GROWTH PREDICTION, DISEASES DETECTION & CLASSIFICATION SYTEM FOR ANTHURIUM PLANTS

Project Id: 17-082

Software Requirement Specification

P.D.P Madavika

BSc. Special (Hons) In Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

May 2017

GROWTH PREDICTION, DISEASES DETECTION & CLASSIFICATION SYSTEM FOR ANTHURIUM PLANTS

Project Id: 17-082

Software Requirement Specification

(Software Requirement Specification Documentation Submitted In Partial Fulfilment Of The Requirement For The Degree Of Bachelor Of Science Special (Honors)

In Information Technology

BSc. Special (Hons) In Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

May 2017

DECLARATION

“I declare that the project would involve material prepared by the group members and that it would not fully or partially incorporate any material prepared by other person for a fee or free of charge or that it would include material previously submitted by a candidate for a Degree or Diploma in any other University or Institute of Higher Learning and that, to the best of my knowledge and belief, it would not incorporate any material previously published or written by another person in relation to another project except with prior written approval from the supervisor and/or the coordinator of such project and that such unauthorized reproductions will construe offences punishable under the SLIIT regulations.

I am aware, that if I am found guilty for the above mentioned offences or any project related plagiarism, the SLIIT has right to suspend the project at any time and or to suspend me from the examination and or from the Institution for minimum period of one year”.

Student Name	Student ID	Date of Submission	Signature
Madavika P.D.P.	IT14045554	2nd May 2017	

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor:…………………….

Signature of the co-supervisor:…………………….

Table of Contents

List of Tables

List of Figures

1. Introduction

The Software Requirement Specification (SRS) gives readers a detailed description of the software to be developed. This mainly contains the functional and non- functional requirements of the system, use case diagrams, use case scenarios, interface requirements and other design and memory constrains that helps to get an idea on how the user interacts with the system. This will be very useful to the software engineers and quality assurance engineers who will build and maintain the system.

1.1 Purpose

The key purpose of this SRS document is to provide a detailed description of the functional and non-functional requirements of the standalone application planned to be implemented at the end of our research. This provides the system overview, interfaces and technologies that are to be used in implementing the software. The intended audience of this document is the Project supervisor and the research team, but will also benefit the former researchers, developers and maintaining team who will be engaging in developing and maintaining the application in the future and any person, who is interested implementing application for plant diseases in image processing.

Overall, this document is written with the purpose of providing a complete description on the behavior of the “classification, quantification and providing necessary treatments to overcome diseases” functionality. For the better understanding of the above functionality, this document includes its sub components, how each sub component works and the relationship between those sub components to the end users and developers. This document is prepared under the guidance of the supervisor and co-supervisor after a literature survey. The requirements of the product discussed throughout the document are expected to be met at the end of project.

1.2 Scope

Agro is a user friendly stand-alone software application. This application does classification of the diseases, monitoring the growth of a plantation by considering the external environmental factors and predicting the outbreak of a disease related to environmental factors as with naked eye observation it is difficult. All these aspects are described clearly throughout the document.

The document further covers the objectives of the software. The main objective of the project is to identify diseases spread among the Anthurium as well as to predict diseases can be occurred due to environmental factors like humidity and temperature etc. Main focus is to prevent spreading and occurring diseases which effect the growth of the plant by warning the owners to pay attention.

The specific objectives are described below.

1. Feature extraction

   In order to identify the diseases properly through the color changes of the leaves and the flower of Anthurium. To extract necessary details method details extraction methods, have to be used. All other processes depend on how the image details are extracted.

2. Detection of diseases

   Objective of this process is to identify the disease which is spreading. The spot colour change is taken into consideration to classify a disease. Then the quantification, causes and treatments are given as text outputs.

3. Predict growth according to the environmental conditions

   Environmental factors like sun light, humidity and temperature can be taken through different sensors accordingly and analyzed to monitor the growth of a whole plantation.

4. Detection of Severity level of the Anthurium diseases by considering environmental conditions.

   Environmental factors like sun light, humidity and temperature can be taken through different sensors and analyzed to predict whether a disease is appearing or not in a certain plantation.

The main goal of the application is to provide a solution for large scale Anthurium growers to maintain their plantations healthy. The benefits of the software are cost effective and less time consuming. When the grower first identifies a disease before spreading, they can reduce the cost spent on fertilizer and other chemicals. They can also treat one infected plant with less time than treating many infected plants.

The document specifically covers all the functional and nonfunctional requirements, technology overview, class diagrams and interface requirements related to the disease classification, quantification and providing causes and treatments.

1.3 Definitions, Acronyms and Abbreviations

Term	**Definition**
SRS	Software Requirement Specification
UML	User Modeling Language
GUI	Graphical User Interface
User	Someone who interacts with the stand-alone application
IDE	Integrated Development Environment
OS	Operating System
SD Card	Secure Digital Card

Table 1: Definitions, acronyms and abbreviations

1.4 References

All the references referred to prepare the SRS document are stated at the end of the document.

1.5 Overview

1.5.1 Main goal

Main goal of the proposing application is to provide a user friendly solution for various problems faced by large scale Anthurium growers in looking after their plantations using new technologies. This provides an overall solution to maintain a healthy plantation with minimum effort.

1.5.2 Users

The mainly target audience of the application are large scale Anthurium growers in Sri Lanka who like to deal with technology and spend a lot of money and time to maintain their plantations as this is a very expensive plant.

