Neighbourhood Farm
Stock System and Data Analysis

Tawira Valentine
[1]UWC, Robert Sobukwe Road, Bellville, Cape Town, 7535, South Africa
Tel: +27 82478617, Email: kudatawira@gmail.com

Abstract: Neighbourhood Farm is initiative committed to the development of community market gardens for the growing of organic crops at schools in the Deep South of Cape Town stretching from Muizenberg through to Kommetjie. This project proposal will attempt the implementation of ICTs to amplify the farming operation of the Neighbourhood farm initiative. This part of the project will attempt to design an ICT based solution to resolve potential problems relating to stock system and data analysis.

Keywords: SMS-Stock Management System, Food Security, URD

1. Introduction

Food insecurity is a major global challenge in both developing and developed countries with its main effects being heavily felt in poor marginalized populations within these countries. Food security exists when all people, at all times, have physical and economic access to sufficient, safe and nutritious food to meet their dietary needs and food preferences for an active and healthy life [1]. According to some population statistics, of the estimated additional 2.2 billion people who may be added to the world's population between 2017 and 2050, 1.3 billion will be added in Africa [2]. From the previous fact, food security has become an area of interest that requires much attention especially in the sub-Saharan region. Governments and non-governmental organizations across the globe have been pursuing programs and policies aimed at attaining food security in the past and recent years, with much effort being channeled in the developing countries where insufficient food is a more a common issue. Despite the efforts, most commonly when such programs and policies are set up and initiated, they face several different challenges that threaten their impact in contributing to food security. Some of the challenges that the programs face can be linked broadly to operations and management. In this project we propose the implementation and use of Information Communication Technologies (ICTs) based solutions as amplifiers in the management of operations at farm level specifically concentrating on the Neighbourhood Farm (NF) initiative, a project that is build up inline with the food security programs. ICTs in the Neighbourhood Farm will be implemented in the areas of stock management, data analysis, payment and pre-order .However this part of the project aims to use ICTs in the areas of stock management and data analysis.

1.1 About Neighbourhood Farm

Neighbourhood Farm is a project committed to the development of community market gardens for the growing of organic crops at schools in the Deep South of Cape Town stretching from Muizenberg through to Kommetjie. The communities in this area represent a full economic spectrum from well-resourced to critically in need. The Neighbourhood Farm project does not just aim to grow vegetables and community spirit, moreover there's an economic heart to the project as well. Beyond showing children how food is grown, the market garden provides a small revenue stream from distributing and selling healthy food to both the school and the local community. This not only makes the gardens sustainable but empowers the community through skills-based training and experience. In turn, these create further economic and entrepreneurial opportunities for marginalized community members [3].

1.2 Problem Statement

Efficiency in the operation of the Neighbourhood Farm initiative is one big stumbling block that can threaten the effective management of the whole initiative. Most projects of similar structure usually fail because of uneconomical management and operations hence a crucial need for the Neighbourhood Farm initiative to operate effectively [1]. With quite a number of players involved in the project operations, complex actions are likely to be experience in the areas of general management of the farms (schools) that include stock management and the analysis of data.

Stock Management Problem

The movement of produce stock in and out of the farm needs to be carefully managed and monitored. The management of stock (produce) is very significant in the successful operations of each farm and the entire project at large mainly because it influences decisions relating to the quantities that need to be harvested, how much produce to be requested from the other farms as well as how much can be transferred to the other nearby farms among other important reasons. The ability to maintain a smooth flow of produce between the farm ground and the shop floor is a a key element. Currently the movement of produce between the farm ground and the shop floor is being tracked in a piece of paper that can be easily misplaced or damaged, resulting in discrepancies in the movement of produce. Allowing third party producers to supply the farm shop with their produce as well as the option of requesting stock from other farms needs a very effectual stock management system to keep track various forms of produce. Poor stock management system often results wastages and loss of the produce (food) which has be argued to be one of the major contributors to food insecurity.

Data Analysis

Analysing data does not only add operational value through motivation but also goes a long way in making inferences about future performance. Problems are likely to be faced in efforts to determine the performances of products at farm level as well as at the project at large. The tracking of produce wastage and loss is also one potential solution to improve efficiency and productivity of the Neighbourhood Farm project.

1.3 Related Work

Malawi Agricultural Commodity Exchange Market (MACE)

MACE is a commodity exchange market platform that was introduced in Malawi by the government of Malawi. The ICT based MACE was introduced to attend to the the constraint of poor access to market information by farmers in order to strengthen the liberalised markets [4]. It was initiated to assist smallholder farmers to have access to market related information to search for better markets and prices for their produces.

Through the use of MACE, an improved market performances of rice markets in Malawi was recorded. The success of the MACE provides a strong argument for the usefulness of implementing ICTs in farming or agriculture.

Israel Farm Management Department on-farm software package (Ma'ayan)

The software runs on a PC and enables the farmer to record production activities, monitor, quantify and cost them, make on farm decisions and follow product marketing. Ma'ayan was introduced after the Farm Management Department noticed that many farmers hand record their relevant data that relates to agricultural production, cost accounting parameters and other critical information resulting in an inability to efficiently recall, analyse and meaningfully use information when needed [5]. The first version of the software was released in 1991 and received favourable acceptance. Following on-farm visits and farmer training updates to the software were made and within a few years from the first version more than 600 copies of the software were sold.

