**IMPLEMENTING DIGITAL IMAGE PROCESSING IN A MOBILE BASED SYSTEM FOR DETECTING CROP DISEASES USING SCANNERS**

A Proposal for a project to be conducted in partial fulfilment of the Requirement for the Bachelor of Science in Information Technology Department of Information

Technology.

JOMO KENYATTA UNIVERSITY OF AGRICULTURE AND TECHNOLOGY

NAME: MARY GLADYS NJOKI WAWERU

REG NO: BIT-001-7289/2014

UNIT CODE: BIT 2303

UNIT NAME: PROJECT (PROJECT DEFINATION AND RESEARCH)

SUPERVISOR: PIUS THUKU

2018

**DECLARATION**

Declaration by the student

I MARY GLADYS NJOKI WAWERU. Declare that this research project is my original work and has not been presented in any institution for the awarding of the same course or any other.

Student’s Name Signature Date

MARY GLADYS NJOKI WAWERU

**CERTIFICATION**

This is to certify that the above-named student has so far carried out the proposal work and submitted for examination with my approval as University Supervisor.

Signed Date

PIUS THUKU,

Project supervisor.

**ABSTRACT**

Crop diseases is a major problem facing small holder farmers crops that lead in the overall decline of crop production in the country lowering the Gross Domestic Product (GDP) focused on agriculture. According to the survey taken on the farmers and relevant stakeholders, they resulted to call out for innovative ideas and skills involving technology to help to succumb the problems and the proficiency extent at which smartphones are being developed and massive scaling to outreach a huge percentage of the people can be used to bring out change or solutions relating to agriculture. Therefore, extensive research on digital image processing techniques that incorporates use of artificial intelligence can be applied in mobile based systems to detect crop diseases in a fast and efficient manner and therefore corrective preventive and curating measures ca be taking to succumb the crop diseases.

**Table of Contents**

[CHAPTER ONE 5](#_Toc5666037)

[1.1 Background study 6](#_Toc5666038)

[1.2 Problem statement 6](#_Toc5666039)

[1.3 Proposed solution 7](#_Toc5666040)

[1.4 Objectives 7](#_Toc5666041)

[1.4.1 General Objective 7](#_Toc5666042)

[1.4.2 Specific objectives 7](#_Toc5666043)

[1.5 Research Questions 7](#_Toc5666044)

[1.6 Problem justification 7](#_Toc5666045)

[1.7 Scope of the study 8](#_Toc5666046)

[1.8 Limitations 8](#_Toc5666047)

[1.9 System development methodology 9](#_Toc5666048)

[1.9.1 Principles of DSDM 10](#_Toc5666049)

[1.10 Data and Information Collection 11](#_Toc5666050)

[1.10.1 Primary methods 11](#_Toc5666051)

[Interviews 11](#_Toc5666052)

[2.0 LITERATURE REVIEW 12](#_Toc5666053)

[2.1Introduction 12](#_Toc5666054)

[2.2.1 Digital image processing techniques in detecting crop diseases. 12](#_Toc5666055)

[STEP 2: Quantification 14](#_Toc5666056)

[STEP 3: CLASSIFICATION 15](#_Toc5666057)

[CHAPTER THREE 20](#_Toc5666058)

[3.1. Introduction 20](#_Toc5666059)

[3.2 Systems Development methodology 20](#_Toc5666060)

[3.3 Feasibility Study 20](#_Toc5666061)

[3.3.1 Economic Feasibility 20](#_Toc5666062)

[Economic Feasibility Report 20](#_Toc5666063)

[3.3.2Operation Feasibility 21](#_Toc5666064)

[3.3.3Technical Feasibility 21](#_Toc5666065)

[3.4 Requirements Elicitation 21](#_Toc5666066)

[3.5 Data and System Analysis 22](#_Toc5666067)

[Analysis of the findings 22](#_Toc5666068)

[3.5.1 Report of Questionnaires 23](#_Toc5666069)

[3.6 Conclusion 25](#_Toc5666070)

[3.6.1 Functional Requirements 25](#_Toc5666071)

[3.6.2 Non-functional Requirements 25](#_Toc5666072)

[3.7.2 Database Design 30](#_Toc5666073)

[Un-normalized Tables of Data 30](#_Toc5666074)

[CHAPTER FOUR 35](#_Toc5666075)

[4.1 Introduction 35](#_Toc5666076)

[4.2 System code generation 35](#_Toc5666077)

[4.3 Testing 35](#_Toc5666078)

[4.3.1 Test plan 35](#_Toc5666079)

[4.3.2 Test Results 35](#_Toc5666080)

[GUI testing 35](#_Toc5666081)

[Test case 2 36](#_Toc5666082)

[Test case 3 36](#_Toc5666083)

[Unit Testing 37](#_Toc5666084)

[Module Testing 39](#_Toc5666085)

[Integration Testing 39](#_Toc5666086)

[Functional Testing 40](#_Toc5666087)

[4.4 Conclusion 40](#_Toc5666088)

[4.5 Limitations 40](#_Toc5666089)

[4.6 Recommendations 40](#_Toc5666090)

[4.8.1Project requirements 43](#_Toc5666091)

[4.8.2Project budget 44](#_Toc5666092)

[4.8.4 Gantt chart 46](#_Toc5666093)

**List of Tables**

[Table 1 1Analysis of Findings 23](#_Toc5664878)

[Table 1 2 Admin Table 32](#_Toc5664879)

[Table 1 3 Farmer Table 32](#_Toc5664880)

[Table 1 4 Plant Table 32](#_Toc5664881)

[Table 1 5 Results Table 33](#_Toc5664882)

[Table 1 6 Disease Table 33](#_Toc5664883)

[Table 1 7 Acceptance testing 39](#_Toc5664884)

[Table 1 8 Module testing 40](#_Toc5664885)

[Table 1 9 Integration testing 40](#_Toc5664886)

[Table 1 10 Functional testing 41](#_Toc5664887)

[Table 1 11 Project Budget 44](#_Toc5664888)

[Table 1 12 Project Schedule 45](#_Toc5664889)

**Table of figures**

[figure1. 1 Report of Questionnaires 24](#_Toc5665628)

[figure1. 2 sampled data from the stakeholders 25](#_Toc5665629)

[figure1. 3 opinion on the effectiveness of the system 26](#_Toc5665630)

[figure1. 4 flow chart diagram 29](#_Toc5665631)

[figure1. 5 Use Case Diagram 29](#_Toc5665632)

[figure1. 6 Sequence Diagram 31](#_Toc5665633)

[figure1. 7 Entity Relationship Diagram 34](#_Toc5665634)

[figure1. 8 Login Page Design 34](#_Toc5665635)

[figure1. 9 Home Page Design 35](#_Toc5665636)

[figure1. 10 Results Page Design 35](#_Toc5665637)

[figure1. 11 Gantt Chart 46](#_Toc5665638)

[figure1. 12 Login Page Test 51](#_Toc5665639)

[figure1. 13 Register Page Test 52](#_Toc5665640)

[figure1. 14 Account Creation Test 53](#_Toc5665641)

[figure1. 15 Home Page Test 54](#_Toc5665642)

[figure1. 16 Results Page Test 55](#_Toc5665643)

# **CHAPTER ONE**

**1.0 Proposal**

# **1.1 Background study**

Bayer East Africa, headquartered in Nairobi, Kenya is a subsidiary of Bayer AG, a global innovation enterprise with core competencies in the Life Science fields of agriculture (crop science) and health care (pharmaceuticals and consumer health). Bayer AG has a base and registration in Germany, with the global headquarters in Leverkusen. As an innovation company, it sets trends in research-intensive areas and its products and services are designed to benefit people and improve their quality life and in doing this it tries to establish a niche in small holder farmers levels crop production and tried to establish some of the root causes that may have resulted from this decline which results into food insecurity.

The achievement of food security is a key objective in the agriculture sector in Kenya as it directly contributes 24% of the Gross Domestic Product (GDP). Food security is defined as “a situation in which all people, at all times, have physical, social and economic access to sufficient, safe and nutritious food which meets their dietary needs and food preferences for an active and healthy life” as by the Kenya Food Security Steering Group in 2008.

The upcoming trends in technology has also been incorporated in agriculture where the use of smartphones is on the rise and innovation technology that can be applied in them in line to solve problems facing the agriculture sector plays a huge role

# **Problem statement**

The achievement of food security is a key objective in the agriculture sector in Kenya as it directly contributes 24% of the Gross Domestic Product (GDP). Food security is defined as “a situation in which all people, at all times, have physical, social and economic access to sufficient, safe and nutritious food which meets their dietary needs and food preferences for an active and healthy life” as by the Kenya Food Security Steering Group in 2008.

Bayer East Africa, noticed the root cause of the decline rate in food production as most small holder farmers crops are infected with plant diseases (crop diseases). Just like every other living organism, plants are susceptible to diseases that affect the quality and final yield potential of crop production. Crop disease involves harmful deviation or alteration from the normal functioning of the physiological processes of a plant.

More so methods of curbing crop diseases are mainly focused on the physical and chemical ways in which a farmer will use such as proper growing methods, application of chemicals and fertilizers and crop rotation just to mention on a few. And introduction on technological techniques in smartphones has been incorporated such as digital image processing but there are different methods or techniques that are used for digital image processing and coming up with the right technique to detect, quantify and classify crop diseases varies from one method to another hence bringing out different outputs therefore following wrong preventive and curative suggestion measures to be considered to curb with the disease.

# **1.3 Proposed solution**

Developing a mobile based system that detects plant diseases by just taking a picture and provides relevant preventive and curative information on demand. This incorporates provision of Qualified Expert or Artificial Intelligence (image recognition technology Apis that are readily available) and gather statistics based on research to provide instant notification of potential disease outbreaks in various regions. Also, by using of a precise digital image processing technique of method right from design stage for detecting, quantifying and classifying crop diseases.

# **1.4 Objectives**

## **1.4.1 General Objective**

To develop a mobile based system with a precise digital image processing technique for detection, quantification and classifying crop diseases by use of scanners.

## **1.4.2 Specific objectives**

To research on digital image processing techniques for detecting crop diseases using scanners.

To conduct testing on the implementation of the mobile based system.

To document the project on digital image processing techniques for detecting crop diseases using scanners.

# **1.5 Research Questions**

1. What techniques are used in digital image processing for detecting, identifying, quantifying and classifying plant diseases from digital images in the visible spectrum?
2. What are the underlying principles that digital image processing has to meet to detect a crop disease?
3. What are the standards of measures in digital image processing?

# **1.6 Problem justification**

This application will enable small holder farmers to conduct good crop disease controlling methods and take proper preventive and curative measures. The end result will be positive, as there will be increased farm yields of good quality and quantity hence increase market for sales leading to high profits, improve the living standards of the farmers and increased food production margin hence boost the economy financially.

