class with which this class must interact in order to accomplish its purpose, and how.
Attributes	Lists each attribute (state variable) associated with each instance of this class, and indicates examples of possible values (or a range).
Operations	Lists each operation that can be invoked upon instances of this class. For each operation, the arguments (and their type), the return value (and its type), and any side effects of the operation should be specified.
Constraints	Lists any restrictions upon the general state or behavior of instances of this class.

Table 68: Class Name - xxx

Abstract or Concrete:	Indicates whether this class is abstract or concrete.
List of Superclasses	Names all immediate superclasses.
List of Subclasses	List of Subclasses
Purpose	Purpose
Collaborations	Names each class with which this class must interact in order to accomplish its purpose, and how.
Attributes	Lists each attribute (state variable) associated with each instance of this class, and indicates examples of possible values (or a range).
Operations	Lists each operation that can be invoked upon instances of this class. For each operation, the arguments (and their type), the return value (and its type), and any side effects of the operation should be specified.
Constraints	Lists any restrictions upon the general state or behavior of instances of this class.

Table 69: Class Name - xxx

Abstract or Concrete:	Indicates whether this class is abstract or concrete.
List of Superclasses	Names all immediate superclasses.
List of Subclasses	List of Subclasses
Purpose	Purpose
Collaborations	Names each class with which this class must interact in order to accomplish its purpose, and how.
Attributes	Lists each attribute (state variable) associated with each instance of this class, and indicates examples of possible values (or a range).
Operations	Lists each operation that can be invoked upon instances of this class. For each operation, the arguments (and their type), the return value (and its type), and any side effects of the operation should be specified.
Constraints	Lists any restrictions upon the general state or behavior of instances of this class.

Table 70: Class Name - xxx

Abstract or Concrete:	Indicates whether this class is abstract or concrete.
List of Superclasses	Names all immediate superclasses.
List of Subclasses	List of Subclasses
Purpose	Purpose
Collaborations	Names each class with which this class must interact in order to accomplish its purpose, and how.
Attributes	Lists each attribute (state variable) associated with each instance of this class, and indicates examples of possible values (or a range).
Operations	Lists each operation that can be invoked upon instances of this class. For each operation, the arguments (and their type), the return value (and its type), and any side effects of the operation should be specified.
Constraints	Lists any restrictions upon the general state or behavior of instances of this class.

Table 71: Class Name - xxx

Abstract or Concrete:	Indicates whether this class is abstract or concrete.
List of Superclasses	Names all immediate superclasses.
List of Subclasses	List of Subclasses
Purpose	Purpose
Collaborations	Names each class with which this class must interact in order to accomplish its purpose, and how.
Attributes	Lists each attribute (state variable) associated with each instance of this class, and indicates examples of possible values (or a range).
Operations	Lists each operation that can be invoked upon instances of this class. For each operation, the arguments (and their type), the return value (and its type), and any side effects of the operation should be specified.
Constraints	Lists any restrictions upon the general state or behavior of instances of this class.

Table 72: Class Name - xxx

Abstract or Concrete:	Indicates whether this class is abstract or concrete.
List of Superclasses	Names all immediate superclasses.
List of Subclasses	List of Subclasses
Purpose	Purpose
Collaborations	Names each class with which this class must interact in order to accomplish its purpose, and how.
Attributes	Lists each attribute (state variable) associated with each instance of this class, and indicates examples of possible values (or a range).
Operations	Lists each operation that can be invoked upon instances of this class. For each operation, the arguments (and their type), the return value (and its type), and any side effects of the operation should be specified.
Constraints	Lists any restrictions upon the general state or behavior of instances of this class.

Table 73: Class Name - xxx

Abstract or Concrete:	Indicates whether this class is abstract or concrete.
List of Superclasses	Names all immediate superclasses.
List of Subclasses	List of Subclasses
Purpose	Purpose
Collaborations	Names each class with which this class must interact in order to accomplish its purpose, and how.
Attributes	Lists each attribute (state variable) associated with each instance of this class, and indicates examples of possible values (or a range).
Operations	Lists each operation that can be invoked upon instances of this class. For each operation, the arguments (and their type), the return value (and its type), and any side effects of the operation should be specified.
Constraints	Lists any restrictions upon the general state or behavior of instances of this class.