Open Food Network

.The Open Food Network is an online marketplace for local food. It enables a network of independent online food stores that connect farmers and food hubs (including coops, online farmers' markets, independent food businesses etc); with individuals and local businesses [6]. It aims to provide farmers and food hubs an easier and fairer way to distribute their food. The Network was initially was started in Australia and is growing to all the other countries including the recent launch in South Africa.

1.4 Proposed Solution

The implementation of ICTs to counter solve the problems that the Neighbourhood farm project will possibly face is one viable solution. Effective implementation of Information Communication Technologies will not only solve the stipulated problems but will also enhance and improve the farming practices in the communities. Using Information Communication Technology to effectively manage the stock system and perform data analysis amplifies the management and operations of the whole initiative.

Stock Management System (SMS)

The proposed stock management system aims to allow effective, productive and flexible stock operations between farmer, seller and third party players.

Data analysis

The proposed system aims to extract and analyse data from all parties involved in the operations of the project. The system will view product performances, track wastages and loss in produces among other critical data analysis solutions.

2. User Requirements

The system will be designed to be used by four users who are entitled to different privileges and are exposed to different functionalities. The first user is a Super User who have total control of the system and who is responsible for managing all the farms (schools). An Admin User is privileged to take control of all activities for a specific farm under him. A Third Party supplier is a community member who is registered with any farm who is entitled to supplier a farm with his produces and will have an option to follow up on his produces using the system. The forth user is a Customer who also can be any community member or organization and is entitled to use the system to follow up orders and contributions to the neighbourhood farm. For this part of the project the user requirements that will be defined are that aligned to the interaction of all these different users with the SMS and Data Analysis.

2.1 Requirements Gathering

For this project a number of requirements gathering techniques were performed in order to have a clear set of the user requirements and a concise view of the expected solutions. First a formal meeting was conducted where the founder of the Neighbourhood Farm initiative introduced the background and underlying details of the project to set a clear picture of the project. Also a similar meeting was set up a few weeks from the first meeting but this second meeting aimed at outlining a high level view of the requirements and addressing some questions from the first gathering. A pilot visit to one of the school that will be used as a model farm followed the two formal meetings, also a visit to a farm with similar setup was done and a more enhanced understanding of the farm set up was gained. The last meeting followed a couple of weeks later with the aim of presenting a first version of a prototype and build up feedback as well as suggestions from the project manager.

Table 1: Requirements Gathering Table

Date	Technique	Goal	Identified requirements
9 Feb 2018	*formal meeting*	● *Background of the Neighbourhood Farm project* ● *Underlying Details*	*Stock management* *Data analysis*
27 Feb 2018	*formal meeting*	● *Highlight problems and requirements* ● *Answer questions and clarifying concepts*	*Produce requisitions* *Produce transfers*
10 Mar 2018	*pilot visit*	● *View ground set up* ● *View similar farm set up*	*Record processed stock* *Track wastages*
7 April 2018	*formal meeting*	● *Presenting first prototype* ● *Get feedback and suggestions*	*Track produce shelf expiry* *View demands performances*
Weekly Meetings (Group)	*Brainstorm*	● *Updates on progress and challenges*	*Order follow ups* *Farm performances*

3. Requirements Analysis

With the user requirements identified during the gathering process, the requirements analysis proceeds, breaking down the gathered requirements into more compact functional and non-functional requirements.

3.1 Functional Requirements

An overview of the high level functional user requirements that were identified and recorded out through the requirements gathering process are formatted in *Fig 1*. The requirements in the above fig are not organised systematically and are organised according to the user type.

