Software Requirement Specification Document

Smart Detection of Plants Disease

Amgad Atef, Mohanad Ayman, Mostafa Medhat

Dr. Tarek Gaber, Eng. Silvia Soliman

February 19, 2017

1 Introduction

1.1 Purpose of this document

The purpose of this document is to give a detailed description of the requirements for the Smart Detection of Plants Diseases project. It will illustrate the purpose and features of the system as well as complete declaration for the development of system. It will also explain system constraint sunder which it must operate and interface.This document is primarily a reference for developing the first version of the system for the development team.

1.2 Scope of this document

This project is a mobile application with a web service to let the farmers detect the diseases of the plants in early stages by taking pictures of the leaves and send it to the server in order to classify the disease. Three developers will work on this system. One developer will work on the pre-processing, the second will develop the feature extraction and the third is responsible for the classification phase which is the main core of the system. The targeted users are the Egyptian farmers. This is why the application should be user friendly and easy to use as much as possible. The early versions will not work on all plants while the database is not populated with many types of plants. The result processing time will depend on the computational power of the server.

1.3 Overview

Pre-processing is the first phase in the system and it should extract the leaf from the image to decrease the time of processing in the classier. And background of each image will be removed using background subtraction technique. Then

feature extraction will be applied in the server, the purpose of feature extraction is to reduce the original dataset by measuring certain features or properties of each image such as texture, color and shape. In order to recognize and identify healthy and infected leaf, measure several numbers of features in acquired image, to be later use for classification. The final step of this system is the classication phase, neural networks algorithm will be applied for classifying the plant leaf image to any of the following states, healthy or infected. The inputs of this stage are training dataset; the outputs will be the decision that determine type of input image (healthy of infected). The final product is a mobile app that will take picture of the leaf and then will send it to the web server which will do all the mentioned processing.

1.4 Business Context

This project helps organizations, individuals and the government. The system can boost the economy by increasing the national income. Farmers wait long time for experts to come in order to examine their plants. And they pay a lot of money for them in order to prescribe pesticides or other solutions. Also, the government and farms owners spend so much money on chemicals to cure the infected plants and to save their crops. Saving this cost is the chance for this system to spread. Beside saving the pesticides money, producing better crops in terms of quality will give them the chance to be exported. All these advantages have a direct positive impact on the economy. The application will be used by specific category of people who would be interested in some sort of advertising that is related to agriculture, which can bring sponsors from the same field.