Table 74: Class Name - xxx

Abstract or Concrete:	Indicates whether this class is abstract or concrete.
List of Superclasses	Names all immediate superclasses.
List of Subclasses	List of Subclasses
Purpose	Purpose
Collaborations	Names each class with which this class must interact in order to accomplish its purpose, and how.
Attributes	Lists each attribute (state variable) associated with each instance of this class, and indicates examples of possible values (or a range).
Operations	Lists each operation that can be invoked upon instances of this class. For each operation, the arguments (and their type), the return value (and its type), and any side effects of the operation should be specified.
Constraints	Lists any restrictions upon the general state or behavior of instances of this class.

Table 75: Class Name - xxx

Abstract or Concrete:	Indicates whether this class is abstract or concrete.
List of Superclasses	Names all immediate superclasses.
List of Subclasses	List of Subclasses
Purpose	Purpose
Collaborations	Names each class with which this class must interact in order to accomplish its purpose, and how.
Attributes	Lists each attribute (state variable) associated with each instance of this class, and indicates examples of possible values (or a range).
Operations	Lists each operation that can be invoked upon instances of this class. For each operation, the arguments (and their type), the return value (and its type), and any side effects of the operation should be specified.
Constraints	Lists any restrictions upon the general state or behavior of instances of this class.

Table 76: Class Name - xxx

Abstract or Concrete:	Indicates whether this class is abstract or concrete.
List of Superclasses	Names all immediate superclasses.
List of Subclasses	List of Subclasses
Purpose	Purpose
Collaborations	Names each class with which this class must interact in order to accomplish its purpose, and how.
Attributes	Lists each attribute (state variable) associated with each instance of this class, and indicates examples of possible values (or a range).
Operations	Lists each operation that can be invoked upon instances of this class. For each operation, the arguments (and their type), the return value (and its type), and any side effects of the operation should be specified.
Constraints	Lists any restrictions upon the general state or behavior of instances of this class.