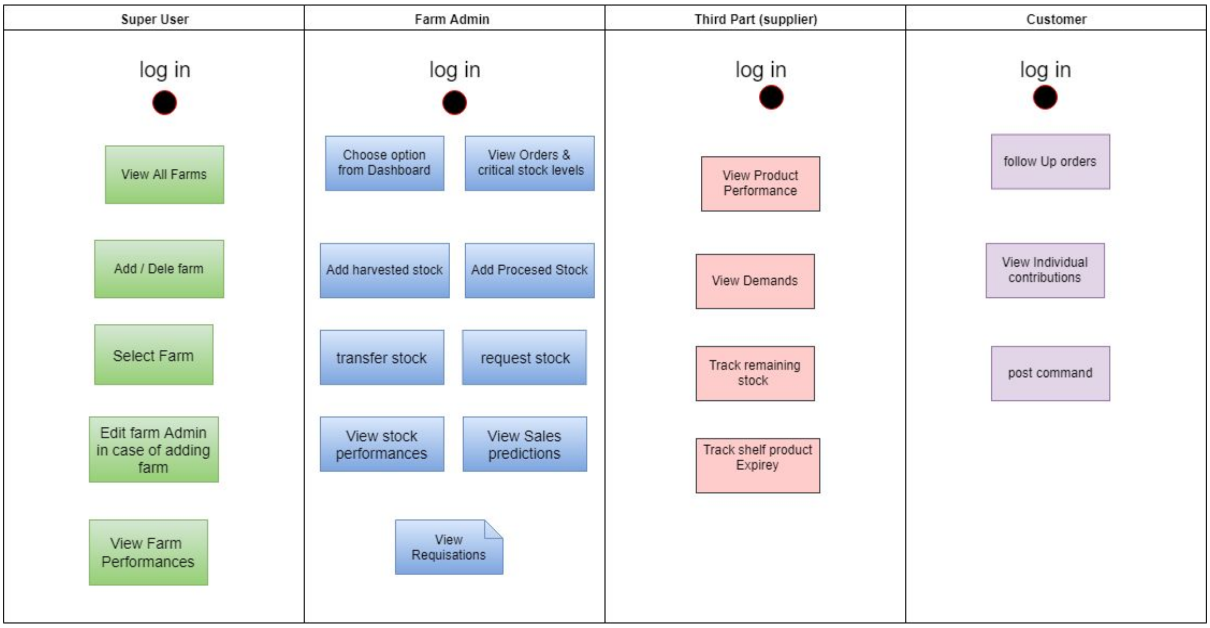


Fig 1: User Requirements Diagram (URD-functional requirements)

3.2 Non-Functional Requirements

A number of non-functional requirements were identified during the data gathering process and some of the requirements include visibility, usability, security and capacity.

Visibility
The aim is to design a simplified system with the most used options clearly visible. Critical notifications should be pushed to the home screen in order to be easily noticed.

Usability
There is a need to design a platform that is easy to use with little or no training needed to operate the system. The main emphasis is to design in a way that allow user to use recognition other than recall

Capacity
There is a need to enable the system especially the SMS to handle large volumes of stock data as large volumes of harvested produces will be pushed into the system.

Security
There is a need to ensure that there is security and data privacy for the users

3.3 Use Case Diagrams

With the user requirements identified during the gathering process, the user requirements from the ERD in Fig 1 can be analysed using the table below.

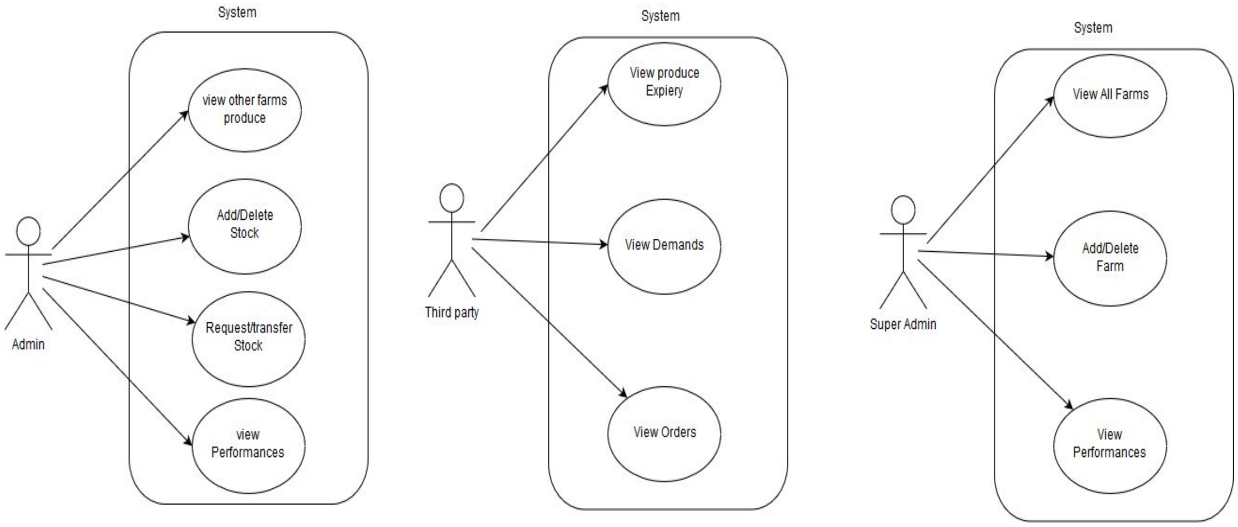


Fig 2: Use Case Diagram (UCD)

4. System Design

The gathered and analysed requirements from the previous chapters provides a good basis to proceed with the designing of the proposed solution. In this section we will transform the analysed requirements into design models that describe the details of the data structures, system architecture, interfaces, and components necessary to implement the proposed solution.

4.1 Data Design

From the requirements gathered it is quite clear that the SMS as well as the Data Analysis (DA) component of the proposed involves some high volumes of data and it is crucial to have some good structures to handle these numerous volumes. The functional requirements highlighted in the URD (*fig 1*) gives a good understanding of the dimension of data that needs to be handled .Data that include the different farms (schools), produces, quantities of produce in stock etc needs to be handled effectively and efficient data designing is required to produce a good product of the proposed solution. Fig 3 below show the dimension of data flow from the major players that will the involved in the proposed solution