2 General Description

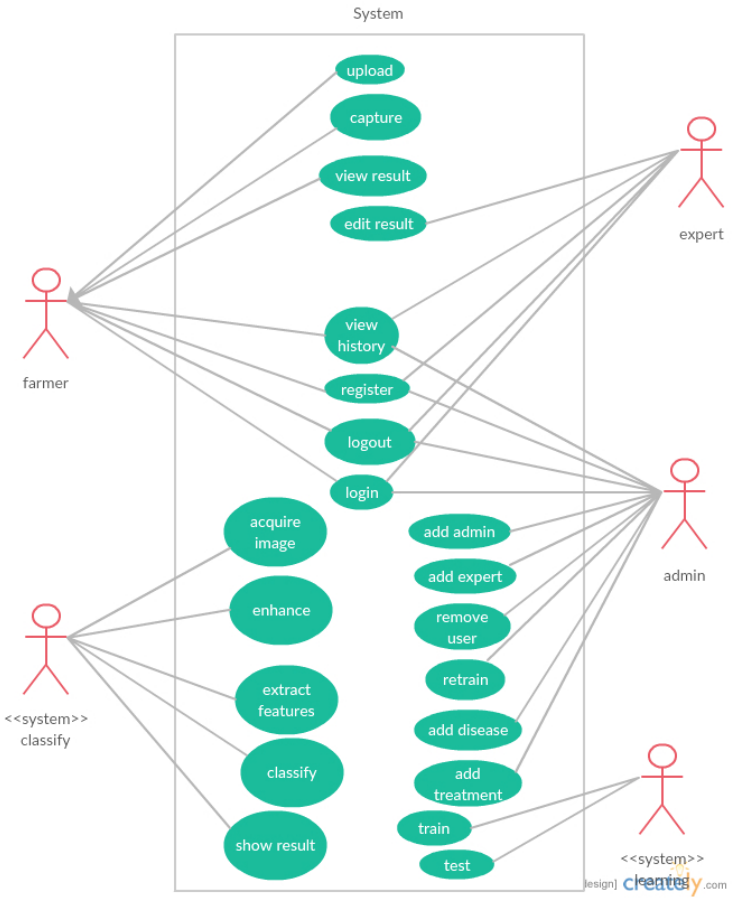

Figure 1: Use Case

2.1 Product Functions

Google sign in API: using google sign in method for easier login.
Capture and Upload: the user should open the camera and take a picture of the infected plant and upload it from the app to get results.
Preprocessing: the system should receive the picture farmer in mobile application and execute some image processing techniques to extract the leaf and the infected area only then send the vector to the server.
Feature extraction: then the feature extraction phase occurs to output a vector of values and send it to the classifier.

Classification: the classifier should then determine what disease the plant has and return it to the user.

2.2 Similar System Information

Ethiopian Coee Plant Diseases[1]: In this paper the researchers focus on the Ethiopian coee plant which contribute with 20% of Ethiopia national income. In this Experimental simulation, the combination of RBF and SOM has a better performance than the other classiers. But when we see the training time of the combination of RBF and SOM, it takes longer time in training. The performance of combination of RBF (Radial basis function) and SOM (Self organizing map) is 90.07%. The summary result of KNN, ANN, Naive and a combination of RBF and SOM are 58.16 %, 79.04%, 53.47%, 90.07% respectively. It shows the result of multiple classiers used and the drawbacks of the highest of them.
Detection and Classication of Diseases of Grape Plant[2]: Plant diseases cause major economic and production losses as well as degradation in both quantity and quality of agricultural production. The critical issue here is to monitor the health of the plants and detection of the respective diseases on a large-scale eld. The technique proposed for identication of plant disease through the leaf texture analysis and pattern recognition. The proposed approach avails advice of agricultural experts easily to farmers with the accuracy of 96.6%. in this work by using segmented leaf image and analyzing it through high pass lter to detect the diseased using SVM classier gives a very high accuracy.

2.3 User Characteristics

People: farmers, agriculture experts, system administrators
Age: +18
Gender: any gender
Illiteracy: farmer might be illiterate
Computer ignorant: farmer might be ignorant about computers
Mobile phones awareness: should has the basic knowledge of starting an app on android system

2.4 User Problem Statement

There are many types of agricultural pests in the planted areas in Egypt which highly affect the crops in a negative way, whether in quality or quantity. Finding a solution for this problem will have a positive impact on our economy and will increase the national income. Farmers wait long time for experts to come to examine their plants. And they pay a lot of money for them in order to prescribe pesticides or other solutions for their plants. Our project aims to help the farmers and their crops by detecting the infected plants in their early stages. Also, reducing the usage of pesticides which helps the environment and the national income.

2.5 User Objectives

The farmers want to identify the disease with minimum cost and time.

2.6 General Constraints

8mp Camera, android phone, network connection, online database and web service.

3 Functional Requirements

Smart Detection of Plant Diseases / SRS / Functional Requirment/ P.01	
Code	P.01
Name	Login
Criticallty	Medium
Input	Google account
Output	Login successful / Login Failed
Description	The farmers should login using google account to save the results. The applicaton will run in android mobiles, so it must have google account.
Priority	6/10
Preconditions	The mobile shall be connected to the internet.
Post-conditions	Login Successful without failure.

Smart Detection of Plant Diseases / SRS / Functional Requirment/ P.02	
Code	P.02
Name	upload photo
Criticallty	High
Input	Photo
Description	The application will get access to the camera of the mobile and let the farmer take a picture.
Priority	10/10
Preconditions	The image to load has an image format supported by android software.
Post-conditions	The file uploaded correctly without errors.
Dependencies	Image type: jpg, png.

Smart Detection of Plant Diseases / SRS / Functional Requirment/ P.03	
Code	P.03
Name	Plant identification
Criticallty	Medium
Input	Vector
Output	Kind of plant.
Description	The system will be able to identify the type of plant
Priority	6/10
Preconditions	The image is for plant leaf and clear with suitable resolution.
Post-conditions	The image will be classified.