1.5.3 Tasks

1. Able to classify, quantify and provide causes and treatments to overcome the disease.

2. Able to predict the growth of a plantation relative to environmental conditions.

3. Able to detect of Severity level of the Anthurium diseases by considering environmental conditions.

1.5.4 Organization of the SRS

A detailed description of the organization of how the SRS is organized is given below.

Chapter 1 contains the introduction, purpose of the software, scope, and overview of the system, definitions, acronyms and abbreviations in the document.

Chapter 2 defines the overall description of the system in terms of product perspective, product functions, constraints, apportioning requirements, user characteristics and assumptions.

Chapter 3 contains the system requirements such as interface requirements, performance requirements and software system attributes.

Chapter 4 describes contains supporting information for the readers of this document

2. Overall description

Agro is a user friendly stand-alone application developed targeting the main users as large scale Anthurium growers. The main target of the application is to warn the users about the spread of the diseases, predict the outbreak of a disease with related to environmental conditions and monitoring the growth of a plantation considering humidity, temperature and sun light. The main problem addressed here are the difficulty in naked eye observation of each and every plant, difficulty in monitoring the plantation growth with the sudden changes in the environmental factors and difficulty in predicting the outbreak of a disease with related to environmental conditions.

The accuracy of the application is expected to be high. The growers can maintain a healthier plantation through the application as it covers every aspect that a grower should see when growing Anthurium currently done through naked eye observation. The security of the application is also very high and newest technologies are uses for implementation process. Therefore users have an opportunity to get familiar with the new technologies through the application.

The description of the functions of the application is as follows.

Feature Extraction

Both healthy and infected parts of the Anthurium plant like spathe, spadix and leaves are considering for the study. In order to identify the parts of the Anthurium plant as healthy or infected, segmentation techniques help to fulfill that task. The segmented parts are used to extract features like shape, texture and color for the future steps in the study.

Growth Prediction

For the growth prediction, we consider the green area of leaves relative to the environmental factors like sun light, temperature and humidity. Using sensors those factors are taken to a raspberry pi and then processing is done. Then processed data will be stored in the database. With respective to the prevailing conditions green area of leaves are calculated. Using a neural network with the available details prediction is done and measure how much prevailing factors will effect on the future growth.

Classification and quantification of the diseases

This functionality is mainly about the classification of a certain disease identified using the colour variation of the spots. When interacting with the growers, we found out that for different diseases, the colour of the spots are different. The growers also identify the diseases through naked eye observations by the spot colour. Since it is difficult to do this with every plant every time, this application helps with the classification process. The application warns the user when some colour of a spot has occurred in leaf, spathe or spadix. To quantify the disease means measuring how much the disease has spread throughout the plant. This can be stated as a percentage. Then the user is acknowledged with different treatments and pesticides that can be used to minimize the disease using text outputs. This component gives the first clue of what really happens and warn the growers which in turn leads to increase the accuracy of the proposed automatic system.

Detection of Severity level of the Anthurium diseases by considering environmental conditions.

Environmental conditions are a main factor that affects for the healthy growth of a plant. Main environmental conditions that affects are sunlight, temperature and humidity of the atmosphere. In the proposed system we are considering about these factors and planning to give the severity level for diseases that can affects for the plantation that happened due to the changes of the environmental factors. Then by considering past history of data, neural network will predict the severity level of affecting those diseases.

2.1 Product perspective

This is a stand-alone application installed on a personal computer or a laptop. The user does not need an internet connection to use this application.

Following are the existing systems available and the comparisons with Agro stand-alone application. In all of the below applications, the external environmental factors are not considered. All the aspects like growth prediction, disease classification and predicting the outbreak of a disease related to environmental conditions have not been built in one single application so far.

Title	**Publication & Year**	**Author**	**Crop/ plant**	**Consider Growth of the Plant**	**Possibility to use in a real plantation**	**External factors considered**
Potato leaf diseases detection and classification system.	IJSMC, 2016.	Mr. Girish Athanikar, Ms.Priti Badar	Potato	NO	Weak	NO
An Overview on Detection and Classification of Plant Diseases in Image Processing.	IJSER, 2014.	Nikita Rishi, Jagbir Singh Gill	Cotton leaf spot, Rice plant, Wheat and sugar beet, Orchid leaf, Apple fruit, Chili plant	NO	Weak	NO
Leaf Disease Detection using Image Processing Technique.	IJIREEICE, 2016.	Piyali Chatterjee1, B. Harikishor Rao2	Maize	NO	Weak	NO
Feasibility Study on Plant Chili Disease Detection Using Image Processing Techniques.	IEEE, 2012.	Zulkifli Bin Husin, Ali Yeon Bin Md. Shakaff, Abdul Hallis Bin	Chili	NO	Weak	NO

		Abdul Aziz, Abdul Hallis bin Abdul Aziz.				
Digital image processing techniques for detecting, quantifying and classifying plant diseases	Barbedo Springer Plus, 2013.	Jayme Garcia Arnal Barbedo	Vegetable pathology	NO	Weak	NO
Proposed System-Agro	Not yet published.		Anthurium	Yes	Excellent	Yes

Table 2: Checking of publications published for Anthurium plant with respect to different factors.