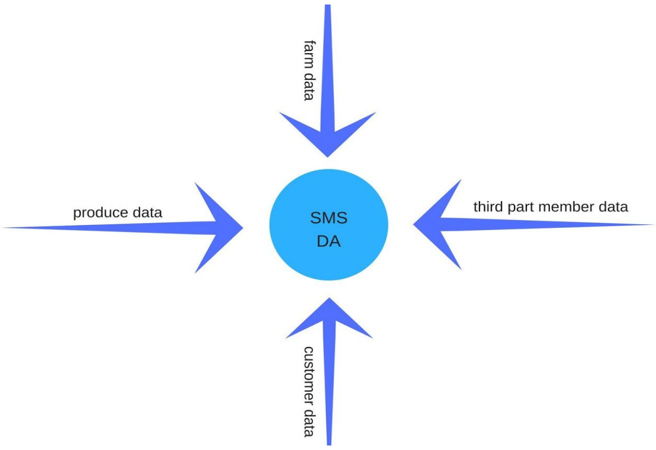


Fig 3: Data Flow Diagram

4.2 Data Analysis Design

The Data Analysis part of the project follows 4 stages that begin with the data being received from different parties that will interact with the proposed system. The data that is generated by the 4 parties highlighted in the data design section is stored in a database. The database is used as a data warehouse medium that stores all the important data relating to the parties.

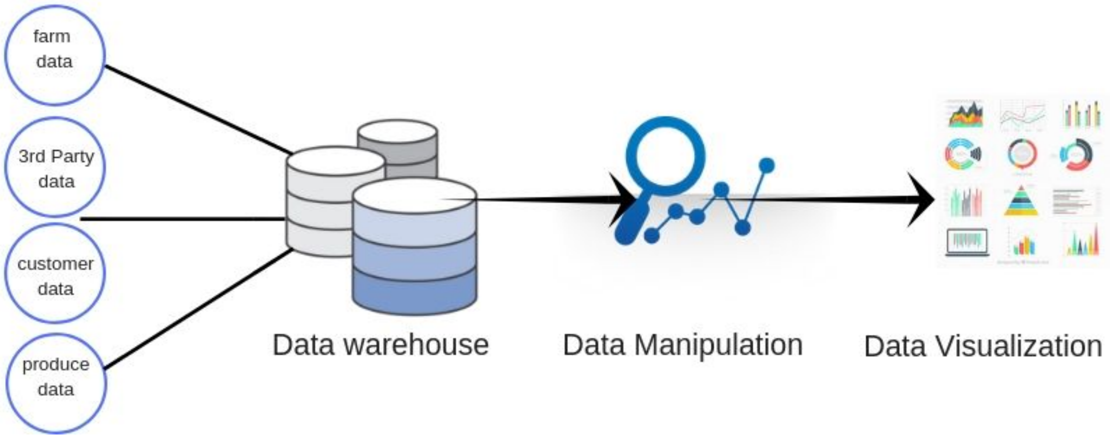


Fig 4: Data Analysis Implementation Diagram

Data that is stored in the database will be retrieved and analysis will be performed on the data. At the data analysis stage *Fig 4*, pandas will be used to carry out some data manipulation and analysis. Pandas is an open source, easy to use python library that is used to perform data manipulation and analysis.

4.3 System Architectural Design

The proposed system follows quite a simple architectural design that allows the system to function as expected. The system will be hosted in the cloud where the application will have access to some crucial tools that are offered in the cloud. In the Fig 4 below the user interacts with a device which acts as a client allowing the user to have access to the application that is hosted on the cloud

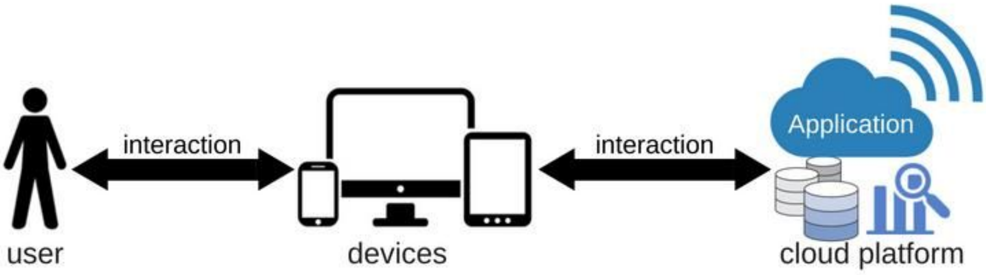
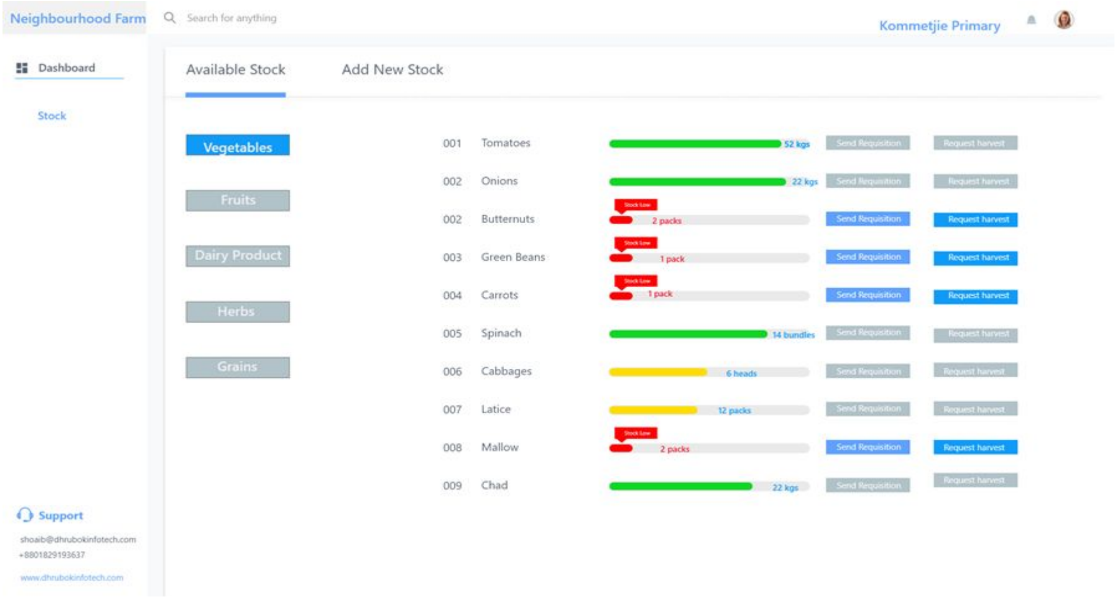