Smart Detection of Plant Diseases / SRS / Functional Requirment/ P.04	
Code	P.04
Name	Infection identification
Criticallty	High
Input	Vector
Output	Infected/ Not infected.
Description	The system will classify if the plant is infected or not.
Priority	10/10
Preconditions	The image is for plant leaf and clear with suitable resolution.
Post-conditions	The image will be classified and it will pass to disease identification.

Smart Detection of Plant Diseases / SRS / Functional Requirment/ P.05	
5- Code	P.05
Name	Disease identification
Criticallty	High
Input	Vector
Output	Type of disease.
Description	The system will be able to identify the type of disease in the plant.
Priority	10/10
Preconditions	The image is for plant leaf and clear with suitable resolution.
Post-conditions	The image will be classified and it will pass to treatment method.

Smart Detection of Plant Diseases / SRS / Functional Requirment/ P.06	
Code	P.06
Name	Check Results
Criticallty	High
Input	farmer ID
Output hlineine Description	Result of his photo The farmer can open the application and see the result of his photo.
Priority	10/10
Preconditions	The user should be uploaded an image to be able to see results.
Post-conditions	The classification should work correctly and give suitable results.

Smart Detection of Plant Diseases / SRS / Functional Requirment/ P.07	
7- Code Name	P.07 Mobile Application
Criticallty	High
Description	A user should be able to download the mobile application through either an application store or similar service on the mobile phone. The application should be free to download.
Priority	10/10