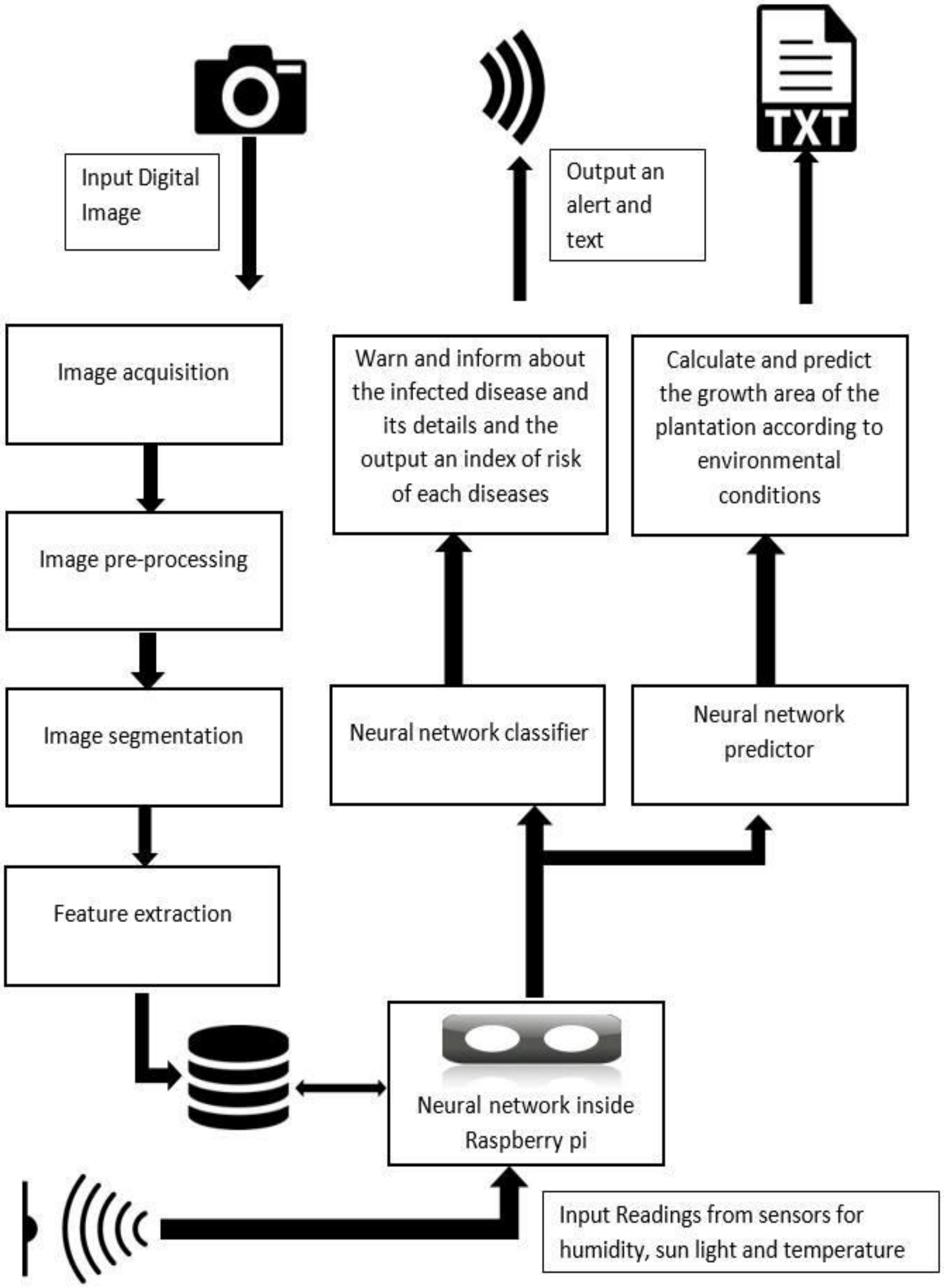

Input Digital Image
Output an alert and text
TXT
Image acquisition
Image pre-processing
Image segmentation
Feature extraction
Warn and inform about the infected disease and its details and the output an index of risk of each diseases
Calculate and predict the growth area of the plantation according to environmental conditions
Neural network classifier
Neural network predictor
Neural network inside Raspberry pi
Input Readings from sensors for humidity, sun light and temperature