10 Operational Scenarios

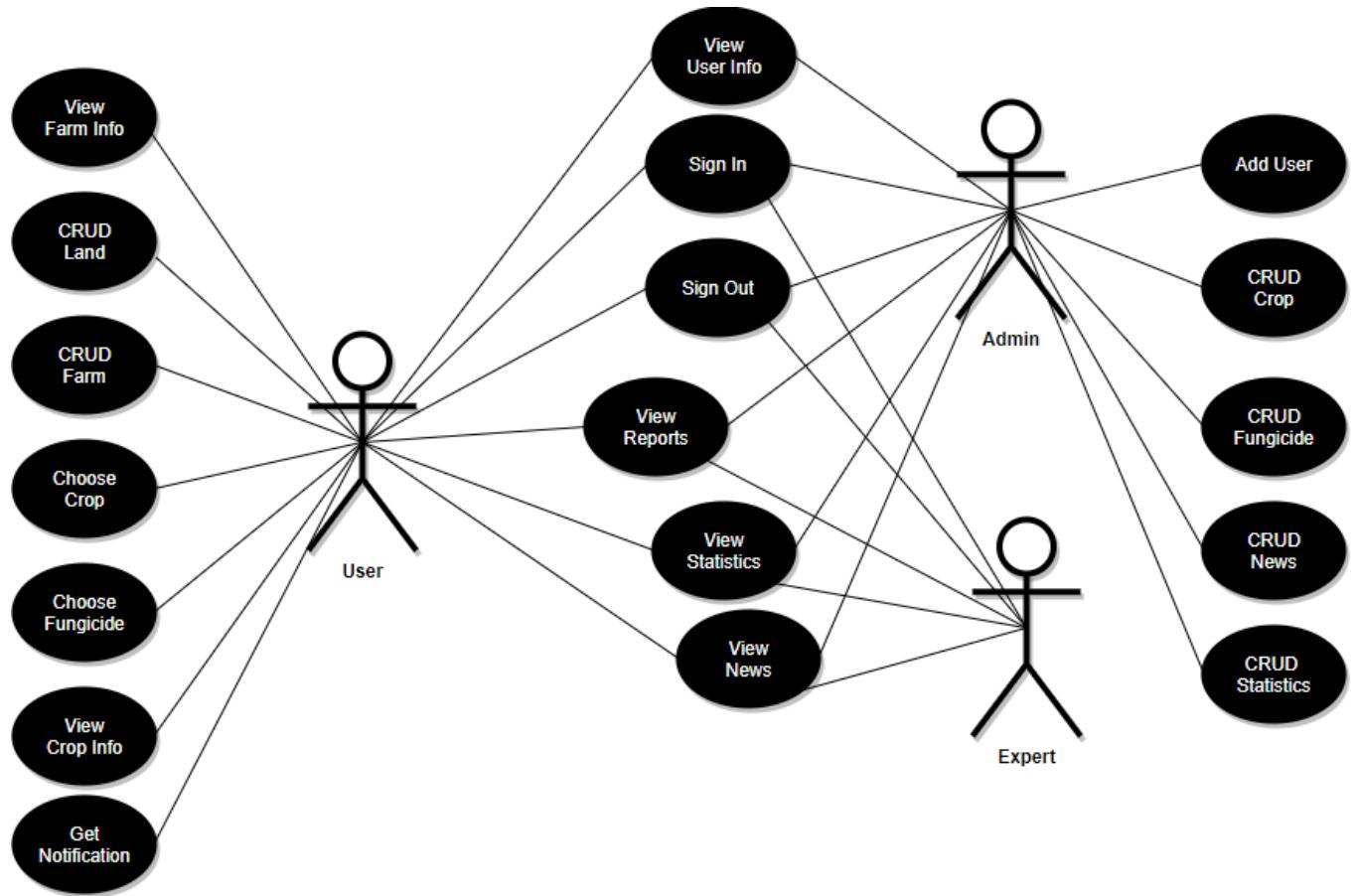


Figure 26: System UseCase

Function:	ADD User
Actors:	Admin
Description:	An Admin accepts then adds a new user.
Data:	The users data
Response:	User added successfully
Comments:	Admin must be signed in.

Function:	Search and View User
Actors:	Admin
Description:	An Admin search and view all user.
Data:	User's ID
Response:	User are displayed
Comments:	Admin must be signed in.

Function:	Create crop.
Actors:	Admin.
Description:	Admin add a new crop type.
Data:	Crop name, Crop type, diseases that affects crop.
Response:	Crop added successfully.
Comments:	the crops that system works with, the admin must be logged in.

Function:	Search and Read Crop.
Actors:	Admin, User.
Description:	Admin or user can search and view crops.
Data:	Crop ID.
Response:	Crops are displayed.
Comments:	Admin or User must be logged in.

Function:	Update crop.
Actors:	Admin.
Description:	Admin can update crop info.
Data:	Crop ID.
Response:	Crop updated successfully.
Comments:	Admin must be logged in.

Function:	Delete Crop.
Actors:	Admin,
Description:	Admin and only admin can delete certain crop.
Data:	Crop ID.
Response:	Crop deleted successfully.
Comments:	Admin must be logged in.

Function:	Create Fungicide.
Actors:	Admin.
Description:	Admin add a new Fungicide.
Data:	Fungicide name, Fungicide type, Fungicide dose, diseases that is treated.
Response:	Fungicide added successfully.
Comments:	Admin must be logged in.

Function:	Search and Read Fungicide.
Actors:	Admin, User.
Description:	Admin or user can search and view Fungicides.
Data:	Fungicide ID.
Response:	Fungicides are displayed.
Comments:	Admin or User must be logged in.

Function:	Update Fungicide info.
Actors:	Admin.
Description:	Admin can update crop info.
Data:	Fungicide ID.
Response:	Fungicide info updated successfully.
Comments:	Admin must be logged in.

Function:	Delete Fungicide.
Actors:	Admin,
Description:	Admin and only admin can delete certain Fungicide.
Data:	Fungicide ID.
Response:	Fungicide deleted successfully.
Comments:	Admin must be logged in.

Function:	Create News.
Actors:	Admin.
Description:	Admin add a new News.
Data:	News attributes and description.
Response:	News added successfully.
Comments:	Admin must be logged in.

Function:	Search and Read News.
Actors:	Admin, User, Expert.
Description:	Admin or user or Expert can search and view News.
Data:	News ID or title or date.
Response:	News are displayed.
Comments:	Admin or User or expert must be logged in.

