Lesotho Food Security Barometer

A final year project submitted by

Kamohelo Francis Makaaka 2017

Kabelo Qhobosheane 201701374

In partial fulfilment to the requirements for

B.Sc. Computer Science

To the

Department of Mathematics and Computer Science

Faculty of Science and Technology

National University of Lesotho.

June 2021

Supervisor: Dr. Lerato Lerato

Acknowledgements

We would like to pass our heartfelt gratitude to our supervisor Dr Lerato Lerato for providing us with the wonderful opportunity to work on the Food security Barometer Project. Not only did he not fully support us throughout but also, he clarified most contents of the project that were bizarre to us. The project aided us in conducting extensive research and learning a great deal about food security and writing projects. Moreover, we would like to thank our colleagues for helping us test and improve the system Lastly, we would also like to express our gratitude to our friends and families for their continued support and encouragement.

Abstract

The branch of Food security called food availability is an important indicator that countries use to determine if there is enough food available to feed their nation. Currently Lesotho lacks a mechanism that is able to continuously monitor food security so that it can be easy to pick up when there is a food shortage. This study is aimed at designing a dashboard which periodically displays food situations in Lesotho with the aim of helping the government officials to promptly take precautionary measures in case of food shortages.

Table of Contents

Table of Figures

Chapter 1 Introduction

Problem Statement

Currently, the Government of Lesotho through the ministry of Agriculture and Food Security collects its food security reports from various organizations, such as the Lesotho Vulnerability Assessment Committee (LVAC), the Southern Africa Regional Vulnerability Assessment and Analysis (SADC RVAA) , Agricultural Census, World Bank (WB) and the Bureau of Statistics (BOS).The data collected from such organizations is used for planning future policies and budgets to be carried out by the government to ensure that Basotho are food secure. The problem is that, if one needs to make a quick decision regarding the collected data such as finding if there is enough food available to feed the population, or maybe wants to find out the trends in the consumption of a certain food type in a particular district then he/she has to search for such records manually, which can be time consuming, moreover, since such data is usually in numeric form, it can be hard to recognize discernible differences in the patterns or trends among numbers [1]. In addition, such data is not always accessible and available in real time, which can at times lead to delayed decision making.

Proposed Solution

We developed a system for collecting and monitoring data on foods available across the different districts of Lesotho so as to offer real time access to such data using a dashboard. The use of dashboards to display the data can simply be described in terms of a car dashboard, a car dashboard gives the driver a general view of the current car status through a combination of visual gauges (speedometer, oil level) and display (oil pressure warning light, engine warning light) so as to help him know if he is driving above the prescribed speed limit or if there is an oil shortage in the car. Therefore, our dashboard works in a similar way by offering visual charts of the collected data, and it quickly sends notifications to respective officials if ever there is a shortage of a certain food types thus improving the time on decision making. It provides the overall status of food in the country. Data offered in visual form is easier to interpret than one in numeric format.

Objectives

- Develop an online system that is used for collecting food Available in a given amount of time

- Develop a dashboard to report food status in the country
- To setup the platform for data collection module, server environment and dashboard environment
- To design the system architecture including major functionalities
- To test the reliability of the system

Aim

To design a dashboard which periodically displays food situation in Lesotho with the aim of helping the Government to promptly take precautionary measures in case of food shortages.

Overview of how the project was carried out

We carried out the project using Agile methodology, specifically the scrum methodology and extreme programing, in scrum, the requirements are defined, then some of them selected, then carried out in short periods of time called sprints which produce a result, and continuous changes are done to the project until the required requirements have all been met. Extreme programming was used to offer increased performance to the team.

Firstly, we defined the system's requirements as much as possible. This was done by interviewing people from the Agricultural department at the NUL campus as well as the ministry of food security. Then we implemented the features the users had requested, then we returned to users for testing, then refine the requirements, this was done until a satisfactory product was achieved.

Overview of the rest of the rest of the chapters in the report

The rest of the report is setup as follows.

Chapter 2- Literature Review

Chapter 3- Requirements, Dashboard Design Theories and Practices

Chapter 4- Design and Implementation

Chapter 5- Experiments and Results

Chapter 6- Conclusion and Recommendations

Chapter 2 Literature Review

Introduction

This chapter will outline what food security is, its related matters, why food security is important, what hinders food security. What issues arise when the country is not food secure. Concepts on how to measure food security and tools involved in four pillars of food security. Finally, the chapter will explain what food insecurity is, and ways it can be improved. Thereby concluding as to how our project will work in improving food security in the country.