Figure 1: Overall system diagram

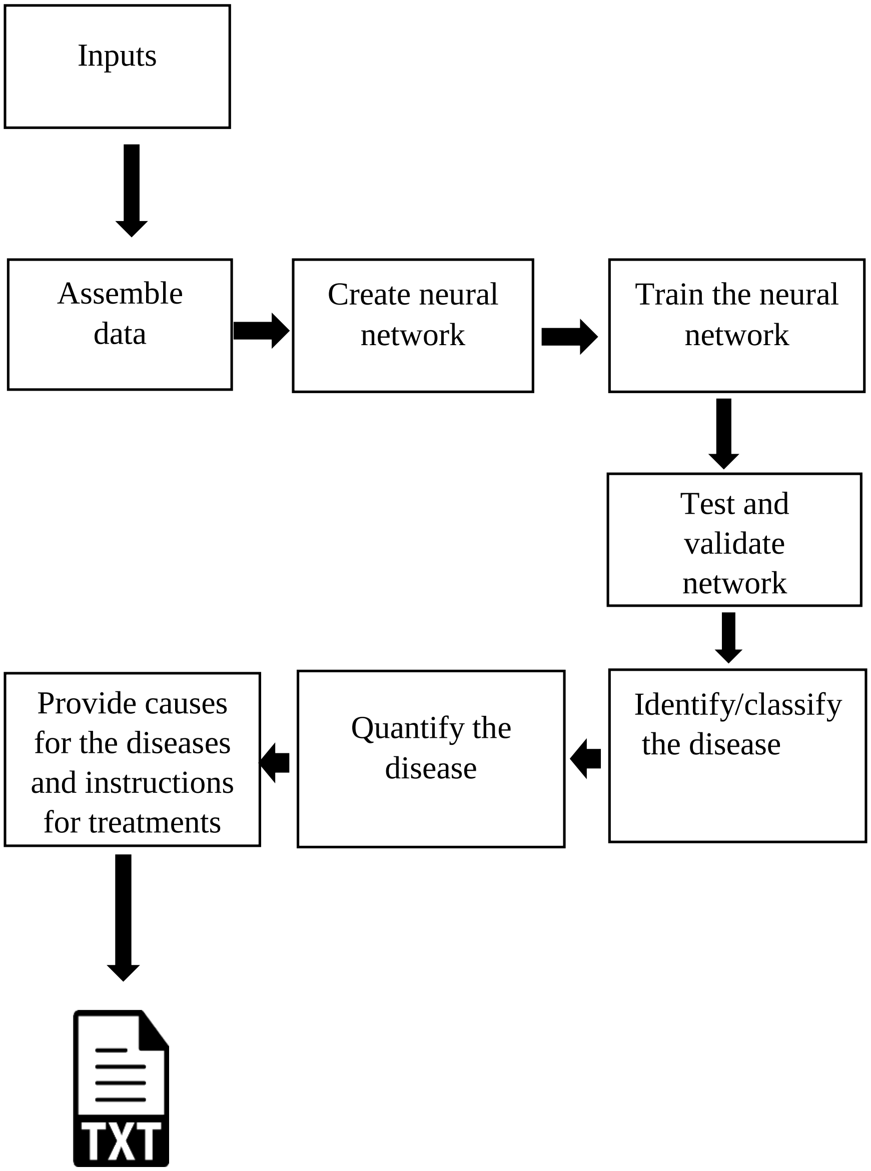

Inputs
Assemble data
Create neural network
Train the neural network
Test and validate network
Identify/classify the disease
Quantify the disease
Provide causes for the diseases and instructions for treatments
TXT

Figure 2: System diagram for disease classification, quantification and treatments

2.1.1 System Interfaces

- Standalone application

A stand-alone application installed in a personal computer or a laptop with Windows 8 or above which user can use anywhere easily.

- Database

A proposed database (MongoDB) is intended to store, retrieve, update and handle the data input by the user and generated by the system itself. This database includes the processed images acquired from the camera and the sensor readings of the environmental factors sunlight, humidity and temperature.

- Integrated Development Environment (IDE)

An IDE that supports Python 2.7 language should be used. PyCharm is the IDE used for the development (coding) process.

- Library for image processing

OpenCV (Open Source Computer Vision) is the library used for image processing. There are bindings available in Python language. This runs on a variety of platforms including Windows.

- Library for machine learning

TensorFlow is the library used for the functions associated with neural networks. It is an open source software and supports Python 2.7.

2.1.2 User Interfaces

The user interfaces of the application and how the user interacts with the system are clearly described in the section 3.1.1 of the document.

2.1.3 Hardware interfaces

- A personal computer or a laptop with windows 8 or above installed.
- Sensor DHT22 - Sensors are needed to get the readings of environmental factors namely temperature and humidity.
  A sensor capable of measuring sunlight is used to get the readings of prevailing sunlight.
- Raspberry Pi 3b - A raspberry pi is necessary to store data taken from sensors
- Fuji film Fine Pix S8300 compact camera - High-quality camera is necessary for image acquisition process for further development of the application.
- SD card – A SD card is used to retrieve data from the raspberry pi and store in the database. This is a non-volatile memory card which can be used in personal computers and laptops.

2.1.4 Software interfaces

- Database

MongoDB is used for database. It is a free and open source cross platform document-oriented database program. Advantages of using MongoDB is that it does not have complex joins and easy to scale. It also has fat in-place updates. MongoDB is accessed through a driver called PyMongo.

- Integrated Development Environment (IDE)

PyCharm is the IDE used for the development process. This the best IDE version used for programming in Python language. This has special features like code analysis, a graphical debugger and an integrated unit tester.

2.1.5 Communication interfaces

Agro desktop application does not need specific communication interfaces as this is a stand-alone application.

2.1.6 Memory Constraints

Minimum 1GB of Virtual Memory (RAM) and 1GB of Memory Space (Secondary Memory) will be required.

2.1.7 Operations

The basic operations related to the disease classification, quantification and providing treatments function are described here. This user friendly application is very helpful because the user can get an idea on every aspect related to a disease through an image uploaded by him. The only task of the user is to then click on a button and the system will display all the information like causes and treatments.

The basic operations related to the disease quantification function is described here. When the user wants to get a rough idea of the severity of the disease, he can upload the image and click on a button and the system will display the severity as a percentage. This is very important for the user to know the amount of fertilizer that should be used for a certain disease. This will cut-off the excess expenditure and save time.

2.1.8 Site adaptation requirements

All the interfaces, instructions and outputs given in the document are available in English language.

2.2 Product functions

Use Case Diagram

The below use case diagram is drawn for the disease classification, quantification and providing treatments and causes for the disease.