Fig 5: Architectural Design of the proposed System

5. Prototype Designing

In this chapter we will develop a typical prototype of the proposed system based on the designs covered in the Design section to depict the concepts, design alternatives, and screen layouts. For the prototype, only the main functionalities of the proposed system are considered and tools such as database and analytical tools are not implemented in the backend.



001	Tomatoes	52 kgs	Send Requisition	Request harvest
002	Onions	22 kgs	Send Requisition	Request harvest
002	Butternuts	Stock Low 2 packs	Send Requisition	Request harvest
003	Green Beans	Stock Low 1 pack	Send Requisition	Request harvest
004	Carrots	Stock Low 1 pack	Send Requisition	Request harvest
005	Spinach	14 bundles	Send Requisition	Request harvest
006	Cabbages	6 heads	Send Requisition	Request harvest
007	Latice	12 packs	Send Requisition	Request harvest
008	Mallow	Stock Low 2 packs	Send Requisition	Request harvest
009	Chad	22 kgs	Send Requisition	Request harvest

shoaib@dhrubokinfotech.com
+8801829193637
www.dhrubokinfotech.com

Fig 7: stocked Produce View

The stock tab allows the user to view the stock level of the produces. The stock will be categorized in different types such as vegetables, fruits, herbs etc. and upon selecting a certain type a list of the produces that falls under the category will be displayed with their respective available quantities. The stock gives the user an option to request or send stock from and to other farms respectively as highlighted in *Fig 7*.

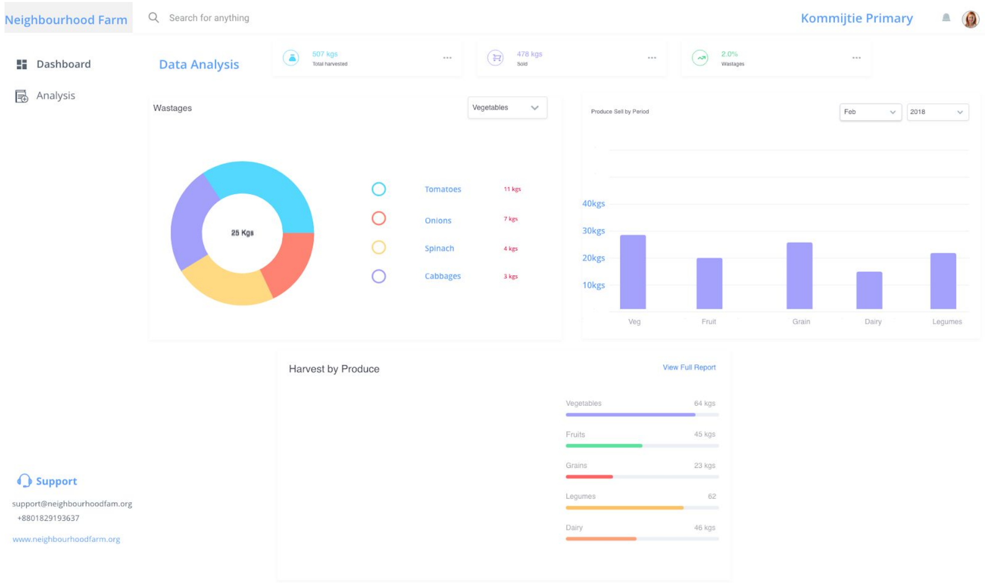
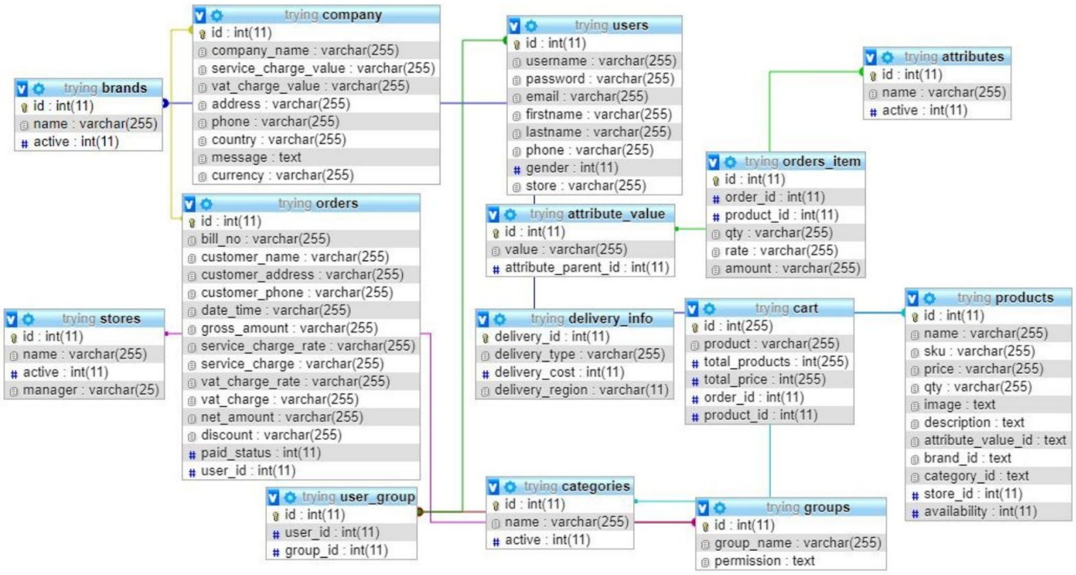