Incorporation of technology appliances for detection of crop diseases is key as technology keeps increasing and use of smartphones has become the norm in developed and developing countries and digital image processing is a rapidly imaging technology that can be use for crop disease detection. David Hughes explained. "Nevertheless, given the expectation that more than 5 billion smartphones will be in use around the world by 2020, almost a billion of them in Africa, we do believe that the approach represents a viable additional method to help prevent yield loss. With the ever-improving number and quality of sensors on mobile devices, we consider it likely that highly accurate diagnoses via the smartphone are only a question of time."

Identifying a disease correctly when it first appears is a crucial step for effective disease management," he said. "With the proliferation of smart phones and recent advances in computer vision and machine learning, disease diagnosis based on automated image recognition, if technically feasible, could be made available on an unprecedented scale.

# **1.7 Scope of the study**

The application is created mainly to be used by the farmer who will notice a certain plant that his affected by a disease and therefore take a picture of it. The application will have a scanner that uses digital image processing technique to detect the type of disease the plant has and therefore give it as a result to the farmer plus the preventive or curative measures that a farmer should take.

The back-end of the application will entail storing of the data for reference, to allow push up notifications over a period of time to check on the progress the plant has made, and allow room for questions and comments.

The application will only be limited to mobile (smart phones) and will require a user-friendly interface where a farmer can log in and proceed using the application. It will also be centered only on the data of the types of crops the application has been set to identify at first then upgrades will be included in the future based on the feedback and milestones we get from the current running application.

# **1.8 Limitations**

Access to organizational or governmental data may be difficult incase researcher meets private or sensitive data. There may also be bias in some data provided to favor different crops over the others

There are also limitations on varieties of crops and the diseases they face as the data will only be for the common food crops and not for all food crops as to find information in them is hard.

The wide-ranging variety of methods of image processing of detection of digital images makes it difficult for one to prospect all possible useful ideas to present results, which can cause potential solutions for problematic issues to be missed.

It's very costly depending on the system used, the number of detectors purchased.

Coming up with the application is time consuming.

Rapid changes in technology, the technique used in implementing in real time application on crop detection keeps changing making it difficult to stick to one method of image processing.

# **1.9 System development methodology**

In order to achieve the objectives of the system, the methodology to be used in this project is Dynamic System Development Method under the entire Agile Software Development. This is approach to system development, and, as the name suggests, develops the system dynamically. DSDM is an iterative and incremental approach that emphasizes continuous user involvement.

Its goal is to deliver projects on time and on budget while adjusting for changing requirements along the way. DSDM is one of a number of agile methods for developing software. This method is particularly useful for the systems to be developed in short time span and where the requirements cannot be frozen at the start of the application building

**Models of Dynamic System Development Method (DSDM)****.**

**Feasibility Study**

This is the phase where the problem is defined and the technical feasibility of the desired application is verified. Apart from these routine tasks, it is also checked whether the application is suitable for Rapid Application Development (RAD) approach or not. Only if the RAD is found as a justified approach for the desired system, the development continues.

**Business Study**,

Here the overall business study of the desired system is done. The business requirements are specified at a high level and the information requirements out of the system are identified.

Once this is done, the basic architectural framework of the desired system is prepared. The systems designed using Rapid Application Development (RAD) should be highly maintainable, as they are based on the incremental development process. The maintainability level of the system is also identified here so as to set the standards for quality control activities throughout the development process.

**Functional Model Iteration**,

The main focus in this phase is on building the prototype iteratively and getting it reviewed from the users to bring out the requirements of the desired system. The prototype is improved through demonstration to the user, taking the feedback and incorporating the changes. This cycle is repeated generally twice or thrice until a part of functional model is agreed upon. The end product of this phase is a functional model consisting of analysis model and some software components containing the major functionality. Design and Implementation.

**Design and Build Iteration,** this phase stresses upon ensuring that the prototypes are satisfactorily and properly engineered to suit their operational environment. The software components designed during the functional modeling are further refined till they achieve a satisfactory standard.

The product of this phase is a tested system ready for implementation. There is no clear line between these two phases and there may be cases where while some component has flown from

the functional modeling to the design and builds modeling while the other component has not yet been started. The two phases, as a result, may simultaneously continue.

**Implementation,** this is the last and final development stage in this methodology. In this phase the users are trained and the system is actually put into the operational environment. At the end of this phase, there are four possibilities: Everything was delivered as per the user demand, so no further development required, a new functional area was discovered, so return to business study phase and repeat the whole process, a less essential part of the project was missed out due to time constraint and so development returns to the functional model iteration or some non-functional requirement was not satisfied, so development returns to the design and build iterations phase. Dynamic System Development Method (DSDM) assumes that all previous steps may be revisited as part of its iterative approach. Therefore, the current step need be completed only enough to move to the next step, since it can be finished in a later iteration. According to this approach, the time is taken as a constraint i.e. the time is fixed, resources are fixed while the requirements are allowed to change.

## **1.9.1 Principles of DSDM**

1. **Active user Involvement is Imperative**-The first principle is considered the most important, because user involvement throughout the project effectively reduces errors in terms of user perception, and therefore reduces error costs.
2. **Teams must be empowered to make Decisions**-To proceed as quickly as possible transaction costs, as resulting from friction in communications of project participants and managers, need to be avoided. Addressing these inefficiencies users and other DSDM participants should be given limited authority to make decisions related to: Requirements in practice, which functionality needs to be in a given increment, prioritization of requirements and features and fine details of the technical solution
3. **Focus on Frequent Delivery**- Frequent deliveries of results ensure that errors are detected quickly, are easily reversed and closer at the source of the error. This applies both to program code as well as to documents like requirements or data models.

**Advantages**

1. It is extendible.
2. Ability of the users to affect the project's direction.
3. It is simple to use.
4. System delivery is on time and on budget.

**Disadvantages**

1. DSDM is costly to implement.
2. As it requires users and developers both to be trained to employ it effectively.
3. It may not be suitable for small organizations.

# **1.10 Data and Information Collection**

## **1.10.1 Primary methods**

## **Interviews**

The stake holders that I will interview are a sample representation of the larger community and will include farmers who are in direct contact with their crops. The information that will be gathered will be very meaningful since there is no biasness.

**Questionnaires**

Here formulation of questions which will be both open-ended and close-ended will be issued to a few stakeholders who will comply by giving their responses. A questionnaire will be developed and sent out to a number of farmers to get their input of the types of crop diseases their crops are facing.

**1.10.2 Secondary methods:**

**Books, journals and internet**

Here reading on what other people have documented on digital image processing for detection of crop diseases using scanners will be done. I will read currently published books and journals on digital image processing in order to gather information.

**CHAPTER TWO**

**2.0 LITERATURE REVIEW**

**2.1Introduction**

Image processing is a method to perform some operations on an image so as to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps of Importing the image via image acquisition tools, Analyzing and manipulating the image and finally giving the output in which result can be altered image or report that is based on image analysis.

There are two types of methods used for image processing namely, analogue and digital image processing. Analogue image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data must undergo while using digital technique are pre-processing, enhancement, and display, information extraction.

**2.2 Theoretical review/Conceptual Framework**

According to Bock et al. (2015), there are several ways to detect plant pathologies. Some diseases do not have any visible symptoms associated, or those appear only when it’s too late to act. In those cases, normally some kind of sophisticated analysis, normally by use of microscopes is necessary. In other cases, the signs can only be detected in parts of the electromagnetic spectrum that are not visible to humans. A common approach in this case is the use of remote sensing techniques that explore multi and hyperspectral image captures. This method that adopt this approach often employ digital image processing tools to achieve their goals. However, due to their many peculiarities and to the extent of literature on the subject. Most diseases, however generate some kind of manifestation in the visible spectrum. This can be solved or at least reduced by use of digital images combined with some kind of image processing. Below steps provides meaningful ways for using digital image processing for detection of crop diseases.

### **2.2.1 Digital image processing techniques in detecting crop diseases.**

**STEP 1: Detection**

Image processing techniques often allows not only detecting the disease, but also estimating its severity, there are not many methods focused only in the detection problem. There are two main situations in which simple detection applies that is: Partial classification where a disease has to be identified amidst several possible pathologies, it may be convenient to perform a partial classification, in which candidate regions are classified as being the result of the disease of interest or not, instead of applying a complete classification into any of the possible diseases. This is the case of the method by Abdullah et al. ([2017](https://springerplus.springeropen.com/articles/10.1186/2193-1801-2-660#CR1)), which is described in Section ‘Neural networks’. Real-time monitoring: in this case, the system continuously monitors the crops, and issues an alarm as soon as the disease of interest is detected in any of the plants. Methods in detection include:

#### **Neural networks**

The method proposed by Abdullah et al. ([2017](https://springerplus.springeropen.com/articles/10.1186/2193-1801-2-660#CR1)) tries to discriminate a given disease from other pathologies that affect rubber tree leaves. The algorithm does not employ any kind of segmentation. Instead, Principal Component Analysis is applied directly to the RGB values of the pixels of a low resolution (15×15 pixels) image of the leaves. The first two principal components are then fed to a Multilayer Perceptron (MLP) Neural Network with one hidden layer, whose output reveals if the sample is infected by the disease of interest or not.

#### **Thresholding**

The method proposed by Sena Jr et al. ([2016](https://springerplus.springeropen.com/articles/10.1186/2193-1801-2-660#CR53)) aims to discriminate between maize plants affected by fall armyworm from healthy ones using digital images. They divided their algorithm into two main stages: image processing and image analysis. In the image processing stage, the image is transformed to a grey scale, thresholded and filtered to remove spurious artifacts. In the image analysis stage, the whole image is divided into 12 blocks. Blocks whose leaf area is less than 5% of the total area are discarded. For each remaining block, the number of connected objects, representing the diseased regions, is counted. The plant is considered diseased if this number is above a threshold, which, after empirical evaluation, was set to ten.

#### **Dual-segmented regression analysis**

Story et al. (2016) proposed a method for monitoring and early detection of calcium deficiency in lettuce. The first step of the algorithm is the plant segmentation by thresholding, so the canopy region is isolated. The outlines of the region of interest are applied back to the original image, in such a way only the area of interest is considered. From that, several color features (RGB and HSL) and texture features (from the gray-level co-occurrence matrix) are extracted. After that, the separation point identifying the onset of stress due to the calcium deficiency is calculated by identifying the mean difference between the treatment and control containers at each measured time for all features. Dual-segmented regression analysis is performed to identify where in time a change point was present between the nutrient-deficit group of plants and the healthy group of plants. The authors concluded arguing that their system can be used to monitor plants in greenhouses during the night, but more research is needed for its use during the day, when lighting conditions vary more intensely.

### **STEP 2: Quantification**

The methods presented in this section aim to quantify the severity of a given disease. Such a severity may be inferred either by the area of the leaves that are affected by the disease, or by how deeply rooted is the affection, which can be estimated by means of color and texture features. Most quantification algorithms include a segmentation step to isolate the symptoms, from which features can be extracted and properly processed to provide an estimate for the severity of the disease. The methods of quantification include:

#### **Color analysis**

Boese et al. (2016) proposed a method to estimate the severity of eelgrass leaf injury, which can be caused by desiccation, wasting disease, and micro herbivory feeding. The first step of the algorithm is the unsupervised segmentation of the leaves into a number of classes (six to ten). In the following, an expert label the classes into one of five possibilities (the three types of injuries, plus healthy tissue and background). After that, the quantification is just a matter of measuring the areas occupied by each of the injuries. According to the authors, their approach still has a number of problems that limit its utility, but it is an improvement over other approaches to quantify complex leaf injuries from multiple stressors.