4 Interface Requirements

4.1 User Interfaces

Figure 2: Wireframes

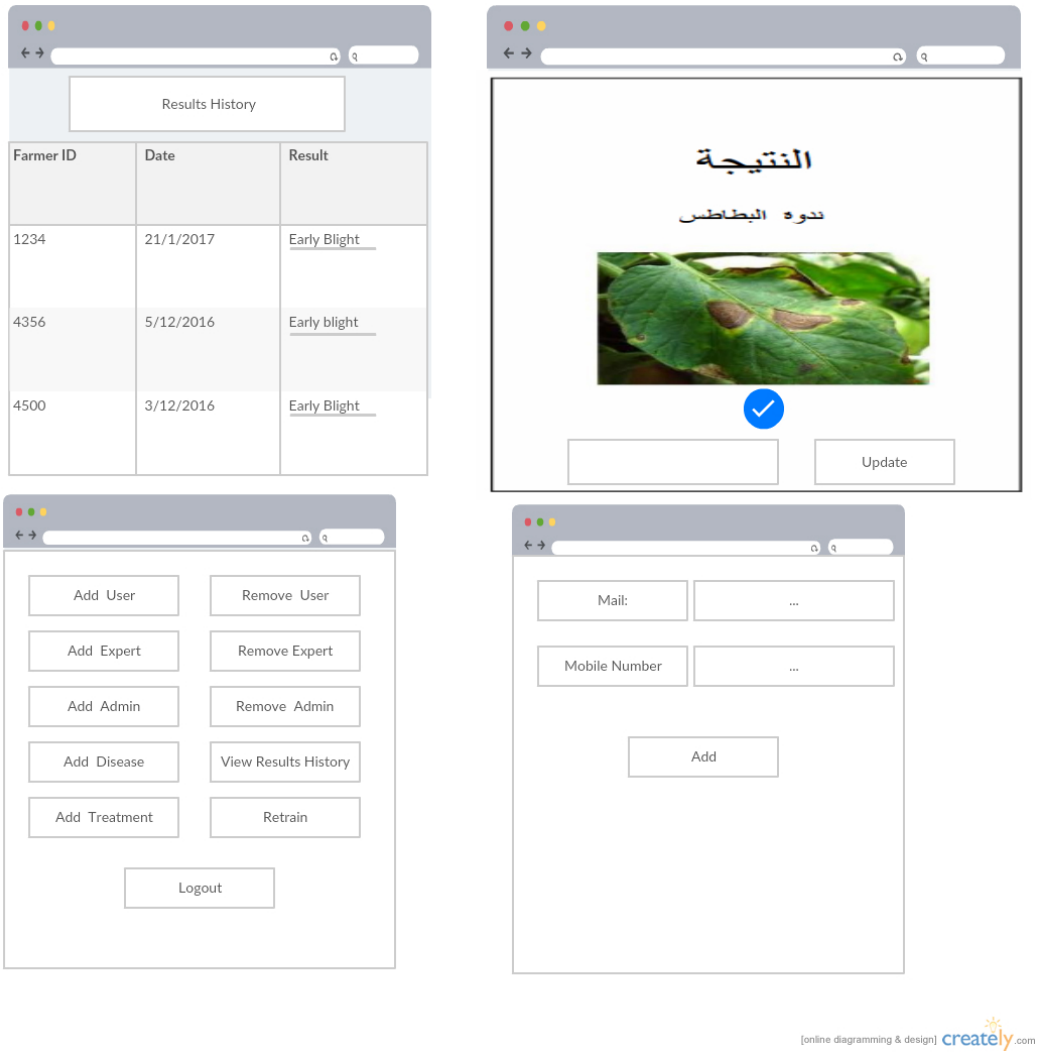

Figure 3: Wireframes

5 Performance Requirements

The classification phase will be processed on the server due to the limited computational power of mobile phones. Another reason, is that the database will be on the server. The first two phases, pre-processing and feature extraction can be done on the phone. The application should not take more than 3MB from phone storage. While the database on the server will grow with time.

6 Design Constraints

6.1 Hardware Limitations

1- The mobile camera should has at least 8mp resolution in order to output good pictures.
2- The mobile application should be connected to the Internet.
3- The classification speed will depend on the server specifications.
4- Minimum server specifications: 2 GHz processor, 8GB ram, C++ compiling.

7 Other non-functional attributes

7.1 Security

The data of the farmer will be encrypted.

7.2 Reliability

The system shall never crash or hang, it might be reliable by more than 90%

7.3 Maintainability

Code shall be fully documented. Each function shall be commented with its conditions. The project will be designed using MVC, single tone design patterns. It will be easy to maintain the system.

7.4 Extensibility

The system can be extended for future upgrades like adding new type of plants.

7.5 Scalability

The system shall be scalable and can be extended to add more types of plants and diseases.

7.6 Portability

The software shall be designed to run on android mobile phones.