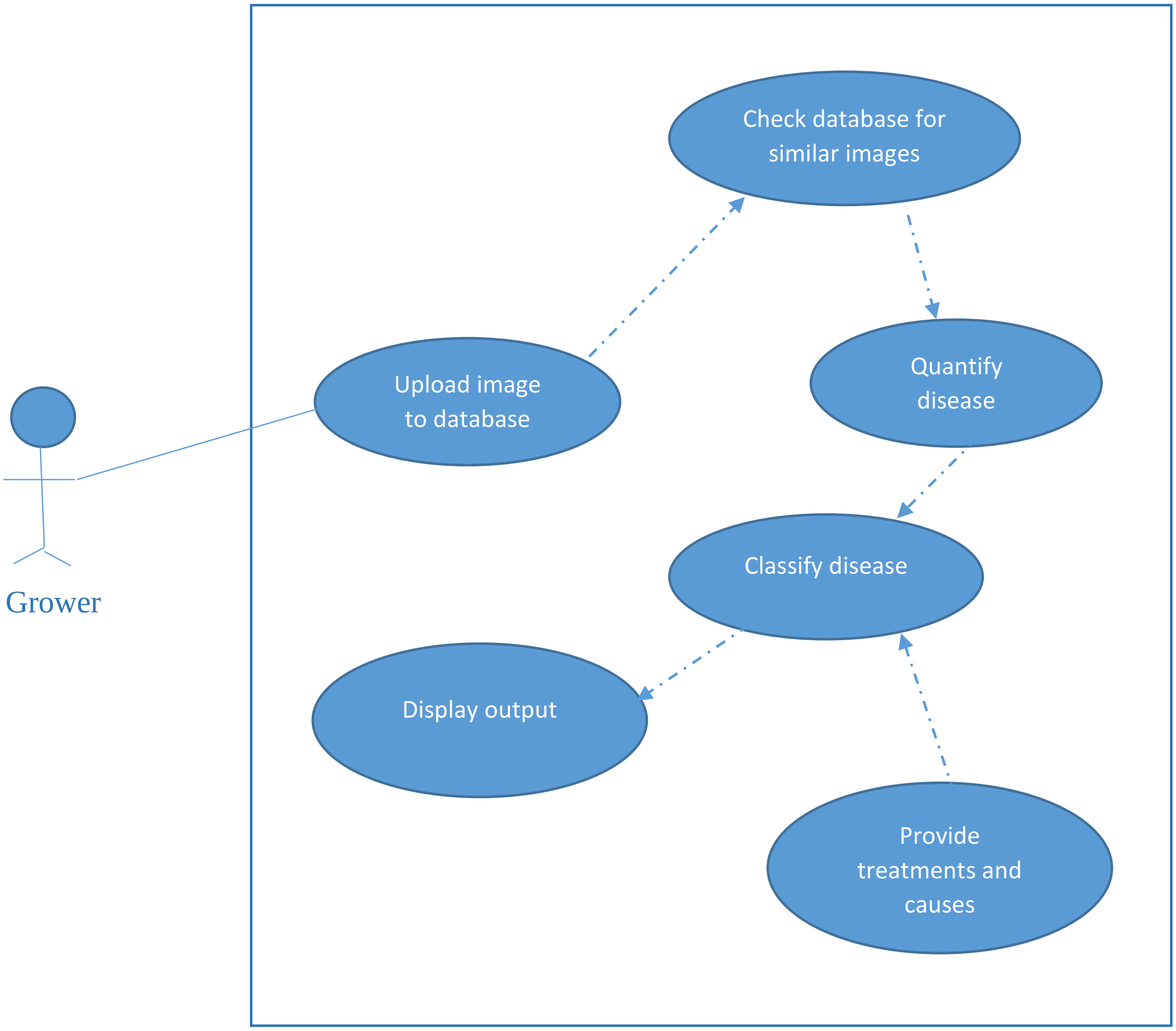


Figure 3: Use case diagram for disease classification

Use Case Scenarios

Use case scenario name	Upload images
Primary actors	Grower
Pre-conditions	System is up and running.
Post-condition	Image is successfully uploaded into the database.
Basic flow of events	1. Use case begins when the user goes to the home page of Agro 2. User clicks on "Classification Results" button in home page. 3.User clicks on Upload Image button 4.User uploads the images 5.User clicks on "Save" button 6. Use case ends when user successfully uploads the images.
Alternatives	5. a. User does not upload images and clicks on "Save" Button, system displays an error message saying "Please upload images"

Table 3: Use case scenario for uploading images to the database

Use case scenario name	Classify disease
Primary actors	Grower
Pre-conditions	System is up and running.
Post-condition	System displays the name of the disease, causes and treatments
Basic flow of events	1. Use case begins when the user goes to the home page of Agro 2. User clicks on “Classification Results” button in home page. 3.User clicks on Upload Image button 4.User uploads the images 5.User clicks on “Save” button 6.User clicks on Result button 7. System checks the database 8. Use case ends when the system displays the name of the disease, causes and treatments of the disease according to the symptom colour.
Alternatives	5. a. User does not upload images and clicks on “Save” button, system generates a message” Error in Uploading Images”. 7. a. Disease with similar symptoms are not available in the database, system generates an error message displaying “Similar disease not found”.

Table 4: Use case scenario for classification of plant diseases

Use case scenario name	Quantify disease
Primary actors	Grower
Pre-conditions	System is up and running.
Post-condition	System displays the severity of the disease as a percentage.
Basic flow of events	1. Use case begins when the user goes to the home page of Agro 2. User clicks on “Classification Results” button in home page. 3.User clicks on Upload Image button 4.User uploads the images 5.User clicks on “Save” button 6. User clicks on severity button
Alternatives	5. a. User does not upload images and clicks on “Save” button, system generates a message” Error in Uploading Images”.