Fig 8: Display of Analysis

The Analysis tab allows the user to view some analysis displayed in graphs and tables. The interface displays tables and graphs with general statistics on wastages, demand and harvest.

6. High Level Design



trying company
id : int(11)
company_name : varchar(255)
service_charge_value : varchar(255)
vat_charge_value : varchar(255)
address : varchar(255)
phone : varchar(255)
country : varchar(255)
message : text
currency : varchar(255)

trying brands
id : int(11)
name : varchar(255)
active : int(11)

trying users
id : int(11)
username : varchar(255)
password : varchar(255)
email : varchar(255)
firstname : varchar(255)
lastname : varchar(255)
phone : varchar(255)
gender : int(11)
store : varchar(255)

trying attributes
id : int(11)
name : varchar(255)
active : int(11)

trying orders_item
id : int(11)
order_id : int(11)
product_id : int(11)
qty : varchar(255)
rate : varchar(255)
amount : varchar(255)

trying orders
id : int(11)
bill_no : varchar(255)
customer_name : varchar(255)
customer_address : varchar(255)
customer_phone : varchar(255)
date_time : varchar(255)
gross_amount : varchar(255)
service_charge_rate : varchar(255)
service_charge : varchar(255)
vat_charge_rate : varchar(255)
vat_charge : varchar(255)
net_amount : varchar(255)
discount : varchar(255)
paid_status : int(11)
user_id : int(11)

trying attribute_value
id : int(11)
value : varchar(255)
attribute_parent_id : int(11)

trying stores
id : int(11)
name : varchar(255)
active : int(11)
manager : varchar(25)

trying delivery_info
delivery_id : int(11)
delivery_type : varchar(255)
delivery_cost : int(11)
delivery_region : varchar(11)

trying cart
id : int(255)
product : varchar(255)
total_products : int(255)
total_price : int(255)
order_id : int(11)
product_id : int(11)

trying products
id : int(11)
name : varchar(255)
sku : varchar(255)
price : varchar(255)
qty : varchar(255)
image : text
description : text
attribute_value_id : text
brand_id : text
category_id : text
store_id : int(11)
availability : int(11)

trying user_group
id : int(11)
user_id : int(11)
group_id : int(11)

trying categories
id : int(11)
name : varchar(255)
active : int(11)

trying groups
id : int(11)
group_name : varchar(255)
permission : text

Fig 9: Entity Relation Diagram (ERD)

The high level design of the backend database can be represented by *Fig 9 above.* The design of the database is structured to represent both the backend and frontend entities. The above ERD only describes all the major entities in the database and their relationship with each other .