#### **Fuzzy logic**

In their paper, Sannakki et al. ([5](https://springerplus.springeropen.com/articles/10.1186/2193-1801-2-660#CR49)) presented a method to quantify disease symptoms based on Fuzzy logic. The tests were performed using pomegranate leaves. The algorithm begins converting the images to the color space. The pixels are grouped into a number of classes through K-means clustering. According to the authors, one of the groups will correspond to the diseased areas, however the paper does not provide any information on how the correct group is identified. In the following, the program calculates the percentage of the leaf that is infected. Finally, a Fuzzy Inference System is employed for the final estimation of the disease rating. The details on how such a system is applied are also absent.

#### **Knowledge-based system**

The aim of the work by Boissard et al. (2016) was a little different from the others presented in this paper, as their method tries to quantify the number of whiteflies in rose leaves as part of an early pest detection system. The method employs two knowledge-based systems (KBS) to estimate the number of insects. The first system, the so-called classification KBS, takes the numerical results from some image processing operations, and interprets them into higher level concepts which, in turn, are explored to assist the algorithm to choose and retain only the regions containing insects. The second system, the so-called supervision KBS, selects the image processing tools to be applied, as well as the parameters to be used, in order to collect and feed the most meaningful information to the first system. According to the authors, their proposal had some problems, but it was a good addition to the efforts towards the automation of greenhouse operations.

### **STEP 3: CLASSIFICATION**

The classification methods can be seen as extensions of the detection methods, but instead of trying to detect only one specific disease amidst different conditions and symptoms, these ones try to identify and label whichever pathology that is affecting the plant. As in the case of quantification, classification methods almost always include a segmentation step, which is normally followed by the extraction of a number of features that will feed some kind of classifier. The methods presented in the following are grouped according to the kind of classification strategy employed.

#### **Neural networks**

A very early attempt to monitor plant health was carried out by Hetzroni et al. ([2014](https://springerplus.springeropen.com/articles/10.1186/2193-1801-2-660#CR20)). Their system tries to identify iron, zinc and nitrogen deficiencies by monitoring lettuce leaves. The capture of the images was done by an analog video camera, and only afterwards the images would be digitized. The first step of the proposed algorithm is the segmentation of the images into leaf and background. In the following a number of size and color features are extracted from both the RGB and HSI representations of the image. Those parameters are finally fed to neural networks and statistical classifiers, which are used to determine the plant condition Both networks have one hidden layer, but the number of neurons in the hidden layer is different (40 for texture and 70 for color). The results returned by both networks are then combined, yielding the final classification.

**Support vector machines**

Meunkaewjinda et al. (2016) proposed a method to identify and classify diseases that affect grapevines. The method uses several color representations and throughout its execution. The separation between leaves and background is performed by an MLP neural network, which is coupled with a color library built a priori by means of an unsupervised self-organizing map (SOM). The colors present on the leaves are then clustered by means of an unsupervised and untrained self-organizing map. A genetic algorithm determines the number of clusters to be adopted in each case. Diseased and healthy regions are then separated by a Support Vector Machine (SVM). After some additional manipulations, the segmented image is submitted to a multiclass SVM, which performs the classification into either scab, rust, or no disease.

#### **Fuzzy classifier**

The method proposed by Hairuddin et al. (2015) tries to identify four different nutritional deficiencies in oil palm plants. The image is segmented according to color similarities, but the authors did not provide any detail on how this is done. After the segmentation, a number of color and texture features are extracted and submitted to a fuzzy classifier which, instead of outputting the deficiencies themselves, reveals the amounts of fertilizers that should be used to correct those deficiencies.

#### **Feature-based rules**

In their two papers, Kurniawati et al. (2016) proposed a method to identify and label three different kinds of diseases that affect paddy crops. As in many other methods, the segmentation of healthy and diseased regions is performed by means of thresholding. The authors tested two kinds of thresholding, Otsu’s and local entropy, with the best results being achieved by the latter one. Afterwards, a number of shape and color features are extracted. Those features are the basis for a set of rules that determine the disease that best fits the characteristics of the selected region.

#### **Color analysis**

The method proposed by Wiwart et al. (2018) aims to detect and discriminate among four types of mineral deficiencies (nitrogen, phosphorus, potassium and magnesium). The tests were performed using faba bean, pea and yellow lupine leaves. Prior to the color analysis, the images are converted to the HSI and L\*a\*b\* color spaces. The presence or absence of the deficiencies is then determined by the color differences between healthy leaves and the leaves under test. Those differences are quantified by Euclidean distances calculated in both color spaces.

#### **Self-organizing maps**

The method proposed by Phadikar and Sil. (2016) detects and differentiates two diseases that affect rice crops, blast and brown spots. First, the image is converted to the HSI color space. Then, a entropy-based thresholding is used to segment the image. An edge detector is applied to the segmented image, and the intensity of the green components is used to detect the spots. Each region containing each detected spot is then resized by interpolation, so all regions have a size of 80×100 pixels. The pixel values (gray scale) are finally fed to a self-organizing map (SOM), which performs the final classification.

**2.2.2 Fundamental Principles of digital image processing**

Image representation where a digital image consists of a finite set of values called picture elements or pixels for short. These pixels should be arranged in a regular grid (or raster) of rows and columns, and so it can be useful to think of an image as a matrix. Every pixel in a greyscale image (also called an intensity image) is an 8-bit unsigned integer, meaning that it can have an integer value between 0 and 255. A value of 0 corresponds to pitch black, a value of 255 to pure white, and values between these extremes produce various grey levels between black and white.

Pixel transformations where simple operations that can be performed to alter or improve the appearance of an image. Here the focus specifically on greyscale images, but most of these operations can be utilized on color images by simply handling every channel separately.

Histogram equalization, here the aim of histogram equalization is to construct a pixel transformation function for a given image such that the histogram of the output is approximately uniform. The intensities in the output image will be more evenly spread, often resulting in a contrast enhancement.

Histogram matching to construct an intensity transformation function for a given image, such that the histogram of the output has a pre-specified shape.

Spatial image filters where a pixel’s new value depends on its old value as well as the intensities of pixels in a small neighborhood around that pixel. For every such neighborhood, a pixel in the output image is created with coordinates equal to the coordinates of the center of the neighborhood and the value of that new pixel is the result of the filtering operation.

**2.2.3 Standards of measures in digital image processing**

The American college of Radiology (ACR) Technical Standard for Digital Image for Data management include: accurate labelling and identification of image data in the acquisition, generation and recording of image data. Transmission of images to appropriate storage medium for retrieval for display and formal interpretation, review and consultation. Compression as there should be appropriate image data compression to facilitate storage or transmission without loss of clinically significant information. The following methods can be used to ensure standards are met in digital image processing.

**Image Capture**, where the practices in digital master file are followed. Digital master file main goal of digital reformatting is to produce a rich digital master image file with the attributes such as digitization as this is done in a “use neutral” manner, not for any specific output. The image quality parameters are selected to satisfy most types of outputs. The primary objective is to create digital images that look like the original objects and are “reasonable” reproductions without enhancements. Master Image Files document the object at the time of capture, not what it might have once looked like if restored to its original condition.

**Color Management**. Since all devices interpret and display colors differently, the aim of color management is to bring all of the various devices to a defined state. This is accomplished by building a profile of each device’s response to color for cameras and scanners, the software measures how, the devices record certain defined colors.  In all cases the software then creates a look up table (LUT) to correct for any deviation from the defined color.  This look up table is usually called the input profile (for cameras and scanners) and display profile (for monitors).

**Editing Space**. A color model is a mathematical representation of color (such as RGB or CMYK). It is a general system for assigning numbers to color. A color space is an instance of a color model. There are many color spaces such as ProphotoRGB and sRGB, the important thing to consider in choosing a color space is how much of the visible color spectrum it represents (gamut) and whether it is a well-documented model.

**Resolution and Bit Depth**. Images such as 16-bit files allow editing of the file without any degradation. Unlike bit depth there is no set resolution for any individual resource or collection. Resolution settings depend on a number of factors such as the type of information content (text, continuous tone images, maps, etc.), the final output for the digital file whether screen image or print or both and the needs of the end user. In keeping with the concept of a rich digital master, resolution should be set with the intent of creating an image file that captures all of the information in the original object but not more in order to keep file sizes reasonable.

**Inventory.** For large imaging projects where you will most likely have large batches of images to work with one needs to check to see that the number of files on the batch spreadsheet matches the number of files in the directory, use “Print Folders” to create a directory listing of the image files and import to the batch spreadsheet and check to see that there is a 1:1 correspondence and check to see that file names in the image directory matches the filename on the spreadsheet. Rename if necessary.

**Quality Control.** Check that files have the proper ICC profile, resolution and bit depth. Open image and view at 100%. Scroll through entire image using either the Navigator in Photoshop or the Hand Tool. Check color target for any color bias and perform any minor color corrections. This should rarely be necessary. The color targets should always remain associated with the respective file and should always be kept within the same directory as the image files when moving the images to the archive server (/fstore/).

**2.3 Conclusion**

The ideal method would be able to identify any disease in any kind of plant. Evidently, this is unfeasible given the current technological level. However, many of the methods that are being proposed not only are able to deal with only one species of plant, but those plants need to be at a certain growth stage in order to the algorithm to be effective. That is acceptable if the disease only attacks the plant in that specific stage, but it is very limiting otherwise. Techniques like neural networks, genetic algorithms and support vector machines are very powerful if properly applied in digital image processing.

# 

# **CHAPTER THREE**

**3.0** **SYSTEM ANALYSIS AND DESIGN**

### **3.1. Introduction**

This process entailed the process of finding out the problems and challenges faced by Bayer East Africa Ltd in trying to detect crop diseases by the use of a crop disease detection system. It also described with use of illustrations the various functionalities of the system.

### **3.2 Systems Development methodology**

In order to achieve the objectives of the system, the methodology to be used in this project is Dynamic System Development Method under the entire Agile Software Development. This is approach to system development, and, as the name suggests, develops the system dynamically. DSDM is an iterative and incremental approach that emphasizes continuous user involvement.

Its goal is to deliver projects on time and on budget while adjusting for changing requirements along the way. DSDM is one of a number of agile methods for developing software. This method is particularly useful for the systems to be developed in short time span and where the requirements cannot be frozen at the start of the application building.

# **3.3 Feasibility Study**

The following will analyses the feasibility study:

### **3.3.1 Economic Feasibility**

Introduction

This study looked at the financial assessment of the project in terms of cost-benefit analysis or it is basically to ensure that the system will be affordable to the farmers and the benefit of the system when implemented will weigh out the cost of the system.