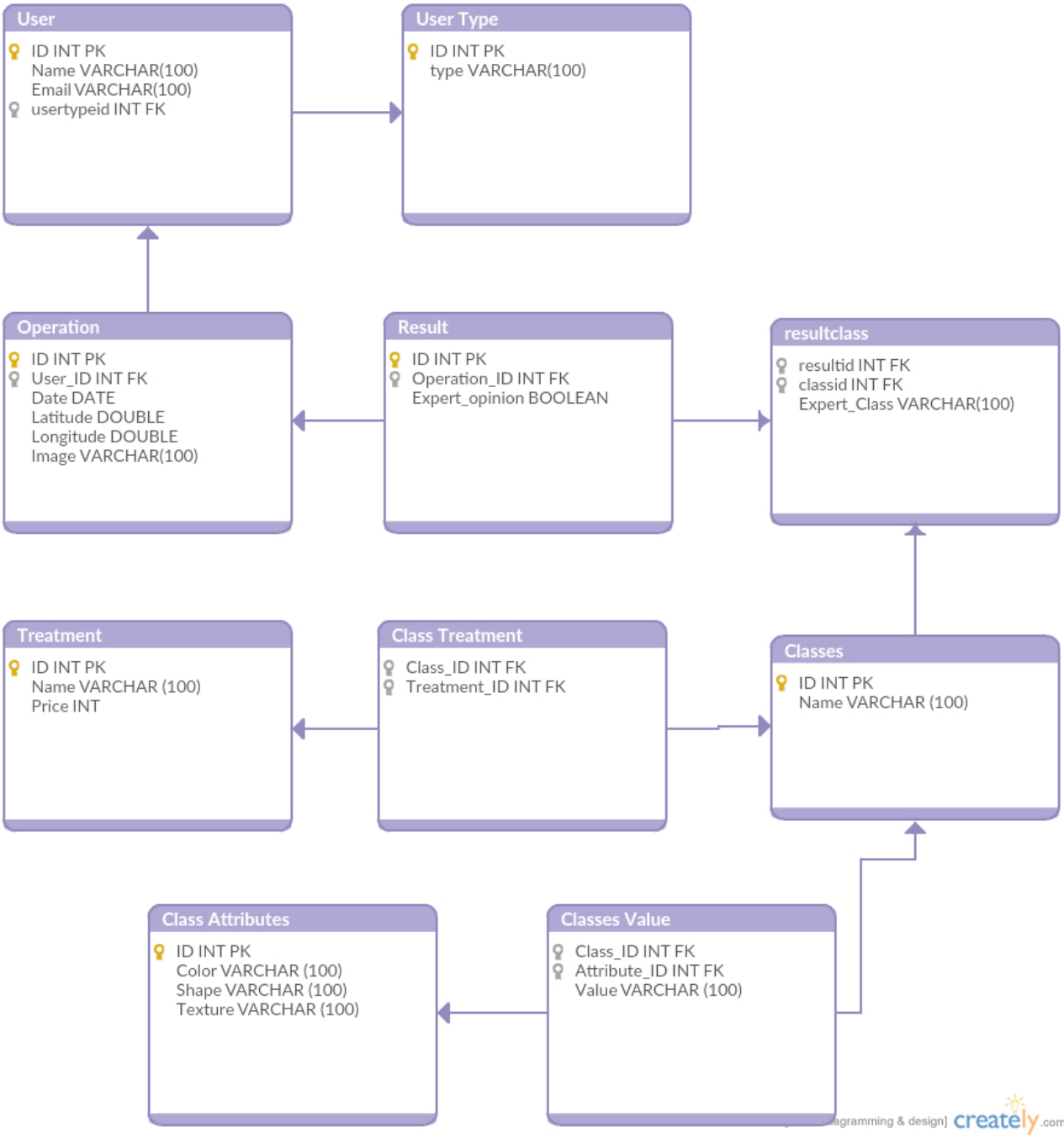

Figure 4: Database Diagram

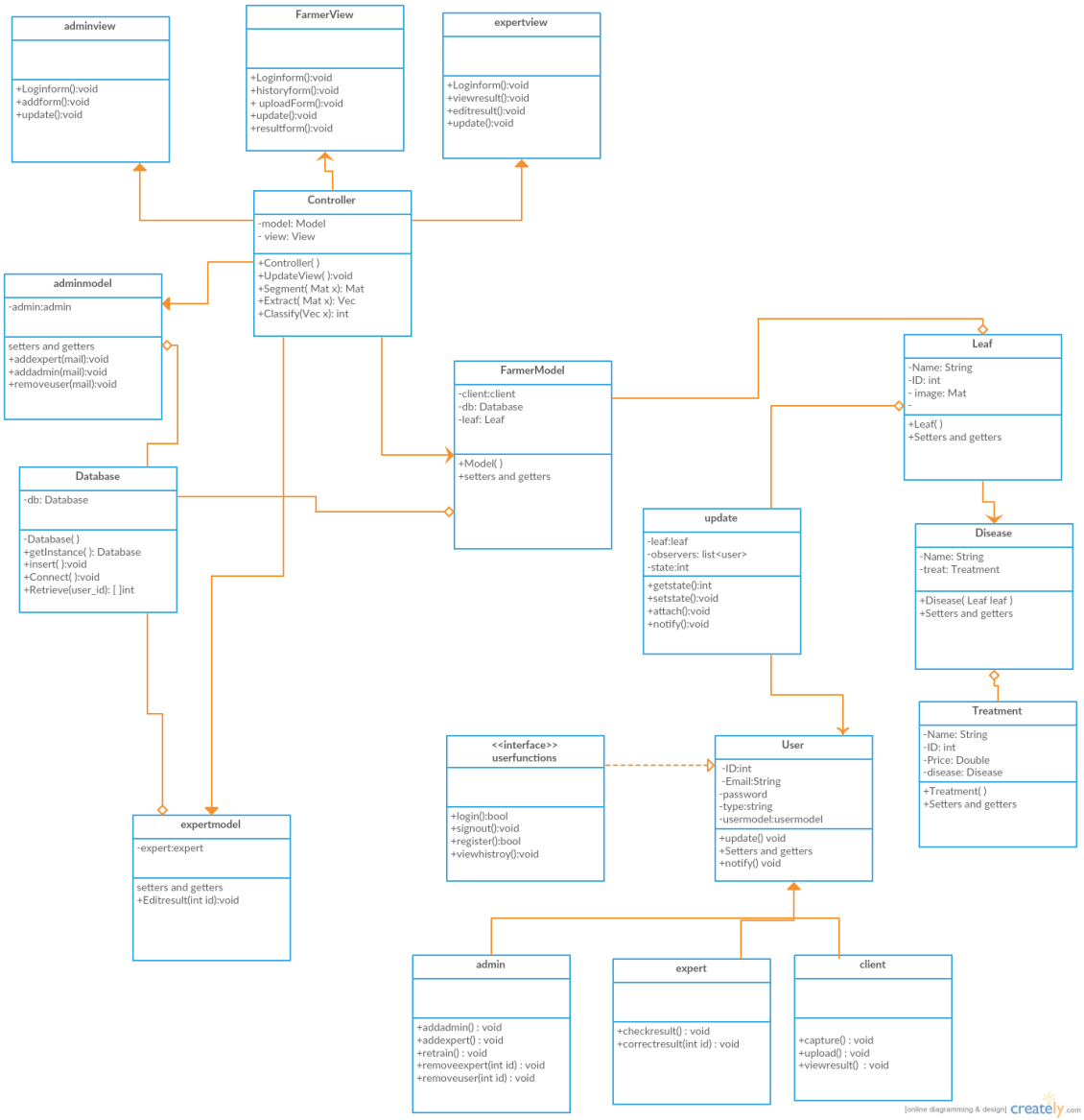

Figure 5: Class Diagram

8 Preliminary Object-Oriented Domain Analysis

class name:adminview
type : concrete
list super class : N/A
list of sub classes : N/A
purpose :display the forms and results concerning the admin
composition : associate with controller
attributes : N/A
operations : log in form , add admin form , add expert form ,remove user form,update view

Class name : farmerview
type : concrete
list super class : N/A
list of sub classes : N/A
purpose :display the forms and results concerning the farmer
composition : associate with controller
attributes : N/A
operations : log in form, history list , upload , result,update view

Class name : expertview
type : concrete
list super class : N/A
list of sub classes : N/A
purpose :display the forms and results concerning the expert
composition : associate with controller
attributes : N/A
operations : log in form, view result,edit result, update view

Class name : controller
type : concrete
list super class : N/A
list of sub classes : N/A
purpose : connect the view classes with the models to achieve the mvc design pattern
composition : associate with farmerview,expertview,adminview,adminmodel,expertmodel,usermodel
attributes : object of each model and view as needed
operations : constractor,updateview,segment,extract,classify

Class name : adminmodel
type : concrete
list super class : N/A
list of sub classes : N/A

purpose : contain all the operations the admin could perform
composition : associate with controller, aggregate database
attributes : object of admin, object of database connection
operations : setters and getters,addexpert,add admin,remove user

Class name : farmermodel
type : concrete
list super class : N/A
list of sub classes : N/A
purpose :contain all the operations the farmer could perform
composition : associate with controller, aggregate database
attributes : object of client, object of database connection, object of leaf
operations : setters and getters,constructor

Class name : expertmodel
type : concrete
list super class : N/A
list of sub classes : N/A
purpose : contain all the operations the expert could perform
composition : associate with controller, aggregate database
attributes : object of expert , object of database connection
operations : setters and getters, edit result

Class name : database
type : concrete
list super class : N/A
list of sub classes : N/A
purpose : make a single connection to database and execute all its operations
composition : aggregated to farmermodel,expertmodel,adminmodel
attributes : database object
operations : constructor,getinstance,insert,connect,retrieve