7. Low Level Design

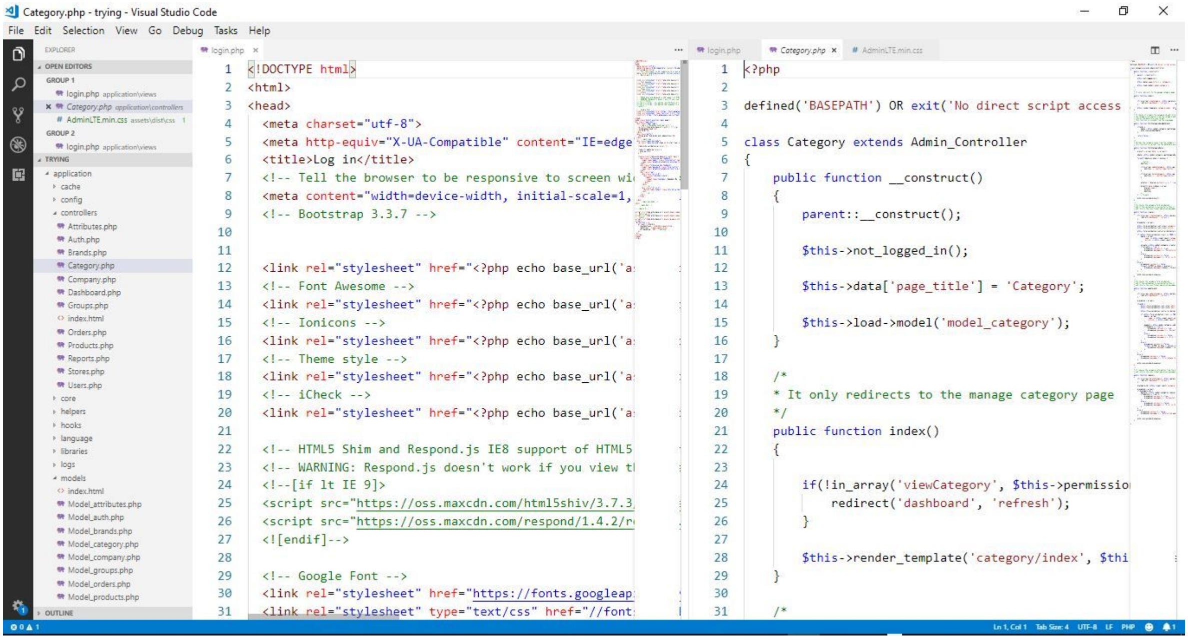
Table 2: Class Description Table

Component	Name of Classes	Description
Controllers	Auth Categories Suppliers Dashboard Orders Reports Stores Products Users	A controllers are a means by which the user interacts with the application. A controller accepts input from the user and instructs the model and view to perform actions based on that input. In effect, the controller is responsible for mapping end-user action to application response. The controller translates interactions with the view into actions to be performed by the model.
Models	Model_Auth Model_Categories Model_Suppliers Model_Dashboard Model_Orders Model_Reports Model_Stores Model_Products Model_Users	An object representing data or even activity. The model manages the behavior and data of the application domain, responds to requests for information about its state and responds to instructions to change state.The model is the piece that represents the state and low-level behavior of the component.
Views	Auth Categories Suppliers Dashboard Orders Reports Stores Products Users	Views are responsible for some form of visualisation of the state of the model. The views manages the graphical and/or textual output to the portion of the bitmapped display that is allocated to its application. The views renders the contents of a model. It accesses enterprise data through the model and specifies how that data should be presented. The views are responsible for mapping graphics onto a device. Views typically has a one to one correspondence with a display surface and knows how to render to it. Views attaches to a model and renders its contents to the display surface.

8. Implementation

For the implementation stage of this project, we opted for the quite a number of tools to write the code and followed the MVC design pattern to structure the application.

8.1 *Code*



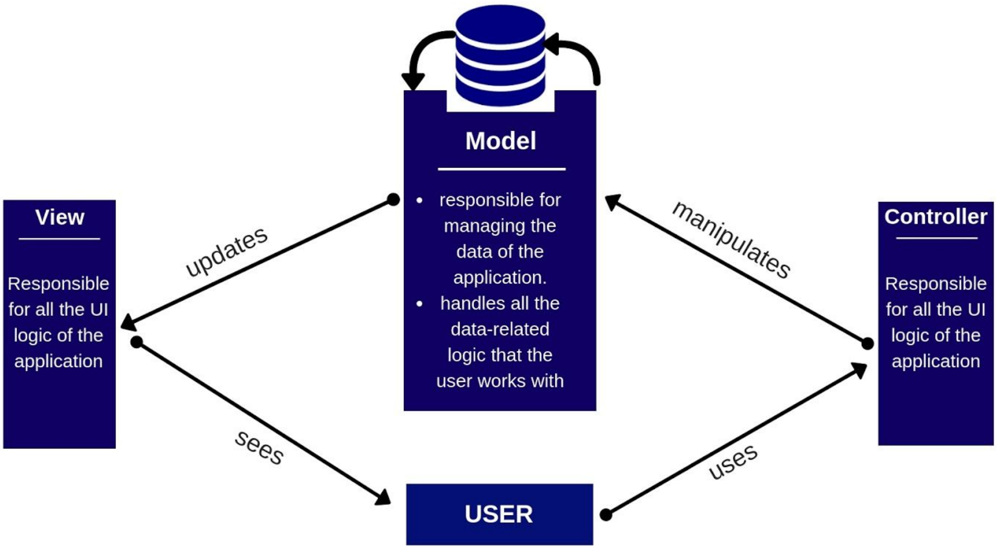
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge
  <title>Log in</title>
  <!-- Tell the browser to be responsive to screen wi
  <meta content="width=device-width, initial-scale=1,
  <!-- Bootstrap 3.3.7 -->


  <link rel="stylesheet" href="<?php echo base_url('a
  <!-- Font Awesome -->
  <link rel="stylesheet" href="<?php echo base_url('a
  <!-- Ionicons -->
  <link rel="stylesheet" href="<?php echo base_url('a
  <!-- Theme style -->
  <link rel="stylesheet" href="<?php echo base_url('a
  <!-- iCheck -->
  <link rel="stylesheet" href="<?php echo base_url('a