### Economic Feasibility Report

According to the analysis done, the organization will incur some cost in purchasing some hardware equipment’s and software which include the purchasing of computers and its peripherals while software cost involves the required software cost, facility cost on money spent in preparation of the of the physical site where the system will be operational such as cabling, flooring and the lighting in additional to air conditioning.

Also, development cost which include the wages and the equipment used in additional to training the staffs, operational cost such as supplies, the maintenance fee of hardware and the software and money paid to professionals responsible for running and maintaining the system.

However, despite these costs the organization is able and willing to invest in this project since the major problems small holder farmers face will be minimized and the way of crop disease detection will be highly improved and thus this project will be economically feasible.

### **3.3.2Operation Feasibility**

**Introduction**

The study was carried out to determine whether the proposed system was meeting the requirements which were gathered during the survey. This feasibility also included setting up of schedules of system implementation.

**Operation Feasibility Report**

For the proposed system to be operationally feasible the organization will have to incur some cost in educating the users on how to work with the new system. The professionals and staff will work in input the data required on the relevant crops and crop disease type. The system will be able give search results based on users input, capture the crop picture taken and give result on the type of crop disease the crop is facing.

### **3.3.3Technical Feasibility**

To be successful for the implementing of the system, the following tools will be required:

Programming tools/Software’s

1. Windows 10 Operating system: To provide a platform for accessing the system.
2. Xampp server: To provide a hosting platform for the system.
3. Android Studio: To provide a platform for writing the code.
4. Database server: to be used for establishing connection between the database and the system.
5. Adobe reader: to be used for reports.

Microsoft word: To be used in writing of project documentation and manuals

Hardware

1. Computer: Core i3. 4GB of RAM 250GB hard disk or higher specifications, to run the application.
2. 2GB Flash Disk: To occasionally port data to another computer if need arises.

**Findings**

The software requirements are readily available for free and thus no much cost will be incurred in attaining them. The hardware requirements are also available for use.

## **3.4 Requirements Elicitation**

The following data collection tools were used:

**3.4.1Questionnaires**

Here, formulated questions which both were open ended and close ended and were issued to the stakeholders (organization managers and technical staff) who complied by giving their responses. This tool helped in the collection of data on the daily activities that were being carried out.

Advantages

* Was easy to Analyze data since the questions that was asked were straight forward.
* Reduced Bias in that most of the questions were in closed format and had answers and therefore answers were straight forward.
* It was relatively quick to collect information since most of the people that were required to answer the questions were available when needed.

**3.4.2Interviews**

Here data was collected where by the researcher moved from one stakeholder (organization managers and Technical staff) to the other looking for necessary information. This involved asking questions and asking for any necessary information that pertained the current system and the improvements that should be made if need be.

The advantages of this is that the researcher was able to get firsthand information that was not biased in any way and also was able to interact with the stakeholders and record their day to day experience with the current system.

## **3.5 Data and System Analysis**

### Analysis of the findings

Below are the individuals who were involved during the data collection phase.

|  |  |
| --- | --- |
| People Involved in Data Collection Process | Number of People Involved |
| Technical staff | 8 |
| Managers | 4 |
| Total | 12 |

Table 1 1Analysis of Findings

## **3.5.1 Report of Questionnaires**

After collecting the data using this tool, the following graphs shows the analysis of some of the major questions contained in the questionnaires for the stakeholders

figure1. 1 Report of Questionnaires

Figure above shows how different farmers responded to the question of does the systems work so as to implement the crop disease detection system in terms of hardware and software necessary in place to develop the crop disease detection system

The figure below shows the sampled data from the stakeholders on how long it takes for the systems to execute in terms of runtime and memory.

figure1. 2 sampled data from the stakeholders

Figure below shows opinion on the cost effectiveness of implementing the system according to questionnaires from the different stakeholders.

figure1. 3 opinion on the effectiveness of the system

### 3.6 Conclusion

According to the analysis shown above, implementation of the proposed system will be of great value to the farmers and stakeholders and will improve greatly the way of enhancing crop disease detection which is easier and faster.

### 3.6.1 Functional Requirements

The system will do the following:

* Recognize and Authenticate users’ credentials.
* Manage and store user information.
* Detect and store crop diseases on the farmers information
* Capture crop disease plant
* Search engine for different types of crops and crop diseases
* Display search results

### 3.6.2 Non-functional Requirements

The non -functional requirements are as follows:

**Adaptability**

The system will be easy to adapt from one type of environment to another without any difficulties.

**Maintainability**

The experts should have the ease of maintaining the system by, correcting errors, preventing breakdown, perfecting the system and ensuring that it adapts to the changing technology and needs of the user.

**Usability**

The system will be friendly to all users with or without much computer knowledge due to simple user interfaces and proper documentation of the system.

**Economy**

The system will be affordable and within the budget specified.

**3.7 System Design**

System design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. System design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

#### **3.7.1. Logical Design**

What a system is doing will change less over time than how it is doing it. This is often conducted via modelling using use case diagrams, class diagrams, Entity- Relationship Diagram to show the flow of activities. In this way, we can furnish an abstraction of the total system through logical design in an orderly explanatory way.

#### **Flow chart Diagram** It shows the flow of activities when the users’ and admin login to the mobile based system

Authorization access

Sign Up

Login

No account

Has account

Sign Up access

Failed

Success

Admin

Home

Home

Farmer

Search

Capture

Results

Results Not found

Results found

Database

Manage Farmers

View Farmer Activities

View Database

Log Out

Start

End

figure1. 4 flow chart diagram

#### **Use case diagram**

Use case diagrams give a graphic overview of the actors involved in a system, different functions needed by those actors and how these different functions are interacted.

Login

Search

Capture

View Results

View Database

Manage Users

View User Activities

Admin

Farmer

Logout

Use Case Diagram

figure1. 5 Use Case Diagram

Sequence diagram, to show how objects operate with one another and in that order. Shows how client and admin interact with the system

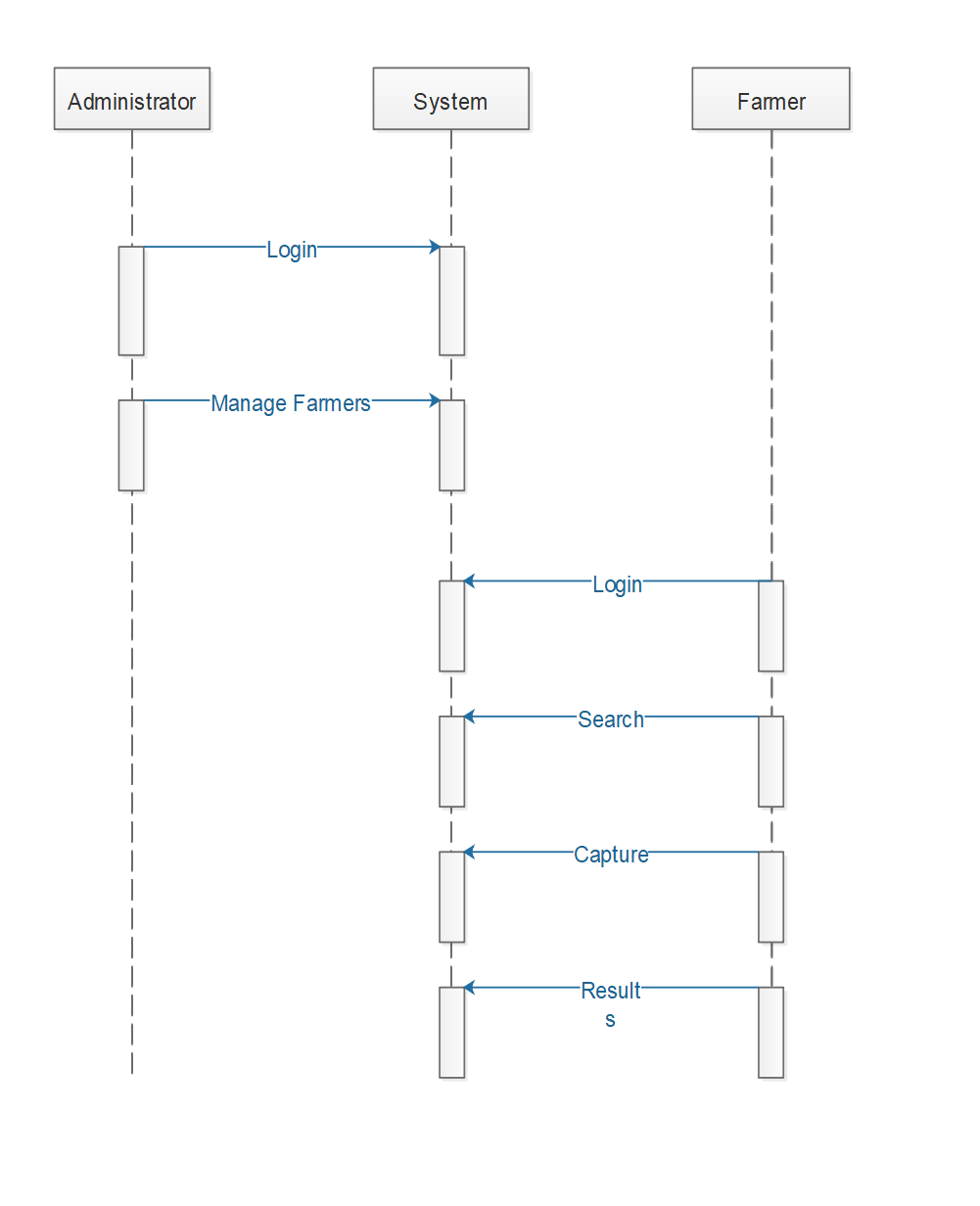


figure1. 6 Sequence Diagram

### **3.7.2 Database Design**

This is the process of producing a detailed data model of the database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language which can then be used to create database. This is also carried out in order to reduce redundancy of information.

**The zero normal form (ONF)**The un-normalized/the zero-normal form data sets would be:

### Un-normalized Tables of Data

User\_id, User\_name, User\_phone, Admin\_id,

Admin\_id, Admin\_username, Admin\_password

**The First Normal Form (1NF)**

All columns must be atomic (no repeating groups). Once the un-normalized data set has been identified, it is converted into the first normal forms and primary keys are identified.

Table 1: Farmer\_id, Farmer\_name, Farmer\_phone, Approver\_id

Table 2plant\_id, plant\_name, plant\_disease, uploader\_id

**The second normal form (2nf)**

This is where all non\_key field depend on component of the primary key. This is guarantee when the primary key is a single field and there must be in first normal form.

Plant\_name, plant\_disease,

**The third normal form (3nf)**

This is where a table must be in second normal form.

No non-key fields depend upon another. That is all non-key field depend on primary key.