Table 5: Use case scenario for quantification of disease

2.3 User Characteristics

This system is targeted for large scale Anthurium cultivators. Therefore it does not have complicated user interfaces. A person who has a very less amount of IT knowledge will also be able to use this easily as this is very user friendly.

2.4 Constraints

In order to use this application user should have raspberry pi with minimum model 2 and a pc or a laptop with windows 8 or higher with space of 5GB of free space. Users need internet connection to connect with the database and get the access for the images available.

This software will target on Anthurium plant growers specifically. So the complete system will be developed using open source technologies (MongoDB, Open source SDK etc.).

Since the users are not well experienced and lack of knowledge about the field, the application must be developed in a simple manner. All the outputs must be given to the user in a simple form in order to understand them easily.

2.5 Assumptions and Dependencies

All the personal computers or laptops using this application should install windows 8 or higher operating system to successfully run the application efficiently.

2.6 Apportioning Requirements

The requirements described in sections 1 and 2 of this document are referred to as primary specifications; those in section 3 are referred to as requirements (or functional) specifications. The two levels of requirements are intended to be consistent. Inconsistencies are to be logged as defects. In the event that a requirement is stated within both primary and functional specifications, the application will be built from functional specification since it is more detailed.

'Essential requirements' (referred to in section 3) are to be implemented for this version of 1.1. 'Desirable requirements' are to be implemented in this release if possible, but are not committed to by the developers. It is anticipated that they will be part of future release. 'Optional requirements' will be implemented at the discretion of developers.

The interfaces mentioned in section 3 might change, other than that this application will only have slight changes in the development process of this application.

3 System requirements

3.1 External Interface Requirement

3.1.1 User Interfaces

<u>Home Page of Agro</u>

The below interface is the Home Page of the application. It is a common interface to all the group members.

This is a very user friendly interface which any person can clealy undestand. User has to click on the necessary tab to proceed further.

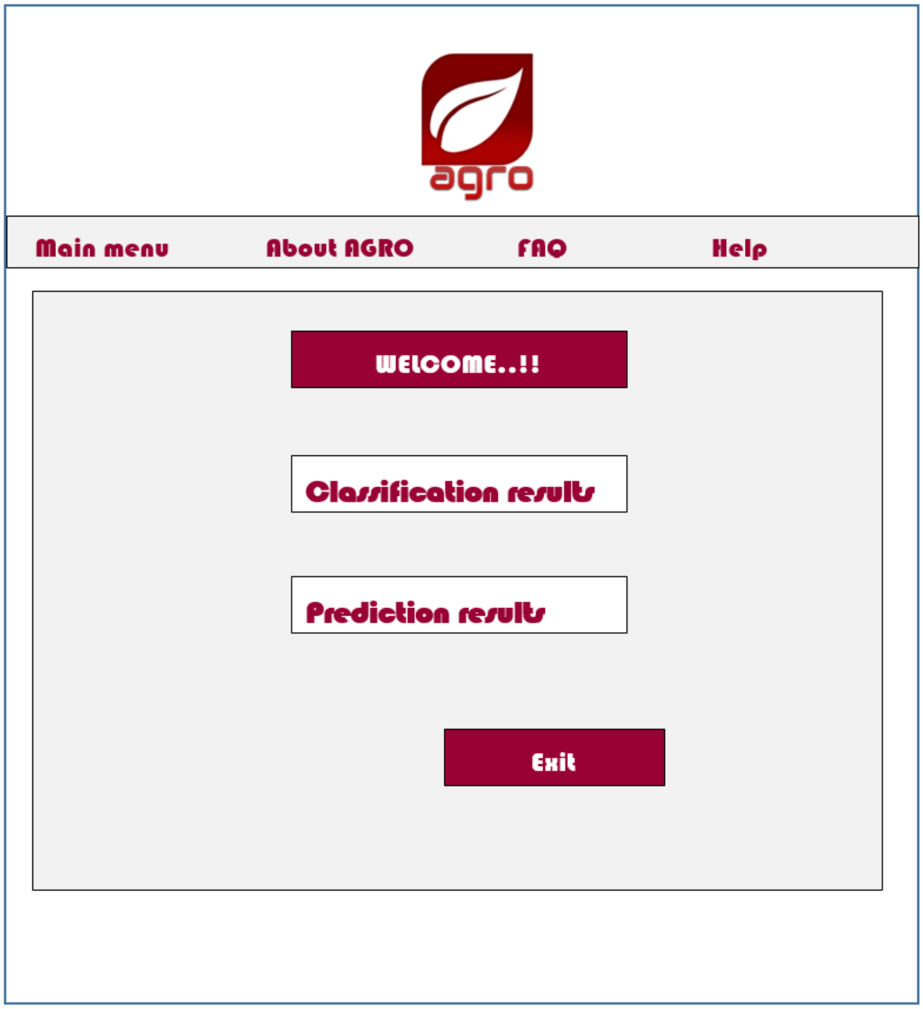
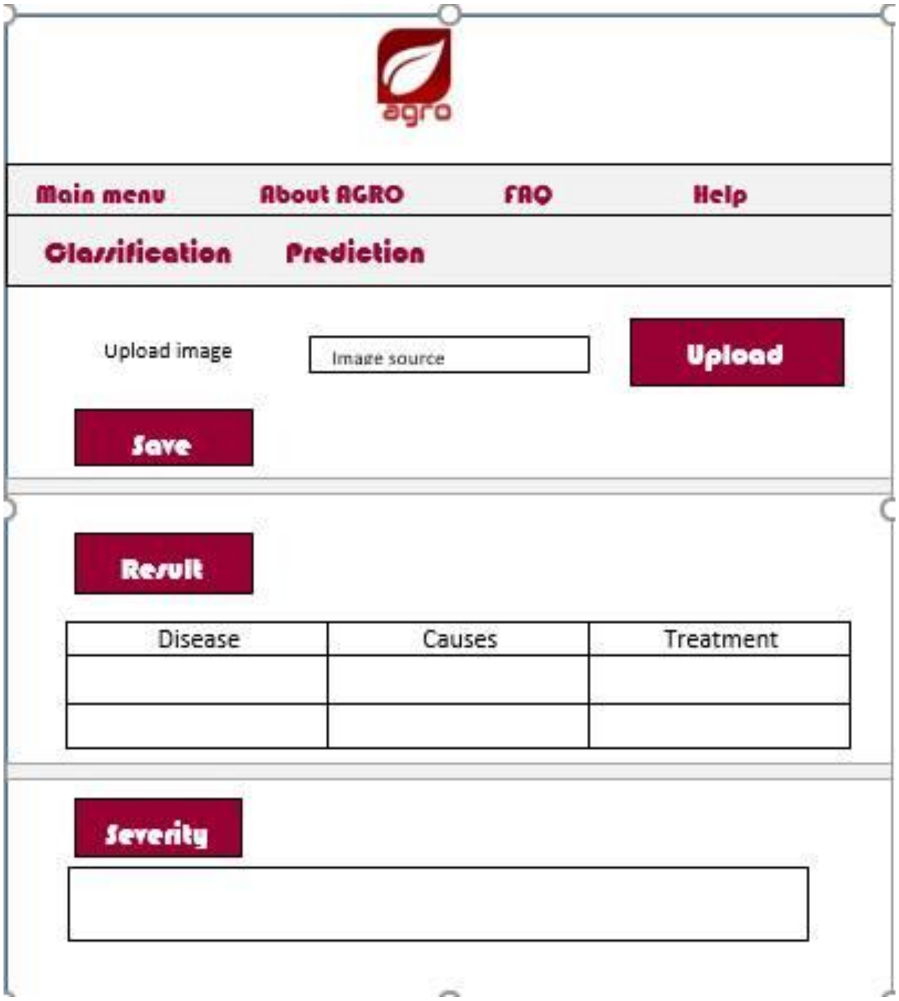