Background Information

The agricultural year in Lesotho begins on the 1st of August and ends on the 31st July of every following year [1]. In Lesotho, there are four geographic zones and these zones mark the ecological and climatological differences. The zones are as follows: Foothills, Lowlands, Mountain and Senqu (Orange) River Valley. Food security is measured more especially on the food produce of the country. A number of food security pillars exist they are: availability, accessibility, utilization and stability but we focused on the availability part of food security pillars.

Lesotho does not currently have a food security barometer system and reports for food security are done on reports produced by the Lesotho Vulnerability Assessment Committee (LVAC), Agricultural Census, the Southern Africa Regional Vulnerability Assessment and Analysis (SADC RVAA), the Bureau of Statistics (BOS), Food and Agriculture Organization (FAO) and the World Bank just to name a few. The proposed system seeks to aid the minister or appropriate government official with compiling the data collected from the various organizations and then displaying it in a summarized way by the use of a dashboard.

Related Work

Currently there exists many solutions to the proposed problem of Lesotho not having a system that will alert the policymakers or appropriate body on food crisis or shortages of the overall food in the country. There is a food security portal that is facilitated by International Food Policy Research Institute (IFPRI). Although the food security portal provides precise and timely data on food security, it however provides information of food security indicators at an overview level of certain regions and countries. It does not fully show all the necessary details about a certain country’s food security is. The link is here: https://www.foodsecurityportal.org Our proposed system will used by elected district data collectors and the responsible body for making decisions on food security in the country.

Food Systems Dashboard - the food systems dashboard combines data from multiple sources to give users a complete view of food systems using interactive data visualizations. It enables comparison of food systems around the world. The system works good in general; the only problem is that it does not have up to date information of some of Lesotho food security indicators, the last data to be recorded was in 2018. The link to the side:
https://foodsystemsdashboard.org/compareandanalyze
Sudan food security dashboard – a dashboard that focuses on presenting food security levels in Sudan using the defined pillars of food security namely availability, accessibility, utilization and stability. The link is here: http://fsis.sd/SD/EN/DashBoard/Indicators/

There is also another solution to the problem, feed the future which is a website that brings together partners from across various sectors and the U.S [1]to try to fight poverty. The link is here: https://www.feedthefuture.gov This solution also does not have most southern countries with Lesotho included. Our system will address our own country and therefore aid the government.

In addition to existing solutions to food security issues there also exists FSIN- Food Security Information Network, this is a technical platform for exchanging expertise and best practices on food security and nutrition analysis. It aims to combine information of different departments and partners in order to inform and try to eradicate poverty as much as it can. More information about the site is found on the link here: https://www.fsinplatform.org

Our proposed Food Security Barometer for Lesotho with provide accurate and timely data on food produce collected from different districts by the hired district data collectors that will be passed to the local server or local database of the government and display a customized dashboard for easier understanding the status of food in the country. Dashboards provide data in a simplified manner that can enforce decision making quickly. This ability to provide dashboard data is not provided by the above FSIN.

Research

We visited Ministry of Agriculture and Food Security at Maseru asked the stuff major questions in order to understand what food security is. The national food security policy that is published by the government and linked to the national vision 2020 and the poverty reduction strategy as well as the global millennium development goals guides the country's efforts to achieve sustained food security. Although the policy shows that to achieve food security the focus should not only be on production of enough food and other methods, our project therefore focuses on the data collection food production of staple foods in the country.

What is Food Security?

Refers to people having physical and economical access to sufficient, nutritious and culturally appropriate food at all times so that they can effectively use it to meet their dietary needs for a healthy and active life.

2.5.1 Food Security is divided into four pillars namely:

- Availability – there must be enough supply of food to feed the entire population, such food can come from different sources such as fields, gardens, shops or food aid
- Accessibility – everyone must be able to afford the available food that they need; irrespective of their gender, age or membership of a particular group of society
- Stability – food should be available and accessible at all times; their supply should not be interrupted by droughts or pests and diseases
- Effective use – food should be safe and nutritious enough to meet the bodily requirements of each individual.

Staple Food in Lesotho include

These are foods eaten on a regular, routine basis and that they meet the standard diet for a given person in a country.

Corn meal(papa), Bread, Potatoes, Rice, Mutton, Chicken, Eggs, Fish, Beans, Cabbage, Pumpkins, Spinach, Apples, Apricots, Peaches, Pears, Quinces [2].

Food Security Monitoring Organizations

- World Food Programme (WFP), Lesotho Vulnerability Assessment Committee (LVAC), Food and Agricultural Organization (FAO), Southern African Development Community (SADC), World Bank (WB) and Agricultural Consensus.

- We used the following food monitoring tools; Intergraded Food Security Phase Classification (IFSPC) and SADC Regional Vulnerability Assessment and analysis (SADC RVAA).