Admin\_id=admin \_username, admin \_password, admin \_phone

The database system should have the following tables;

Admin Table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data type** | **Primary Key** | **Null** |
| Admin\_Id | Int (10) | Yes | No |
| Username | Varchar(20) | No | No |
| Password | Varchar (20) | No | No |
| Phone | Int(20) | No | No |

Table 1 2 Admin Table

Farmer Table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data type** | **Primary Key** | **NULL** |
| User \_id | Varchar (20) | Yes | No |
| Username | Varchar (20) | No | No |
| Phone number | Int (20) | No | No |
| Password | Varchar (20) | No | No |
| Approver \_id | Int (20) | No | No |

Table 1 3 Farmer Table

Plant Table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data type** | **Primary Key** | **Null** |
| Plant\_id | INT (30) | Yes | No |
| Plant name | Varchar (20) | No | No |
| Uploader \_id | Varchar (255) | No | No |

Table 1 4 Plant Table

**Results Table.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data type** | **Primary Key** | **Null** |
| results \_id | Int (10) | Yes | No |
| user\_id | Int (10) | No | No |
| Plant\_id | Int (30) | No | No |
| Match percentage | Double(12) | No | No |

Table 1 5 Results Table

**Diseases table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Primary Key** | **Null** |
| Disease\_id | Int(20) | **Yes** | **No** |
| Plant\_id | Int(30) | **No** | **No** |
| Disease\_name | Varchar(255) | **No** | **No** |

Table 1 6 Disease Table

Entity Relationship Diagram

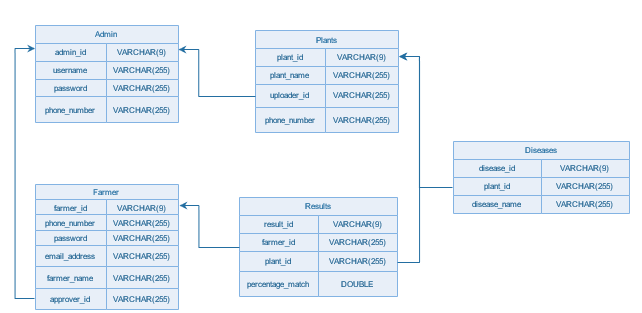


figure1. 7 Entity Relationship Diagram

**3.7.3 Physical design**

To show the actual input and output of the system

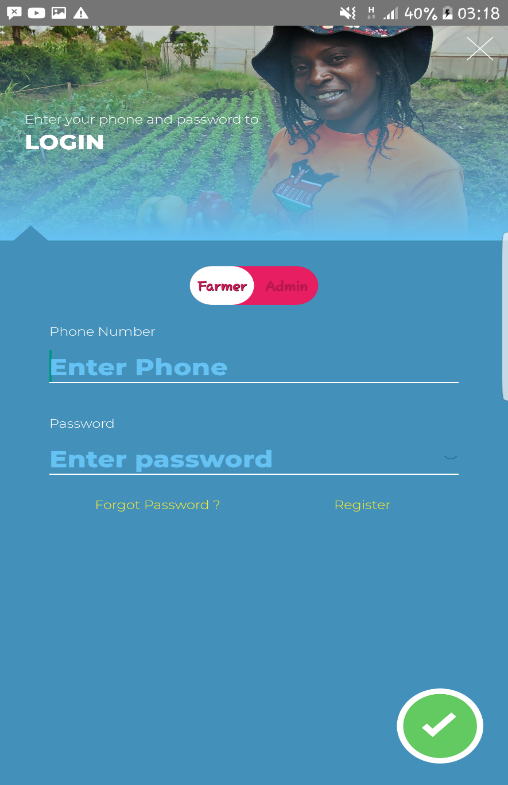


figure1. 8 Login Page Design

To show how the user logs in into the system.

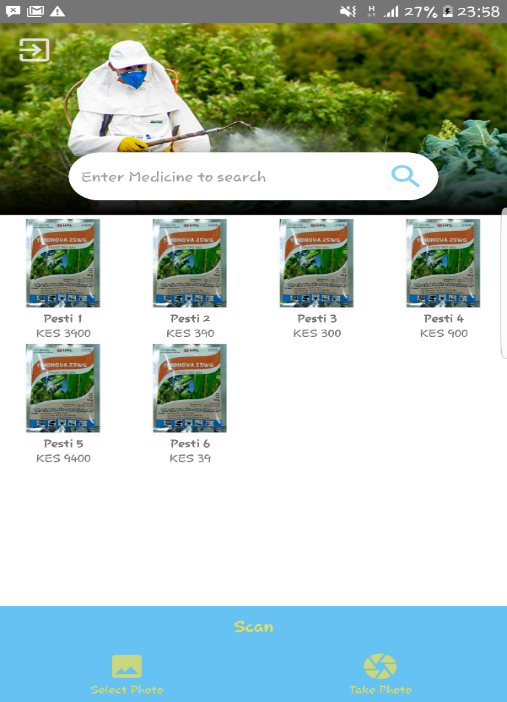


figure1. 9 Home Page Design

To show how the search engine and scan plant functions will look like.

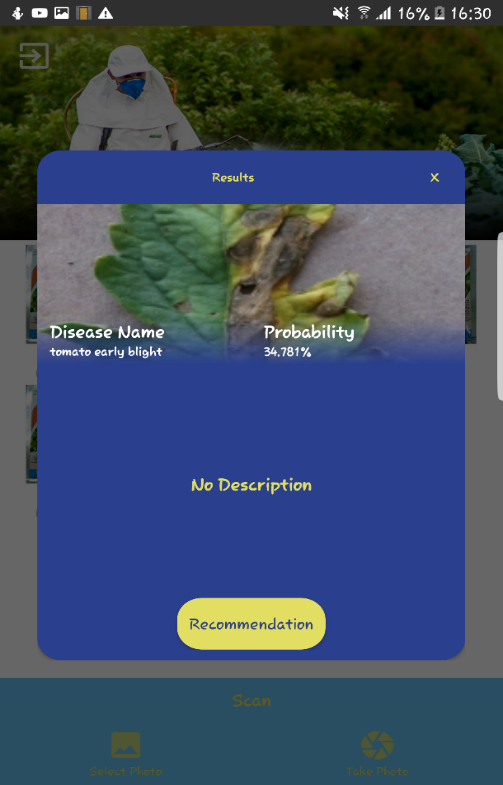


figure1. 10 Results Page Design

To show how the results of crop disease will be displayed.

# **CHAPTER FOUR**

**4.0 SYSTEM CODE GENERATION AND TESTING, CONCLUSIONS AND RECOMMENDATIONS**

## **4.1 Introduction**

This chapter contains the sample code of the system in terms of functions and also the various test plans that were carried out to make sure that the system was undertaking the various activities in the correct manner. The chapter also contains the challenges that were faced during system development and the recommendations suggested.

## **4.2 System code generation**

Below are sample codes that are contained in the system that represent then various functionalities of the system.

## **4.3 Testing**

Testing was done to determine whether the system was meeting the user requirements, it was done after the system was put in place

### **4.3.1 Test plan**

The Software Test Plan is designed to prescribe the scope, approach, resources, and schedule of all testing activities. The plan will identify items to be tested, the features to be tested, the types of testing to be performed, the personnel responsible for testing, the resources and schedule required to complete testing.

**Objectives of Testing**

1. To achieve the correct code and ensure all Functional and Design requirements are implemented as specified in the documentation.
2. To provide a procedure for Unit and System Testing.
3. To identify the test methods for Unit and System Testing.

The following is test plan which was used:

1. Unit testing
2. Integration testing
3. System testing
4. Module testing
5. Acceptance testing

### **4.3.2 Test Results**

### **GUI testing**

This was done to ensure that first the navigation within the system was easy and that

the color used in the entire system and ensuring that the sequencing of events was

proper and happened as specified.

#### **Test case 1**

#### **Error generation**

Expected outcomes:

Proper errors and warning with appropriate error and warning messages should be generated in the case that the user performs a wrong operation or an operation that transforms the database from one consistent state to another, for instance entering wrong password, and the errors generated should be in an appropriate color.

Actual outcomes:

Proper errors and warning were generated when the user performed a wrong operation and suggest the right action. The font of the generated errors was in the color red and a white for warnings.

### **Test case 2**

##### **Confirmation messages**

Expected outcomes:

The user should receive confirmation messages that is some form of feedback on an action he or she carried out on the system for example successful log in.

Actual outcomes:

Confirmation messages for all major operations are generated as some form of feedback to the user confirming their action.

### **Test case 3**

##### **Navigation Verification 1**

Expected outcome

All links or buttons provided on the GUI should lead to the locations where they are really supposed to lead to, both internally and externally, and not wrong destinations.

Actual outcome:

All links or buttons on the GUI lead to the correct destinations and are working appropriately.

#### **Navigation Verification 2**

Expected outcome

Users should be able to view login credentials to confirm, if need be, that they have provided the correct details. They should also be able to view the same credentials (at login and after login), recover blocked accounts and change credentials if they so wish.

Actual outcome:

The users were able to:

1. View credentials at login and after login
2. Recover blocked accounts
3. Change credentials

### . **Unit Testing**

Unit testing on the digital crop detection system concentrated on verification on the smallest element of the software module

|  |  |  |  |
| --- | --- | --- | --- |
| Type of Test | Features to be tested | Given results | Expected results |
| Unit Test | Ability to grant access to the users | The system was able to grant access to the relevant parties depending on their levels of success | The system should be able to allow only those users that exist in the system to login into it |
| Ability to search and generate results | The system was be able to generate results required by the different stakeholders. | At click of buttons in the UI, the system should be able to respond and generate the required results |
| White Box Testing | Internal Structure and design | The system should be able to respond to various queries from the users | The system should be able respond to various queries from the user |

Table 6 Unit test

**Acceptance Testing**

This was carried out to test the user acceptance of the system after completion of the system development and testing. It involved introducing the new system to the user and giving them the system performance.

|  |  |  |  |
| --- | --- | --- | --- |
| Types of Test | Areas To be Tested | Given Results | Expected Results |
| User requirements | Finding out whether the system met the specified requirements specified by the user | The system was able to meet most of the given user requirements while others were much complex to achieve due to time factor and others will be recommended to be included in the future | The system should be able to meet a larger part of the requirements specified by the client while still creating room for additional requirements. |
| Black Box testing | Working of the internal structure and design | All internal components were able to communicate effectively | Efficient internal working structure |

Table 1 7 Acceptance testing

### **Module Testing**

The module interface was tested to ensure that information properly flows into and out of the software unit tested. The local data structure was considered to ensure that data stored temporarily maintained its integrity for all stages in an algorithm’s execution.

|  |  |  |  |
| --- | --- | --- | --- |
| Type of Test | Area To be Tested | Given Result | Expected Results |
| Module Testing | Add user module | The system was able to add new user into the system | The system should be able to add new users into the system and categorize them according to their states |
| logs module | The system administrator should be able to view all logs of activities done in the system. | The system should be able to keep logs. |
| Search, detect and give results module | The module was able to search, detect and give results | The system should be able to search, detect and give results |

Table 1 8 Module testing

### **Integration Testing**

This is the testing was done to find out whether the various modules making up the system were able to work together and generate the expected results. This was also done to determine whether the client and the server interacted correctly.

|  |  |  |  |
| --- | --- | --- | --- |
| Type of Testing | Area To be Tested | Given Results | Expected Results |
| Integration testing | The login, search, capture, detect and give results | The system was able to manipulate the various values and items in the database | The system shall be able to search, capture, detect, and give results as per the database. |

Table 1 9 Integration testing

### **Functional Testing**

It’s done to validate an application and ensure that the system performs all the expected functionalities correctly. It included testing of each function step by step including the database modules.

|  |  |  |  |
| --- | --- | --- | --- |
| Type of test | Areas to be tested | Given results | Expected results |
| Functional testing | Entire system | All the deployed functions were able to perform and also interact with the modules contained in the system | All functions should perform and result to the expected outputs and the interaction between the various components. |

Table 1 10 Functional testing

## **4.4 Conclusion**

Comparing the existing crop disease detection system and the digital image processing crop disease detection system, the latter system was more preferred because it enhanced secure, reliable and faster method of detecting crop diseases. I believe the system has met the predefined requirements thus able to perform its function well effectively and efficiently.