Figure 4: Interface *home Page*



Disease	Causes	Treatment

Figure 5: Interface for disease classification, quantification and providing treatments

The above interface is related to the disease classification, quantification and providing causes for the disease and necessary instructions to minimize the disease. After warning the user that a disease has spread in the plantation, user can get an idea of other aspects related to the disease. When user uploads an image with a disease and clicks on “Result” button, name of the disease, causes and treatments are given. This is mainly done through identifying the colour of the spot. Then when he clicks on “Severity” button, the severity of the disease is shown as a percentage. This interface is very user-friendly and can be easily handled by growers who want to cooperate with technology.

3.1.2 Hardware Interfaces

All the hardware interfaces are described in section 2.1.3 of the document.

3.1.3 Software Interfaces

For the implementation of the desktop application, several open source software interfaces will be used. The following illustrates the software interfaces used for each module of Agro stand- alone application.

For image segmentation and feature extraction process OpenCV image processing libraries are using for the implementation and TensorFlow libraries for artificial neural network implementations will use for the system.

PyCharm IDE for python 2.7 will use for MongoDB database and PyMongo driver is using for the database access process.

3.1.4 Communication Interfaces

As mentioned in section 2.1.5, Agro does not need any specific communication interfaces.

3.2 Classes

The below part of the UML diagram is drawn only for classification, quantification and providing necessary treatments to minimize the disease.