Function:	Update News info.
Actors:	Admin.
Description:	Admin can update News.
Data:	News ID.
Response:	News info updated successfully.
Comments:	Admin must be logged in.

Function:	Delete News.
Actors:	Admin,
Description:	Admin and only admin can delete News.
Data:	News ID.
Response:	News deleted successfully.
Comments:	Admin must be logged in.

Function:	Create Statistics.
Actors:	Admin.
Description:	Admin add a new Statistics.
Data:	Statistics attributes and description.
Response:	Statistics added successfully.
Comments:	Admin must be logged in.

Function:	Search and Read Statistics.
Actors:	Admin, User, Expert.
Description:	Admin or user or Expert can search and view Statistics.
Data:	Statistics ID or name or date.
Response:	Statistics are displayed.
Comments:	Admin or User or expert must be logged in.

Function:	Update Statistics info.
Actors:	Admin.
Description:	Admin can update Statistics.
Data:	Statistics ID.
Response:	Statistics info updated successfully.
Comments:	Admin must be logged in.

Function:	Delete Statistics.
Actors:	Admin,
Description:	Admin and only admin can delete Statistics.
Data:	Statistics ID.
Response:	Statistics deleted successfully.
Comments:	Admin must be logged in.

Function:	Create Reports.
Actors:	Admin.
Description:	Admin add a new reports.
Data:	Report attributes and description.
Response:	Reports added successfully.
Comments:	Admin must be logged in.

Function:	Search and Read Reports.
Actors:	Admin, User, Expert.
Description:	Admin or user or Expert can search and view Reports.
Data:	Reports ID or name or date.
Response:	Reports are displayed.
Comments:	Admin or User or expert must be logged in.

Function:	Update Reports info.
Actors:	Admin.
Description:	Admin can update Reports.
Data:	Reports ID.
Response:	Reports info updated successfully.
Comments:	Admin must be logged in.

Function:	Delete Reports.
Actors:	Admin,
Description:	Admin and only admin can delete Reports.
Data:	Reports ID.
Response:	Reports deleted successfully.
Comments:	Admin must be logged in.

Function:	Create Land.
Actors:	User.
Description:	User add a new Land.
Data:	Land attributes and description.
Response:	New land added successfully.
Comments:	User must be logged in.

Function:	Search and Read Land.
Actors:	Admin, User.
Description:	Admin or user or can search and view Land.
Data:	Land ID or owner name.
Response:	Land are displayed.
Comments:	Admin or User must be logged in.

Function:	Update Land info.
Actors:	User.
Description:	User can update land info.
Data:	Land ID.
Response:	Land info updated successfully.
Comments:	User must be logged in.

Function:	Delete Land.
Actors:	User
Description:	User can delete Land.
Data:	Land ID.
Response:	Land deleted successfully.
Comments:	User must be logged in.

Function:	Add Farm.
Actors:	User.
Description:	User add a new Farm.
Data:	Farm attributes and description.
Response:	New Farm added successfully.
Comments:	User must be logged in.

Function:	Search and Read Farms.
Actors:	Admin, User.
Description:	Admin or user or can search and view Farm.
Data:	Farm ID or owner name.
Response:	Farm are displayed.
Comments:	Admin or User must be logged in.

Function:	Update Farm info.
Actors:	User.
Description:	User can update Farm info.
Data:	Farm ID.
Response:	Farm info updated successfully.
Comments:	User must be logged in.

Function:	Delete Farm.
Actors:	User
Description:	User can delete Farm.
Data:	Farm ID.
Response:	Farm deleted successfully.
Comments:	User must be logged in.

Function:	Choose Crop.
Actors:	User.
Description:	The user selects and chooses the crop to keep track on from provided system's crops.
Data:	Crop ID, name.
Response:	Crop selected successfully.
Comments:	User must be logged in.

Function:	Choose Fungicide.
Actors:	User.
Description:	The user selects and chooses from recommended fungicides.
Data:	Fungicide ID, name.
Response:	Fungicide selected successfully.
Comments:	User must be logged in.

Function:	Read and view crop info.
Actors:	User.
Description:	The user view his crop status, health and info .
Data:	Crop ID, name.
Response:	Crop Status viewed successfully.
Comments:	User must be logged in.