## **4.5 Limitations**

During the process one of the problems I encountered was training the machine on the crop disease data set that I used to be able to output the crop disease detected either by search or capture functions as well as integrating it in the mobile based system.

## **4.6 Recommendations**

I recommend the implementation of the digital image processing crop disease detection in the Bayer East Africa crop disease detection system to help detect crop diseases in a faster and efficient manner hence helping the farmers gain a high-quality crop output. Improvements can be made on this system to accommodate more crops and crop diseases that can be detected by the mobile based system.

**4.7 References**

1. American College of Radiology Technical Standard for Digital Image Management (2016).
2. http://www.siimweb.org/assets/6D4AC337-AE70-42A2-AF7D-93A7938013F3.pdf.
3. Abdullah NE. (2017). Classification of rubber tree leaf diseases using multilayer perceptron neural network. In 2017 5th student conference on research and development. Selangor: IEEE; 2017:1-6.
4. BioMed Central Ltd. (2018). Springer Nature. Retrieved from Digital image processing techniques for detecting, quantifying and classifying plant diseases: https://springerplus.springeropen.com/articles/10.1186/2193-1801-2-660
5. Bock CH. (2015). Plant disease severity estimated visually, by digital photography and image analysis and by the hyperspectral imaging. Critical Rev Plant Sci2015,29(2):59-107.10.1080/07352681003617285.
6. Boese BL. (2016). Digital image analysis of Zostera marina leaf injury. Aquat Bot 2016, 88: 87-90. 10.1016/j.aquabot.2016.08.016
7. Boissard P. (2016). A cognitive vision approach to early pest detection in greenhouse crops. Computer Electron Agric 2016, 62(2):81-93. 10.1016/j.compag.2016.11.009
8. David Hughes, S. M. (2015). Artificial intelligence to diagnose crop diseases. Retrieved from Pennsylvania State University: https://news.psu.edu/story/429727/2015/10/04/research/artificial-intelligence-could-help-farmers-diagnose-crop-diseases
9. Hairuddin MA, Tahir NM, Baki SRS. (2015) Overview of image processing approach for nutrient deficiencies detection in Elaeis Guineensis. In 2015 IEEE international conference on system engineering and technology. Shah
10. Hetzroni A. (2014). Machine vision monitoring of plant health. Adv Space Res 2014, 14(11):203-212. 10.1016/0273-1177(14)90298-4
11. IFPRI. (2018). Food security Portal. Retrieved from http://www.foodsecurityportal.org/kenya/food-security-report-prepared-kenya-agricultural-research-institute
12. Imaging standards and procedures, Columbian University Libraries,2019. https://library.columbia.edu/bts/imaging.html
13. Kurniawati et al. (2016) Investigation on image processing techniques for diagnosing paddy diseases. In 2016 international conference of soft computing and pattern recognition. Malacca: IEEE; 2016a:272-277.
14. Meunkaewjinda A, Kumsawat P, Attakitmongcol K, Srikaew A. (2016) Grape leaf disease detection from color imagery using hybrid intelligent system. In 2016 5th international conference on electrical engineering/electronics, computer, telecommunications and information technology. Krabi: IEEE; 2016:513-516.
15. Phadikar S, Sil J. (2016) Rice disease identification using pattern recognition techniques. Khulna: IEEE; 2016. pp 420–423
16. Sannakki et al. (2015). Leaf disease grading by machine vision and fuzzy logic. Int J 2015, 2(5):1709-1716.
17. Sena DG Jr, Pinto FAC, Queiroz DM, Viana PA. (2016) Fall armyworm damaged maize plant identification using digital images. Biosyst Eng 2016, 85(4):449-454. 10.1016/S1537-5110(16)00098-9
18. Story D, Kacira M, Kubota C, Akoglu A, An L. (2016) Lettuce calcium deficiency detection with machine vision computed plant features in controlled environments. Computer Electron Agric 2016, 74(2):238-243. 10.1016/j.compag.2010.08.010
19. Sygenta Global. (2018). Empower smallholders. Retrieved from https://www4.syngenta.com/what-we-do/the-good-growth-plan/empower-smallholders?utm\_source=google&utm\_medium=cpc&utm\_content=empower-smallholders&utm\_campaign=good-growth-plan&gclid=Cj0KCQiAw5\_fBRCSARIsAGodhk\_OUgR3jbX5V9cgbQLPmrE8QuciwiUWvo5fndGBryHUDTk1js

**4.8 Appendices**

Resources, Time plan and Budget

# **4.8.1Project requirements**

Hardware requirements:

* One laptop with the following requirements
  + RAM 4GB
  + 2.5GHz
  + 500GB hard disk
* Modem and internet bundles
* HD printer
* Backup media such as Blank CDs and Hard disk

Software requirements:

* Windows 10
* Mozilla Firefox
* Office 2016
* Android Studio Version 3.0
* IntelliJ
* PyCharm

Other requirements:

* Stationary such as pens, books
* Travel costs and other miscellaneous costs

# **4.8.2Project budget**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NO** | **ITEM NAME** | **ITEM QUANTITY** | **PRICE PER ITEM** | **TOTAL** |
| 1 | HP LAPTOP | 1 | 50,000 | 50,000 |
| 2 | Android Studio | 1 | Free license |  |
| 3 | Faiba modem | 1 | 4,000 | 4,000 |
| 4 | PyCharm | 1 | Free License |  |
| 5 | HP LaserJet printer | 1 | 11,500 | 11,500 |
| 6 | Stationary (pens, books) | 2 | 200 | 200 |
| 7 | Travel costs | Per trip | 500 | 500 |
|  |  |  | **TOTAL** | **66,200** |

Table 1 11 **Project Budget**

**4.8.3 Project schedule**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S/N** | **ACTIVITY** | **DURATION IN HOURS** | **EXPECTED START DATE** | **EXPECTED END DATE** | **ACTUAL START DATE** | **ACTUAL END DATE** | **DELIVERABLES** |
| **1.** | Project idea generation | 10 Hours | 17/9/2018 | 21/9/2018 | 24/9/2018 | 10/10/2018 | Project Idea |
| **2.** | Project Proposal | 10 Hours | 24/9/2018/ | 15/11/2018 | 14/10/2018 | 21/11/2018 | Complete project proposal |
| **3.** | Feasibility study | 10 Hours | 19/12/2018 | 30/12/2018 | 8/1/2019 | 29/1/2019 | Questionnaires and data collection |
| **4.** | System analysis and design | 20 hours | 7/1/2019 | 14/3/2019 | 5/2/2019 | 19/3/2019 | End user requirement |
| **5.** | Coding and Testing | 4 hours per day | 21/1/2019 | 1/3/2019 | 8/2/2019 | 6/4/2019 | System compile-time and run-time |
| **6.** | Deployment and installation | 5 hours | 4/3/2019 | 8/3/2019 | 6/4/2019 | 9/4/2019 | Application installation |
| **7.** | Documentatio**n** | 10 hours | 17/9/2019 | 15/3/2019 | 24/9/2019 | 8/4/2019 | Application documentation |
| **8.** | Maintenance |  | 1/4/2019 | 12/4/2019 | 8/4/2019 |  | Updating and maintenance of the application |

Table 1 12 Project Schedule

# **4.8.4 Gantt chart**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TIVITY | **17th Sept,2018**  **To**  **21stNov,2018** | **19 Dec,2018**  **To**  **21 Jan, 2019** | **7th Jan,2019**  **To**  **5th Feb 2019** | **4th Mar,2019**  **To**  **8th April,2019** |
| Project Proposal and idea  Generation |  |  |  |  |
| Feasibility  study |  |  |  |  |
| System  Analysis |  |  |  |  |
| Coding and Testing |  |  |  |  |
| System Implementation |  |  |  |  |
| Documentation and Maintenance |  |  |  |  |

figure1. 11 Gantt Chart

**4.8.5 Sample code**

package org.tensorflow.demo.Account.Fragments;  
  
  
import …  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \*/*public class CreateAccountFragment extends Fragment implements View.OnClickListener {  
  
 @BindView(R.id.*swUserType*)  
 MySwitch swUserType;  
 @BindView(R.id.*ivBackButton*)  
 ImageView ivBackButton;  
 @BindView(R.id.*etName*)  
 EditText etName;  
 @BindView(R.id.*etPhoneNumber*)  
 EditText etPhoneNumber;  
 @BindView(R.id.*llButtonDone*)  
 LinearLayout llButtonDone;  
  
 @BindView(R.id.*tvLogin*)  
 TextView tvLogin;  
 AccountActivity parentActivity;  
 SweetAlertDialog loadingDialog;  
 AwesomeValidation awesomeValidation = new AwesomeValidation(ValidationStyle.*BASIC*);  
  
 public CreateAccountFragment() {  
 // Required empty public constructor  
 }  
  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 View rootView = inflater.inflate(R.layout.*fragment\_create\_account*, container, false);  
 parentActivity = (AccountActivity) getActivity();  
 ButterKnife.*bind*(this, rootView);  
 setupViews();  
  
 return rootView;  
 }  
  
 public void setupViews() {  
 ivBackButton.setOnClickListener(this);  
 etPhoneNumber.setOnClickListener(this);  
 llButtonDone.setOnClickListener(this);  
 tvLogin.setOnClickListener(this);  
  
 awesomeValidation.addValidation(etPhoneNumber, Utils.*REGEX\_KENYAN\_PHONE*, "Please enter the correct phone number format ex. 0712345678");  
 }  
  
 @Override  
 public void onClick(View v) {  
 switch (v.getId()) {  
 case R.id.*ivBackButton*:  
 Utils.*startNewActivity*(parentActivity, AccountActivity.class, true, null);  
 break;  
 case R.id.*llButtonDone*:  
 if (awesomeValidation.validate()) {  
 String name = etName.getText().toString();  
 String phoneNumber = etPhoneNumber.getText().toString();  
 Utils.AccountType accountType = Utils.AccountType.*FARMER*;  
 VerificationUtils verificationUtils = new VerificationUtils(parentActivity);  
 parentActivity.createAccountModel = new AccountUtils.CreateAccountModel(accountType , name, phoneNumber);  
 loadingDialog = Utils.*startDialog*(parentActivity, "Sending Verification Code", "We would like to verify that your phone number is OK", SweetAlertDialog.*PROGRESS\_TYPE*);  
 verificationUtils.requestVerificationCode(parentActivity.createAccountModel.getAccountType(), parentActivity.createAccountModel.getPhoneNumber()).subscribe(new Observer<String>() {  
 @Override  
 public void onSubscribe(Disposable d) {  
  