Food Security Indicators

2.8.1 Utilization

Percentage of population using at least basic drinking water services

It measures the percentage of people using basic water services and safely managed water services. Basic services mean water should come from an improved source e.g., piped water, boreholes, protected springs, given that the water collection round trip should be no more than 30 minutes [3].

Percentage of population using at least basic sanitation services

It measures the percentage of the people using improved basic sanitation services and safely managed sanitation services that are not shared by other households. The improved sanitation services include septic tanks, pit latrines and flush/pour toilets.

Percentage of population using safely managed drinking water services

This indicator measures the percentage of people consuming drinking water that comes from an improved source that is accessible, available and not contaminated. Improved water sources include boreholes, protected springs, protected dug wells, piped water and packaged water.

Percentage of population using safely managed sanitation services

It measures the percentage of people using improved sanitation facilities that are not shared with other households and excreta are safely disposed of in situ or transported and treated offsite. Improved sanitation facilities include flush/pour toilets, septic tanks or pit latrines [4].

Percentage of population under 5 years of age affected by wasting

It measures the percent of children aged 0-59 months who experience a loss of body weight in relation to height and it could be moderate or severe. Moderate means weight for height below 2 standard deviations of the WHO Child Standards median, while severe means weight for height below 3 standard deviations of the WHO Child Growth Standards median.

Percentage of population under 5 years of age who are stunted

This indicator measures the percentage of the children aged 0-59 months who have a low height for their age and it can be moderate or severe. Moderate means height for age below 2 standard deviations of the WHO Child Standards median, while severe means height for age below 3 standard deviations of the WHO Child Growth Standards median.

Prevalence of low birthweight

The indicator measures the total number of live births that weigh less than 2,500 grams in a given time-period divided by the total number of live births in the same time-period.

2.8.2 Availability

Dietary diversity

Refers to the number of different foods or food groups consumed over a given reference period. It can be measured either at household or individual level and by assessing the variety of different food groups consumed in a specific recall period. Household level measures include Household dietary diversity score (HDDS) and Food consumption score (FCS). Individual level measures include Minimum acceptable diet (MAD) and Minimum dietary diversity (MDD). Higher scores represent a more diverse diet.

Average value of food production is classified into the following;

Crop production

This indicator deals with growing crops for use as food and fiber.

Vegetable production

This indicator deals estimates of vegetables grown per household.

Livestock production

This indicator shows the number of domesticated animals to produce food, fiber and labor.

Cereal production

It shows the quantity of cereal produced for animal and human consumption.

Distribution of Consumption of Vitamin A, protein and iron rich foods

It shows the household intake of iron rich foods, protein and vitamin A and nutritional outcomes.

2.8.3 Accessibility

Gross Domestic Product per capita (in purchasing power equivalent)

It measures the average level of national income per person. It is found by dividing the gross domestic product of the nation by the total population

Percentage of population under 5 years of age who are underweight

it measures the percent of children aged 0-59 months whose body weight is considered to be too low for their age. It measures weight for age in below 2 standard deviations of the WHO Child Growth Standards median.

Prevalence of moderate or severe food insecurity in the total population

it measures the share of the population who lack regular access to enough, safe and nutritious food that help promote an active and healthy life.

Access to land

It measures the people's ability to access water and other resources, as well as basic services such as sanitation.

2.8.4 Stability

Cereal import dependency ratio

It measures how much of the available domestic food supply of cereals has been imported and how much comes from the country's own production.

Value of food imports over total merchandise exports

It measures the variations of the per capita food production across countries and time.

What is Food Insecurity?

When people are unable to consistently access or afford adequate food necessary for a healthy and active lifestyle due to lack of money or other resources such as land.

2.9.1 Causes of food insecurity

a) The financialization of food

Food, although a basic human need, is being traded as a commodity on international markets by speculators who have no direct interest in or need for the foodstuffs they trade, speculation makes food prices volatile and price hikes can cause severe and/or prolonged food crises.

b) Market dominance of multinational agribusinesses and exporters

Large corporations take up majority of the market while small scale farmers have limited avenues to market their produce. This can cause the small-scale farmers to sell their produce at unfair prices.

c) Unfair trade rules

Large companies do business with local food producers in a way that benefits them while being unfair to farmers and at times may not reward them for their labor or produce.

d) Lack of access to farming land

People need land to produce food as well as generate income. But many people simply do not have resources or an opportunity to own land.

e) Natural disasters and climate change

Natural disasters such as floods, hurricanes, drought can wipe out entire harvests or kill livestock, leaving a devastating on people who rely on their harvests and livestock for daily food.

f) Conflicts

Conflicts and wars can have a great impact on food production and supplies. Conflict linked food shortages can trigger years of food crises, even after the fighting has ended.

g) Inefficient use of water, crop rotations and fertilizers.