Function:	Alert.
Actors:	User.
Description:	The user receives alerts and notification about his crops, land, farm and news about weather climate.
Data:	alert ID.
Response:	pop-ups alerts.
Comments:	User must be logged in.

Function:	View and update user info
Actors:	User
Description:	The user can edit his information
Data:	User attributes
Response:	Successfully updated
Comments:	User must be logged in.

Function:	Sign In
Actors:	Admin, User, Expert.
Description:	Actors sign in to use the system
Data:	User name, password.
Response:	Welcome logged in successful. or incorrect entry.
Comments:	The system needs to have access to internet.

Function:	Sign Out.
Actors:	Admin, User, Expert.
Description:	Actors sign out to end session and disconnect their profile.
Data:	Session ID.
Response:	Good bye, logged out successful.
Comments:	The system needs to have access to internet.

Function:	
Actors:	
Description:	
Data:	
Response:	
Comments:	

11 Project Plan

Task	Start Date	End Date
Proposal evaluation	12/10/2020	26/10/2020
Writing SRS document	26/11/2020	28/12/2020
SRS evaluation	28/12/2020	29/12/2020
Writing SSD document	1/1/2020	12/2/2020
SDD evaluation	12/2/2020	15/2/2020
System implementation	25/4/2020	21/5/2020
Implementation evaluation	21/5/2020	24/5/2020
Final presentation	4/7/2020	4/7/2020

27.

77 shows a simple example.

Table 77: Project name time plan

Project Plan				
Id	Task	Start Date	Number of Days	Team Member
1	Collect Dataset	12/10/2020	10	X, Y
2	Work on GUI	12/21/2020	15	Z, Y
3	Pre-Processing	12/21/2020	5	X
4	Feature Extraction	12/26/2020	5	X, Z, Y
5	Classification	12/31/2020	10	Z
6	Writing Paper	01/05/2021	30	X, Z, Y
7	Experiments	01/10/2021	20	X, Z, Y

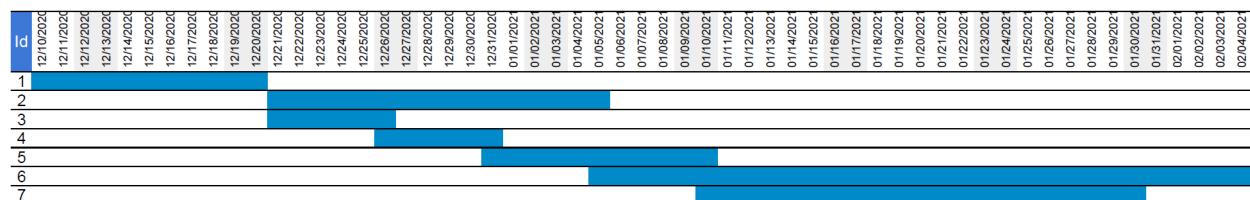


Figure 27: Project name GANTT Chart

12 Appendices

Specifies other useful information for understanding the requirements. All SRS documents should include at least the following two appendices:

12.1 Definitions, Acronyms, Abbreviations

- UI: User interface is the space where interactions between humans and machines occur.
- GUI: Graphical user interface.
- API: An application programming interface (API) is a set of routines, protocols, and tools for building software applications.
- FR: Functional Requirement.