 }  
  
 @Override  
 public void onNext(String response) {  
 try {  
 JSONObject jObject = new JSONObject(response);  
 int success = jObject.getInt("success");  
  
 switch (success) {  
 case 1:  
 loadingDialog.setTitleText("Verification Sent");  
 loadingDialog.changeAlertType(SweetAlertDialog.*SUCCESS\_TYPE*);  
 loadingDialog.setContentText("Please use the Code sent to your phone to create your account");  
 loadingDialog.setConfirmText("OK").setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener() {  
 @Override  
 public void onClick(SweetAlertDialog sweetAlertDialog) {  
 Utils.*setVisibleFragment*(parentActivity, parentActivity.createPasswordFragment, true, R.id.*frame\_container*);  
 loadingDialog.dismissWithAnimation();  
 }  
 });  
 break;  
 default:  
 loadingDialog.setTitleText("Verification Failed");  
 loadingDialog.changeAlertType(SweetAlertDialog.*WARNING\_TYPE*);  
 loadingDialog.setContentText("We couldnt send the Verification Code to your phone. Please retry");  
 loadingDialog.setConfirmText("OK").setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener() {  
 @Override  
 public void onClick(SweetAlertDialog sweetAlertDialog) {  
 loadingDialog.dismissWithAnimation();  
 }  
 });  
 break;  
 }  
  
  
 } catch (JSONException e) {  
  
 loadingDialog.setTitleText("Verification Failed");  
 loadingDialog.changeAlertType(SweetAlertDialog.*ERROR\_TYPE*);  
 loadingDialog.setContentText("We couldnt send the Verification Code to your phone. Please retry");  
 loadingDialog.setConfirmText("OK").setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener() {  
 @Override  
 public void onClick(SweetAlertDialog sweetAlertDialog) {  
 loadingDialog.dismissWithAnimation();  
 }  
 });  
 e.printStackTrace();  
 }  
  
  
 }  
  
 @Override  
 public void onError(Throwable e) {  
 Log.*d*("loginError",e.toString());  
 loadingDialog.setTitleText("Verification Failed");  
 loadingDialog.changeAlertType(SweetAlertDialog.*ERROR\_TYPE*);  
 loadingDialog.setContentText("Please check that your internet is fine and retry");  
 loadingDialog.setConfirmText("OK").setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener() {  
 @Override  
 public void onClick(SweetAlertDialog sweetAlertDialog) {  
 loadingDialog.dismissWithAnimation();  
 }  
 });  
 }  
  
 @Override  
 public void onComplete() {  
  
 }  
 });  
 }  
  
 break;  
 case R.id.*tvLogin*:  
 Utils.*setVisibleFragment*(parentActivity, parentActivity.loginFragment, false, R.id.*frame\_container*);  
  
 break;  
 }  
 }  
  
}

**4.8.6 Screen Shots of Tested Results**

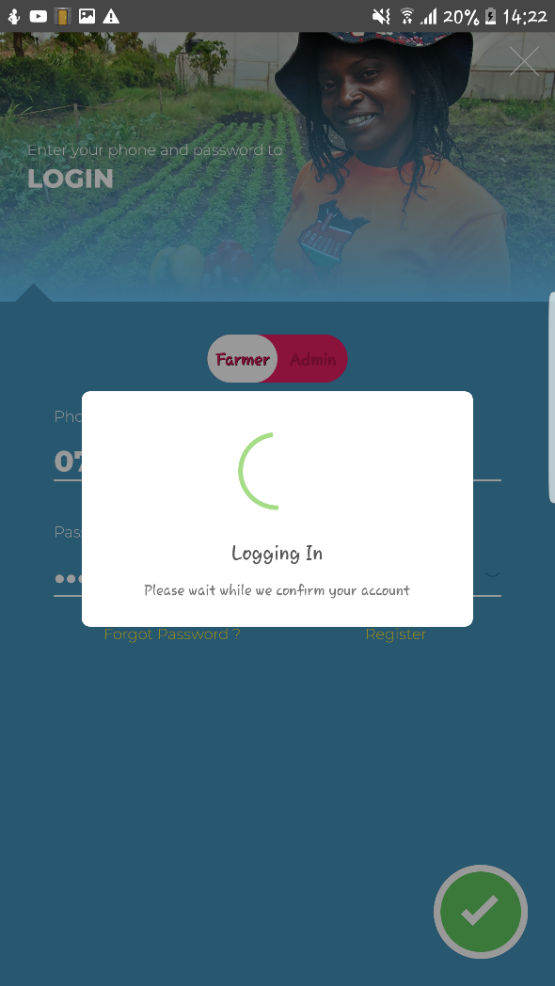
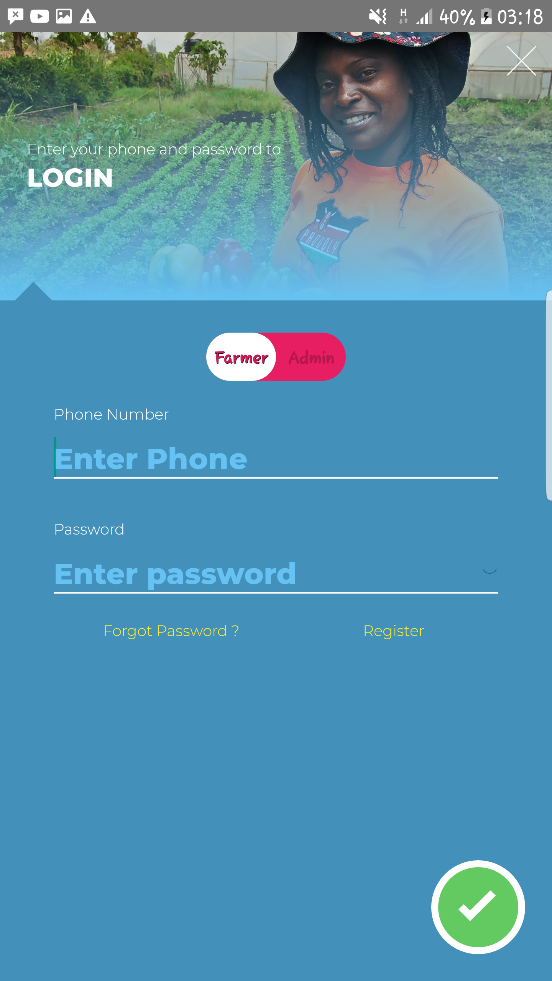


figure1. 12 Login Page Test

The login of user credentials of phone and password

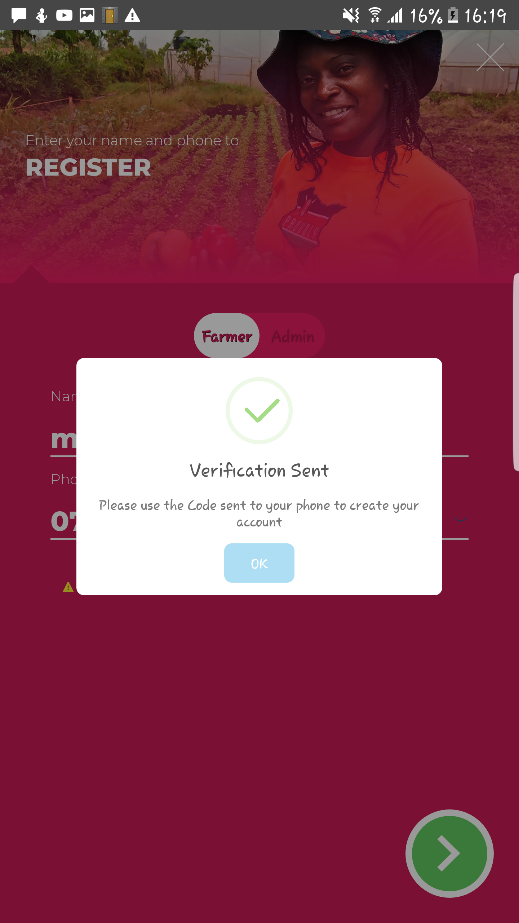
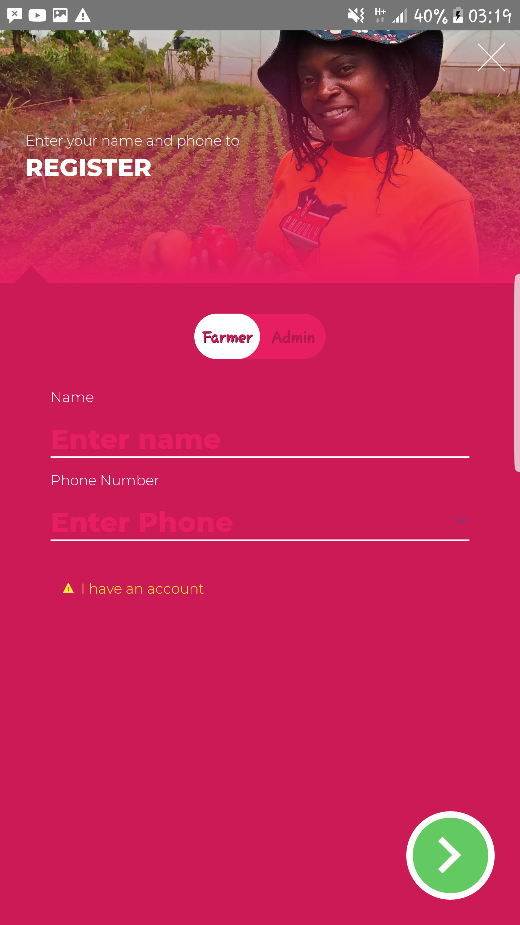
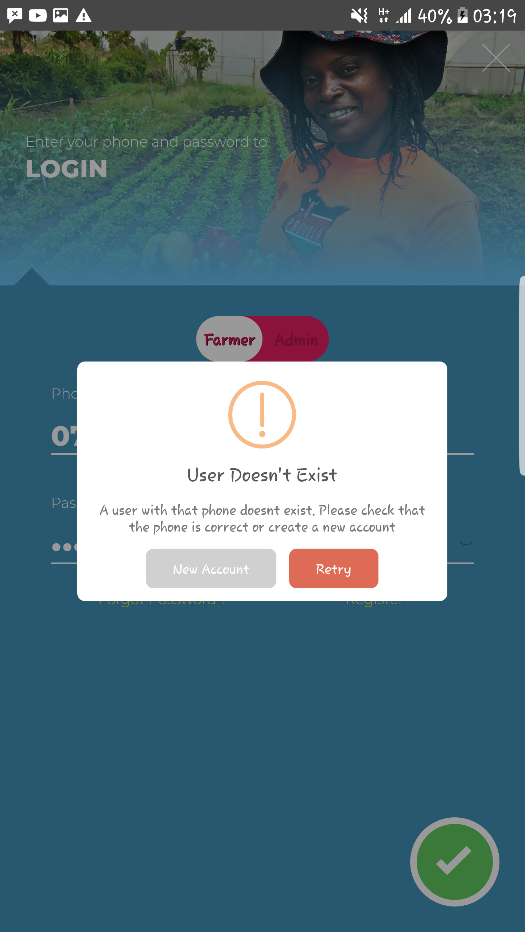


figure1. 13 Register Page Test

Registration process of user credentials and verification code set so as user to set password.