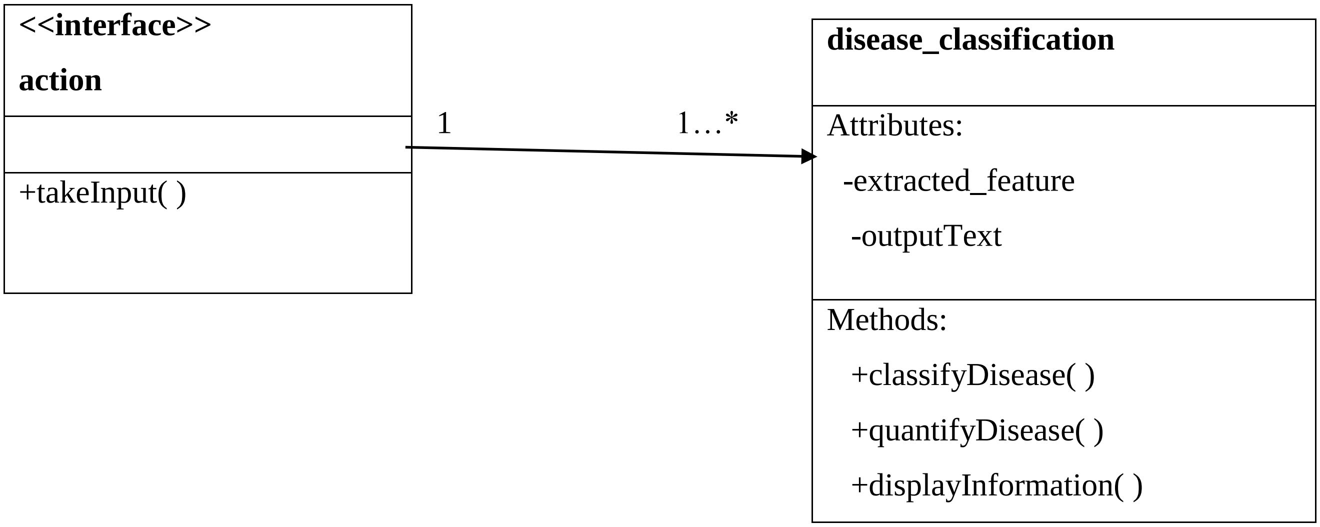


Figure 6: Class diagram

3.3 Performance Requirements

Since the application runs in the background, considerable resources will be allocated, and hence a personal computer or a laptop with significant amount of RAM (at least 2 GB) will be required for smooth usage of the software. Application should not freeze while working.

3.4 Design Constraints

The GUI s are developed attractively and in a user friendly way to attract users to the application at their first sight so that even users with a very less amount of IT knowledge can use the software application easily.

3.5 Software System Attributes

3.5.1 Reliability

Agro stand-alone application will provide its service to users without any issue effecting their satisfaction of the software. It will work according to its specifications. Since users dealing with the application expect something more than doing it manually, the accuracy of the outputs should be high.

3.5.2 Availability

Stand-alone application will work 24x7 and can be used by the users at any time without any issue. Since this is a stand-alone application, no internet connection is needed. Therefore no Wi-Fi related issue will occur.

3.5.3 Maintainability

It will be easy to maintain since it will be developed in the ways it can be extended in future versions and because it is not a complex software. To eliminate issues due to poor coding standards, we are using a common coding standard.

3.5.4 Security

Since the application uses Mongo DB, it provides security to data by creating role based access control mechanism, and can run Mongo DB with dedicated user and provides back up mechanism to your data also.

1.4 References

[1] P. B. G. Athanikar, "Potato Leaf Diseases Detection and Classification System," International Journal of Computer Science and Mobile Computing, vol. Vol.5 (2), pp. 76- 88, February 2016.

[2] N. Rishi and J. S. Gill, "An Overview on Detection and Classification of Plant Diseases in Image Processing," International Journal of Scientific Engineering and Research, vol. 3, no. 5, pp. 114-117, 2015.

[3] B. R. P. Chatterjee, " Leaf Disease Detection using Image Processing Technique," International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, vol. 4, no. 9, pp. 90-93, September 2016.

[4] A. Y. B. M. S. A. H. A. A. S. M. F. Z.B. Husin, "Feasibility Study on Plant Chili Disease Detection Using Image Processing Techniques," in Sch. of Comput. & Commun. Eng., Univ. Malaysia Perlis, Kuala Perlis, Malaysia, Kota Kinabalu, 2012.

[5] J.Barbedo, "Digital image processing techniques for detecting, quantifying and classifying plant diseases," in https://springerplus.springeropen.com/articles/10.1186/2193-1801-2-660, 2013. [6] D. S. A. Verma, "Content Based Image Retrieval Using Color, Texture and Shape Features," International Journal of Advanced Research in Computer Science and Software Engineering Research Paper, pp. 780-784, 2007.

4 Appendices

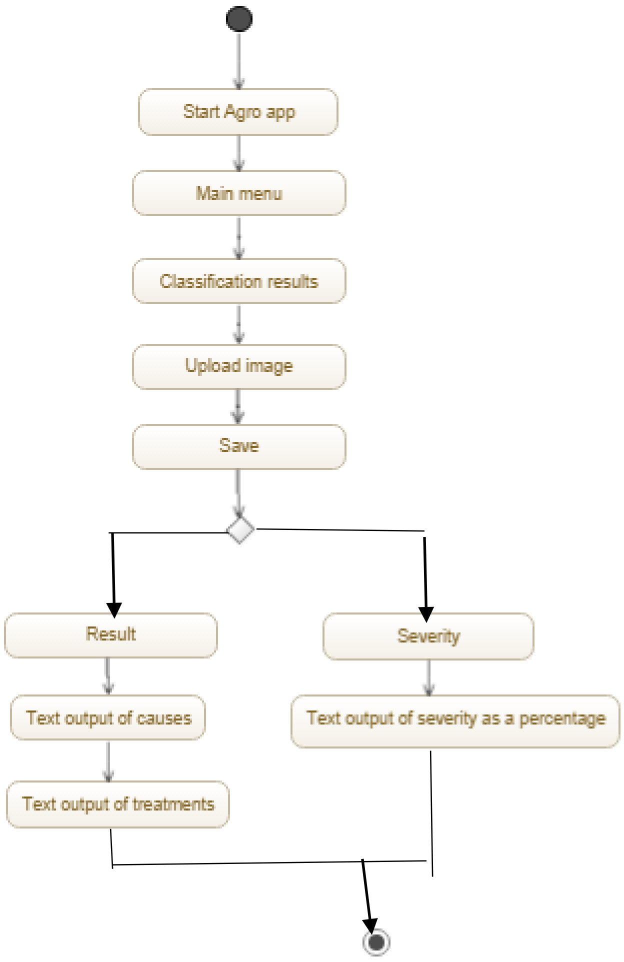


Figure 7: Activity diagram