12.2 Supportive Documents

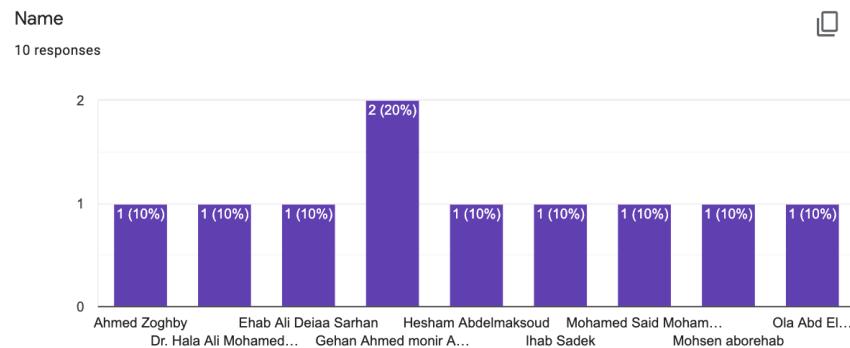


Figure 28: Form Responses

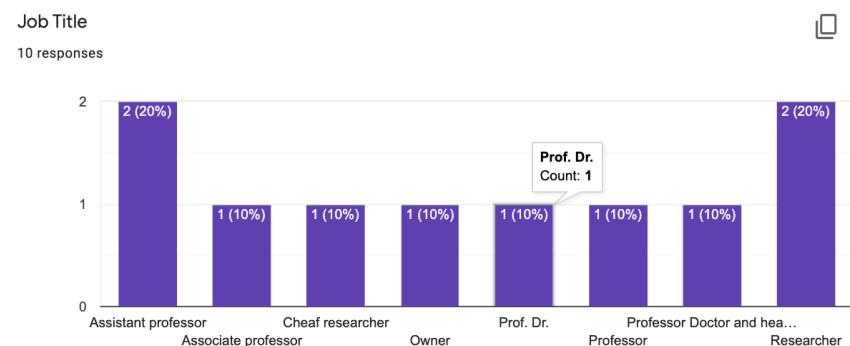


Figure 29: Form Responses

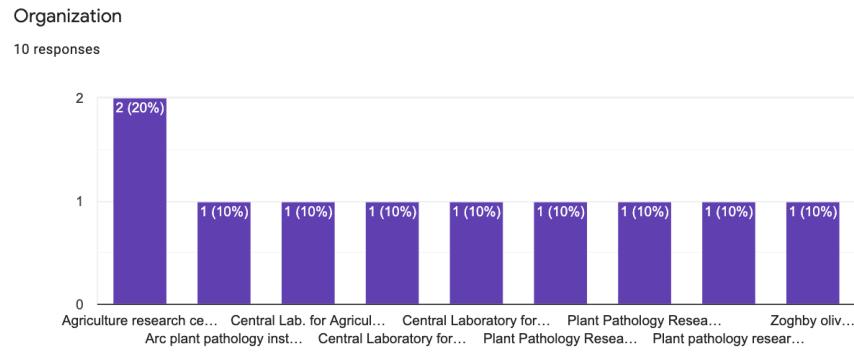


Figure 30: Form Responses

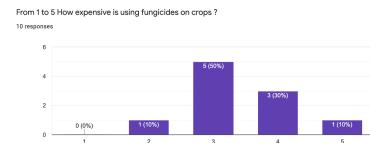


Figure 31: Form Responses

Have you used an agriculture technology system before?
10 responses

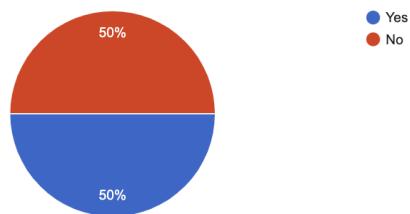


Figure 32: Form Responses

Did you hear about an agriculture system that identify the amount of fungicides needed before?

10 responses

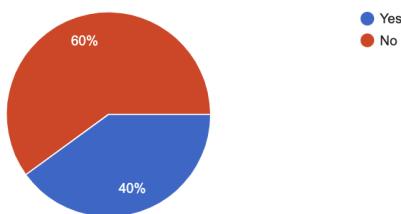


Figure 33: Form Responses

From 1 to 5 in your opinion how much can technology improve the agriculture sector?

10 responses

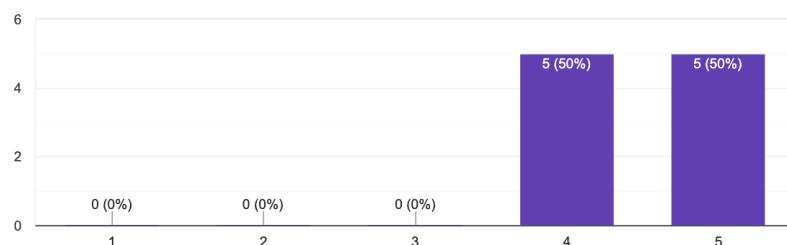


Figure 34: Form Responses

From 1 to 5 How much are the agriculture sector need to use the Smart Farm system?