Class name : update
type : concrete
list super class : N/A
list of sub classes : N/A
purpose : notify the users of the system
composition : aggregated to leaf, associated with user
attributes : object of leaf, list of users, state
operations : setters and getters, attach, notify

Class name : leaf
type : concrete
list super class : N/A
list of sub classes : N/A
purpose : describes the leaf

composition : associated with disease, aggregated with update
attributes : name, id, image
operations : constructor, setters and getters

Class name : disease
type : concrete
list super class : N/A
list of sub classes : N/A
purpose : describes the diseases
composition : associated with leaf, aggregated with treatment
attributes : name, object of disease
operations : constructor, setters and getters

Class name : treatment
type : concrete
list super class : N/A
list of sub classes : N/A
purpose : describes the treatment
composition : aggregated with disease
attributes : name, id, price, object of disease
operations : constructor, setters and getters

Class name : user
type : concrete
list super class : N/A
list of sub classes : expert , admin , client
purpose : describes user functionality.
composition : associated with update
attributes : id, email, password , type, object from usermodel
operations : notify, update, setters and getters

Class name : userfunctions
type : interface
list super class : N/A
list of sub classes : N/A
purpose : interface to some functions used differently by each type of user
composition : implemented in user
attributes : N/A
operations : log in,sign out,register,viewhistory

Class name : admin
type : concrete
list super class : user
list of sub classes : N/A
purpose :describes admin functionality.
composition : inherited from user

operations : constructor, addadmin, addexpert, retrain, removeexpert, removeuser
Class name : expert
type : concrete
list super class : user
list of sub classes : N/A
purpose : describes expert functionality.
composition : inherited from user
operations : constructor, checkresult, coorectresult

Class name : client
type : concrete
list super class : user
list of sub classes : N/A
purpose : dsecribes client functionality.
composition : inherited from user
operations : constructor, capture, viewresult

9 Operational Scenarios

1- The user downloads the application.
2- The user starts the application from his phone.
2.1- Click the capture button.
2.1.1- The camera opens and the user capture an image.
2.1.2- The user can choose between taking the image again or send it to the classification phase.
2.1.3- The user receives the result.
2.2- Click the history button.
2.2.1- View previous results.

10 Preliminary Schedule

Task Name	Du...	Start	Finish
Decide the algorithms for each phase	11d	11/01/16	11/15/16
implementing the algorithms	67d	11/15/16	02/15/17
Testing the implemented system and check the merged cod	11d	02/15/17	03/01/17
Start training the system with ready images from the databa	23d	03/01/17	03/31/17
Test the accuracy of the classification	1d	03/31/17	04/31/17
Release the first version	3d	04/30/17	05/02/17

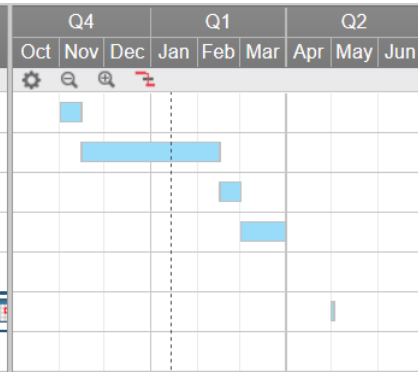

Figure 6: Timeline

11 Appendices

11.1 Definitions, Acronyms, Abbreviations

MVC: Model view control design pattern.
API: application programming interface.
KNN: K-nearest neighbour algorithm
RBF: Radial basis function
SOM: Self organizing map

12 References

[1] Abrham Debasu Mengistu1*, Dagnachew Melesew Alemayehu2 and Seffi Gebeyehu Mengistu3 1Bahir Dar University, Bahir Dar Institute of Technology, Bahir Dar, Ethiopia Ethiopian Coffee Plant Diseases Recognition Based on Imaging and Machine Learning Techniques
[2] Harshal Waghmare. Student, Dept. of Electronics and Telecommunication Engineering, Vishwakarma Institute of Information Technology, Pune, India. Radha Kokare. Student, Dept. of Electronics and Telecommunication Engineering, Vishwakarma Institute of Information Technology, Pune, India. Detection and Classification of Diseases of Grape Plant Using Opposite Colour Local Binary Pattern Feature and Machine Learning for Automated Decision Support System