  <!-- HTML5 Shim and Respond.js IE8 support of HTML5
  <!-- WARNING: Respond.js doesn't work if you view t
  <!--[if lt IE 9]>
  <script src="https://oss.maxcdn.com/html5shiv/3.7.3
  <script src="https://oss.maxcdn.com/respond/1.4.2/r
  <![endif]-->

  <!-- Google Font -->
  <link rel="stylesheet" href="https://fonts.googleap
  <link rel="stylesheet" type="text/css" href="//font
```

```
<?php

defined('BASEPATH') OR exit('No direct script access

class Category extends Admin_Controller
{
    public function __construct()
    {
        parent::__construct();

        $this->not_logged_in();

        $this->data['page_title'] = 'Category';

        $this->load->model('model_category');
    }

    /*
    * It only redirects to the manage category page
    */
    public function index()
    {

        if(!in_array('viewCategory', $this->permissio
            redirect('dashboard', 'refresh');
        }

        $this->render_template('category/index', $thi
    }

    /*
```

Fig 10:Code Snippet

The above fig shows a snippet of theapliction's code built during the implementation stage.

8.2 *Architectural Design pattern*



Model
responsible for managing the data of the application.
handles all the data-related logic that the user works with
View
Responsible for all the UI logic of the application
Controller
Responsible for all the UI logic of the application
USER
updates
manipulates
sees
uses

Fig 10: MVC architectural Design pattern

The MVC paradigm is a way of breaking an application, or even just a piece of an application's interface, into three parts: the model, the view, and the controller [7]. MVC was originally developed to map the traditional input, processing, output roles into the graphical user interface realm. The classes that are described in the Low Level Design section *Table 2* are organised in the MVC design pattern.

8.3 *Frontend and Backend Integration*

As the application is designed in a two faced format (frontend-backend) , the main goal after successfully completing the design of these two ends is to integrate and connect them .The front end of the application is developed using the ionic framework as a separate project from that of the backend.

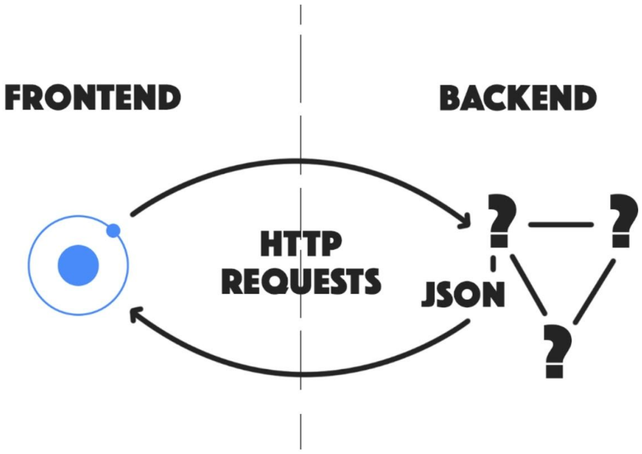


Fig 11: App Integration (frontend-backend)

To complete the connection of the frontend and the backend, an API (Application Programming Interface) need to be implemented that will allow the frontend application to interact with the backend structure through some HTTP request. The app can request data through http request and the backend will respond with json file with the data.

8.4 *Tools Used*

Quite a number of tools has been used in this implementation stage. Php has been used for the server-side scripting. SQL a domain-specific language is also considered in the design and management of the database. HTML and CSS is also used for structuring the web pages and styling respectively. We used the php CodeIgniter framework for the development of the application. To create a local server, xampp was also used.

9. Future Work

Many different functionalities have not been completed and have been left for the future due to a lack of time. There is a need to relook and consider the issue of assigning a shelf life to perishable produce when being captured to the system. Also there is motivation to allow different quantity units to different produces. The Data Analysis part of the project is also another area that needs much attention in the future. The next phrase will involve conducting different testing methods to the application.

References

[1] Food and Agriculture Organisation of the United Nations. The State of Food Insecurity in the World 2011. Food and Agriculture Organisation; Rome, Italy: 2011. p. 51.

[2] United Nations World Population Prospects The 2017 Revision. Key Findings and Advance Tables. ESA/P/WP/248

[3] J. Bonello, Neighbourhood Farm, Neighbourhood Farm, 2018. [Online]. Available: http://neighbourhoodfarm.org. [Accessed: 03- May- 2018].

[4] Katengeza, S.P.; Mangisoni, J.H. and Okello, J.J. The Role of ICT-based Market Information Services in Spatial Food Market Integration: The Case of Malawi Agricultural Commodity Exchange, Contributed Paper presented at the Joint 3rd African Association of Agricultural Economists (AAAE), September 19-23, 2010.

[5] Bracha Gal, Yoav Gal, Ehud Gelb. Adoption of Information Technology for Farm Management, A Case Study. International Farm Management Congress 20013

[6] "Organisation", Open Food Network, 2018. [Online]. Available: https://openfoodnetwork.org/about/organisation/. [Accessed: 14- Oct- 2018].

[7] T. Marston, "The Model-View-Controller (MVC) Design Pattern for PHP", Tonymarston.net, 2018. [Online]. Available: https://www.tonymarston.net/php-mysql/model-view-controller.html. [Accessed: 14- Oct- 2018].