10 responses

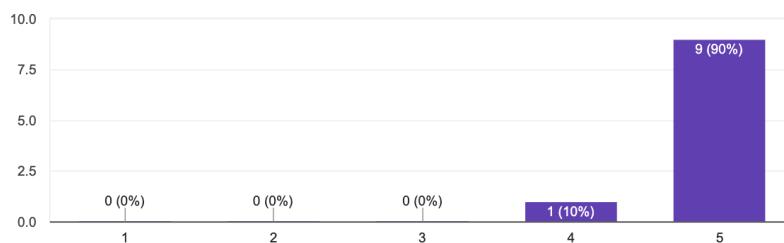


Figure 35: Form Responses

Stage	Temperature (°C)				
	Minimum	Optimum	Maximum	Light	Free water
Leaf rust					
Germination	2	20	30	Low	Essential
Germling	5	15-20	30	Low	Essential
Appressorium		15-20		None	Essential
Penetration	10	20	30	No effect	Essential
Growth	2	25	35	High	None
Sporulation	10	25	35	High	None
Stem rust					
Germination	2	15-24	30	Low	Essential
Germling		20		Low	Essential
Appressorium		16-27		None	Essential
Penetration	15	29	35	High	Essential
Growth	5	30	40	High	None
Sporulation	15	30	40	High	None
Stripe rust					
Germination	0	9-13	23	Low	Essential
Germling		10-15		Low	Essential
Appressorium			(not formed)		
Penetration	2	8-13	23	Low	Essential
Growth	3	12-15	20	High	None
Sporulation	5	12-15	20	High	None

Figure 36: DR.Khadega's research model



Figure 37: contacting wheat expert for her work

13 References

30 Reference Minimum (80% Academic)

References

- [1] Samuel Sahile, Chemeda Fininsa, PK Sahuja, et al. “Effect of mixed cropping and fungicides on chocolate spot (*Botrytis fabae*) of faba bean (*Vicia faba*) in Ethiopia”. In: *Crop protection* 27.2 (2008), pp. 275–282.
- [2] H Geoffrey Hewitt et al. *Fungicides in crop protection*. Cab International, 1998.
- [3] Qingxin Xiao, Weilu Li, Yuanzhong Kai, et al. “Occurrence prediction of pests and diseases in cotton on the basis of weather factors by long short term memory network”. In: *BMC bioinformatics* 20.25 (2019), p. 688.
- [4] Spyridon Mourtzinis, David Marburger, John Gaska, et al. “Corn and soybean yield response to tillage, rotation, and nematicide seed treatment”. In: *Crop Science* 57.3 (2017), pp. 1704–1712.
- [5] Sonja Brodt, J Six, G Feenstra, et al. “Sustainable agriculture”. In: *Nat. Educ. Knowl* 3.1 (2011).
- [6] Maria Tackenberg, Christa Volkmar, Michael Schirrmann, et al. “Impact of sensor-controlled variable-rate fungicide application on yield, senescence and disease occurrence in winter wheat fields”. In: *Pest management science* 74.6 (2018), pp. 1251–1258.
- [7] Warren E Copes, Austin Hagan, and John Olive. “Timing of fungicides in relation to calendar date, weather, and disease thresholds to control Rhizoctonia web blight on container-grown azalea”. In: *Crop Protection* 42 (2012), pp. 273–280.
- [8] Sukumar Chakraborty, R Ghosh, M Ghosh, et al. “Weather-based prediction of anthracnose severity using artificial neural network models”. In: *Plant Pathology* 53.4 (2004), pp. 375–386.
- [9] Ravi P Singh, Pawan K Singh, Jessica Rutkoski, et al. “Disease impact on wheat yield potential and prospects of genetic control”. In: *Annual review of phytopathology* 54 (2016), pp. 303–322.
- [10] André C Velásquez, Christian Danve M Castroverde, and Sheng Yang He. “Plant–pathogen warfare under changing climate conditions”. In: *Current Biology* 28.10 (2018), R619–R634.
- [11] RA McIntosh and ZA Pretorius. *Borlaug Global Rust Initiative provides momentum for wheat rust research*. 2011.
- [12] Marlon Tagliapietra Stefanello, Ricardo Silveiro Balardin, Simone Gripa Minuzzi, et al. “Effect of the interaction between fungicide application time and rainfall simulation interval on Asian Soybean Rust control effectiveness”. In: *Semina: Ciências Agrárias* 37.6 (2016), pp. 3881–3892.
- [13] Robert Park, Tom Fetch, Dave Hodson, et al. “International surveillance of wheat rust pathogens: progress and challenges”. In: *Euphytica* 179.1 (2011), pp. 109–117.