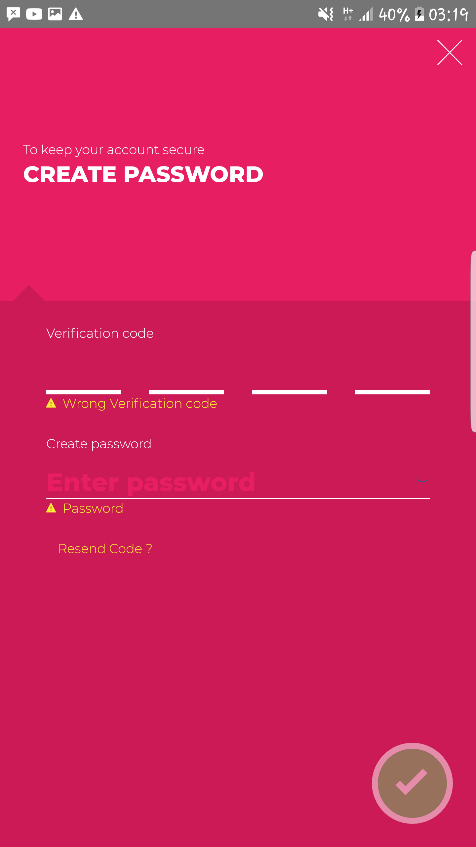
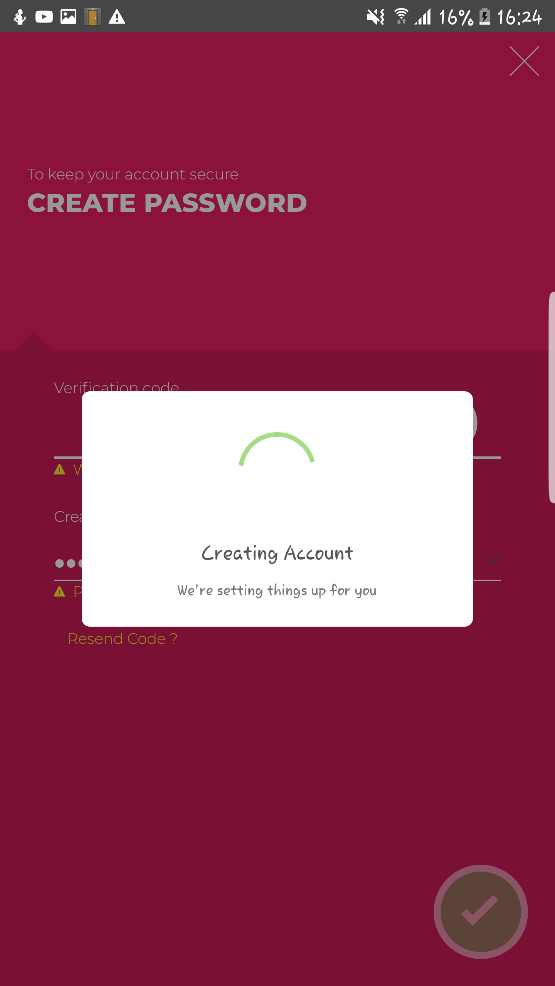
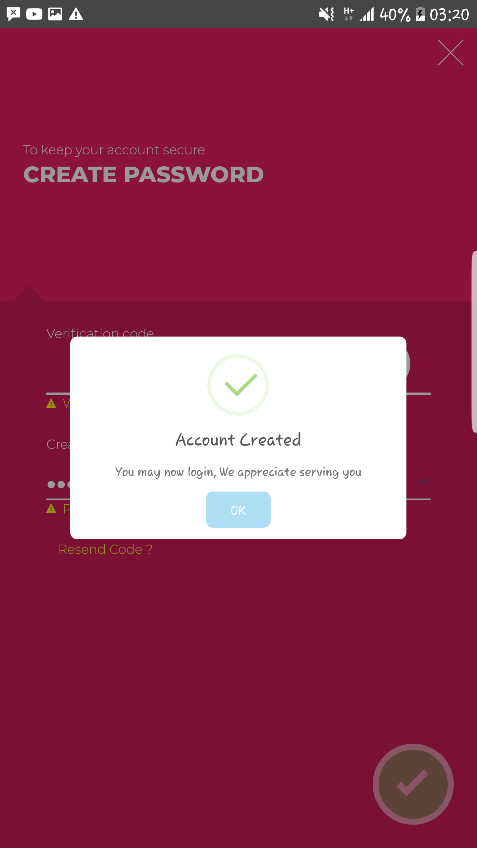
 

figure1. 14 Account Creation Test

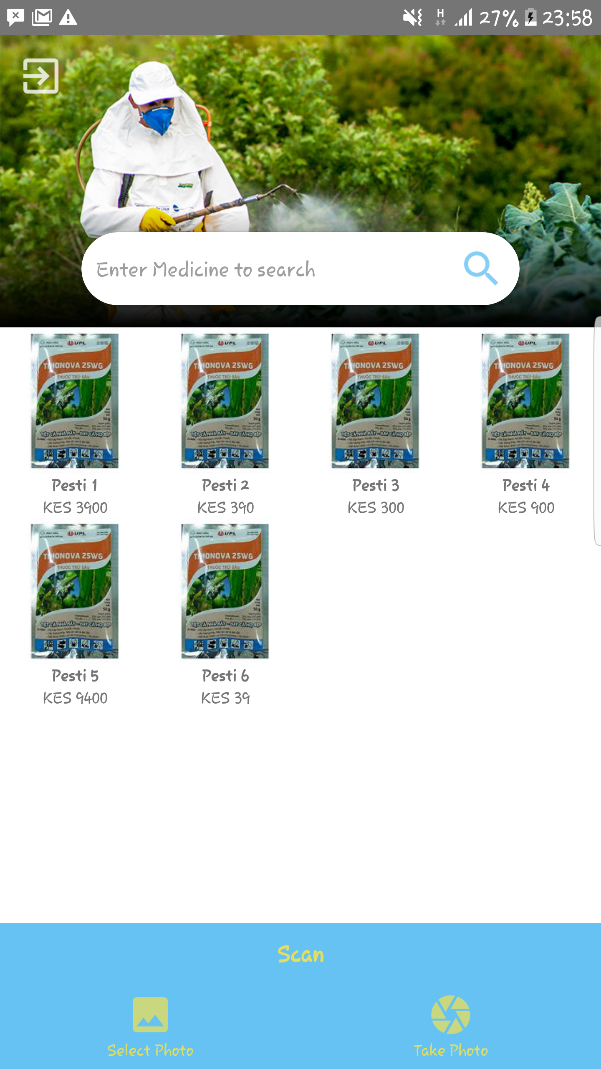


figure1. 15 Home Page Test

After successful creation of user account, a user can go ahead to search for plant medicine/fertilizer, select a photo already taken previously by user to scan crop disease or take photo of a plant and scan for crop disease.

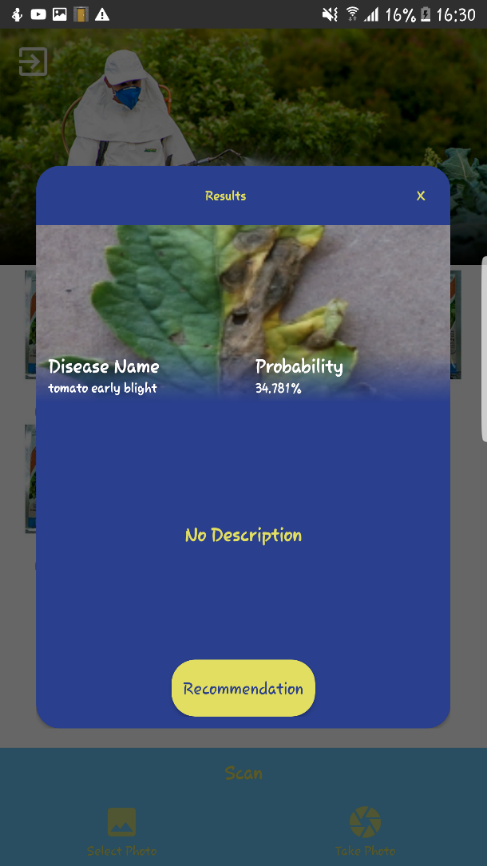


figure1. 16 Results Page Test

The results page shows the type of crop disease the plant has from the scanned images and its probability of its severity.

**4.9 QUESTIONNAIRES**

Questionnaire for Bayer East Africa Ltd managers

Dear Respondent,

I'm MARY GLADYS NJOKI WAWERU from Jomo Kenyatta University of Agriculture and Technology pursuing a Bachelors’ degree in information technology. Please take a few minutes to express your opinions on the below questions. Your answers are important to the success of this study. Please answer the following questions with ‘yes or no’ where required. This is the questionnaire that deals with Bayer East Africa Ltd management team on their view on implementation of the crop disease detection system.

SECTION A:

1. Personal Details

Name

Phone number

Address

Email

SECTION B: FARMERS OFFICIALS DETAILS

1. Are there hardware and software necessary in place to develop the crop disease detection system memory?

Yes []

No []

1. How long does it take for the systems to execute in terms of memory and speed?

Fast [ ]

Slow [ ]

No response [ ]

1. How cost effective is it to develop a crop detection system in a rate of zero to ten?

0-10[ ]

1. Are there any improvements you likely to be included in the systems according to the accountability of implementing the crop disease detection system?

YES [ ]

NO [ ]

Questionnaire for Bayer East Africa Ltd Technical staff

Dear Respondent,

I'm MARY GLADYS NJOKI WAWERU from Jomo Kenyatta University of Agriculture and Technology pursuing a Bachelors’ degree in information technology. Please take a few minutes to express your opinions on the below questions. Your answers are important to the success of this study. Please answer the following questions with ‘yes or no’ where required. This is the questionnaire that deals with Bayer East Africa Ltd technical staff on their view on implementation of the crop disease detection system.

SECTION A:

1. Personal Details

Name

Phone number

Address

Email

SECTION B: FARMERS OFFICIALS DETAILS

1. Are there hardware and software necessary in place to develop the crop disease detection system memory?

Yes []

No []

1. How long does it take for the systems to execute in terms of memory and speed?

Fast [ ]

Slow [ ]

No response [ ]

1. How cost effective is it to develop a crop detection system in a rate of zero to ten?

0-10[ ]

1. Are there any improvements you likely to be included in the systems according to the accountability of implementing the crop disease detection system?

YES [ ]

NO [ ]