- [14] Munees Ahemad and Mohammad Saghir Khan. “Biotoxic impact of fungicides on plant growth promoting activities of phosphate-solubilizing *Klebsiella* sp. isolated from mustard (*Brassica campestris*) rhizosphere”. In: *Journal of Pest Science* 85.01 (2012), pp. 29–36.
- [15] Geoffrey L Bateman and Bruce DL Fitt. “Effects of fungicides on eyespot in five successive crops of winter wheat in plots with different initial or fungicide-selected populations of *Pseudocercosporella herpotrichoides*”. In: *Crop Protection* 10.6 (1991), pp. 479–484.
- [16] DV Richmond and RJ Pring. “Some cytological effects of systemic fungicides on fungi and plants”. In: *Netherlands Journal of Plant Pathology* 83.1 (1977), pp. 403–410.
- [17] NV Hardwick. “Weather and plant diseases”. In: *Weather* 57.5 (2002), pp. 184–190.
- [18] Helen Ogle. “Other biotic causes of plant diseases”. In: *Plant pathogens and plant diseases. Australasian Plant Pathology Society, Australia* (1997), pp. 143–145.
- [19] Jim Isleib. “Signs and symptoms of plant disease: Is it fungal, viral or bacterial”. In: *Michigan State University Extension* (2012).
- [20] Alan P. Roelfs, R. P. Singh, and E. E. Saari. *Rust diseases of wheat: concepts and methods of disease management*. CIMMYT, 1992.
- [21] James A Kolmer. “Tracking wheat rust on a continental scale”. In: *Current opinion in plant biology* 8.4 (2005), pp. 441–449.
- [22] David H Gent and Howard F Schwartz. “Validation of potato early blight disease forecast models for Colorado using various sources of meteorological data”. In: *Plant disease* 87.1 (2003), pp. 78–84.
- [23] Ricardo Brustolin, Sandra Maria Zoldan, Erlei Melo Reis, et al. “Weather requirements and rain forecast to time fungicide application for Fusarium head blight control in wheat”. In: *Summa Phytopathologica* 39.4 (2013), pp. 248–251.
- [24] Odile Carisse and Vanessa McNealis. “Identification of weather conditions associated with the occurrence, severity, and incidence of black seed disease of strawberry caused by *Mycosphaerella fragariae*”. In: *Phytopathology* 108.1 (2018), pp. 83–93.
- [25] K-H Dammer, H Thöle, T Volk, et al. “Variable-rate fungicide spraying in real time by combining a plant cover sensor and a decision support system”. In: *Precision agriculture* 10.5 (2009), pp. 431–442.
- [26] Lauren A Rhodes and Bruce A McCarl. “An Analysis of Climate Impacts on Herbicide, Insecticide, and Fungicide Expenditures”. In: *Agronomy* 10.5 (2020), p. 745.
- [27] Halah A.M. El – Dakar Moustafa M.S.H and A.A.Ismail. “Adverse Effect of Rainfall on the Efficacy of some Fungicides Used to Control Chocolate Spot of Faba Bean (*Vicia faba*)”. In: *International Journal of Scientific Engineering Research, Volume 6, Issue 12, 6.12* (2015), pp. 1–10.
- [28] O Christen, K Sieling, and H Hanus. “The effect of different preceding crops on the development, growth and yield of winter wheat”. In: *European Journal of Agronomy* 1.1 (1992), pp. 21–28.

- [29] Savita Kolhe, Raj Kamal, Harvinder S Saini, et al. “A web-based intelligent disease-diagnosis system using a new fuzzy-logic based approach for drawing the inferences in crops”. In: *Computers and Electronics in Agriculture* 76.1 (2011), pp. 16–27.
- [30] IA Rahmon. “A Neuro-Fuzzy System For Diagnosis of Soya-Beans Diseases”. In: () .