2.9.2 Consequences of food insecurity

- children that come from food insecure households perform worse on literacy, numeracy and short-term memory tests.
- people end up eating a lower quality diet
- people suffer from poor mental health such as depression

- some people can only afford processed food which are energy dense and high in fats and sugars, as a consequence they end up suffering from Obesity and other health related issues.

- An increase in unemployment which can lead to poverty

2.9.3 Ways food security can be improved

1. Optimizing land use: if all the available land, livestock, labor and water resources are brought together in the right way, they will help boost food security in the long term.

2. Prioritizing intensive farming methods: Land whether around a home or in the distant fields can be worked year-round if the right combination of crops and organic farming principles, a family can use whatever little land they own to produce food for themselves as well as sell the surplus.

3. Adopt appropriate farming practices: good farming practices such as planting field crops in areas that are agro-economically suitable, block farming and introduction of improved agricultural technologies can help yield high produce.

4. Develop appropriate irrigation methods: good irrigation methods make the best use of water in conjunction with the land and human resources, as well as other essential inputs (energy, machinery, fertilizers and pest control measures) with the aim of improving crop production. There are various irrigation methods such as sprinkle irrigation and surface irrigation.

5. Reduce food waste: nearly one-third of all food produced in the world is discarded or wasted for various reasons such as food spoilage. People can reduce food wastage if they buy food enough to meet their needs, also storing food correctly can help e.g., bananas need to be separated from other foods in the fridge as it produces ethylene gas which can cause other food to spoil.

6. Incorporate agro-forestry practices into all scales of farming systems: agroforestry is the international integration of trees and shrubs are grown around or among crops or pastureland. This helps promote long term sustainable and renewable forest management, especially for small scale producers

7. Encourage appropriate animal husbandry and fodder production: animal husbandry is concerned with animals that are raised for meat, fiber, milk or other products. fodder refers to food given to animals. Good animal husbandry practices improve the animal's environment, prevents animal stress that can lead to disease and reducing the need for drugs. Fodder production is a cheap and easy way to fulfil the feeding requirements of dairy animals.

Methods and Tools

2.10.1 Software:

1) Operating System: This is the base platform that with it we were able to use other programs and able to work between hardware and software of the system. For Our project we used Windows 10 Pro because of its simplicity and its ability to easily install other necessary software.

2) Xampp: We used Xampp to run our database on the local server of our laptops. With it we are able to Apache server and MySQL database.

3) NodeJS: This tool enables users to write both the client-side and server-side code in JavaScript A programmer can use JavaScript for building both frontend and backend of a web application [5] .

4) PhpStorm: This is the programming Ide that can write in multiple programming languages and the reason we chose this tool is because of its ability to autosave code anytime we program. PhpStorm helps to work around the code more efficiently and save time when working with large projects. Its ability to jump to a method, function or variable definition in just clicks and shortcuts is efficient [6].

5) Composer: This is a dependency manager for the PHP which functions as some sort of project manager to assist the programmer manage dependencies that will be used on a project-to-project basis [5]. This is a php program that works with php and useful for downloading relating php programs.

6) Android Studio: This is an integrated development environment for Google's Android operating system based on IntelliJ IDEA [7].

7) Grafana: is an open-source visualization and analytics software for developing mobile responsive dashboards that allows the user to query, visualize and get alerts on metrics,

no matter their location [8]. It can use multiple data source from multiple database management system such as MySQL or influx DB and convert the data into beautiful dashboard visualizations such as graphs, bar gouges, maps etc., that are easy to interpret

Programming Languages and Frameworks

Programming languages are used to communicate and interact with computers. Programming languages are important for developing programs and systems and a programming language needs to be chosen. They are used to communicate with the machine, by creating programs, applications, scripts, or other set of instructions for computers to execute and control the behavior and to express algorithms [7].

2.11.1 Client-Side

The web programming in this side is related to the user. This side is the front-end side of application development. The users therefore use the web browser to access the information and the web server will execute the script and send information to the user's interface. Client-side scripting is executed and interpreted by browsers; therefore, it is viewable by anyone visiting that side [8]. There several client-side scripting technologies and these include; HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), JavaScript, jQuery, JAVA.

In our system both the web application and android mobile application are in the client-side. All of the above-mentioned languages were used to develop our system in client side.

The web browser will read the documents by interpreting the tags and closing tags of the html and display the content of the web pages. When the user opens the web page the browser fetches the html together with the CSS to control the presentation and layout of the webpage, then the JavaScript will control the behavior of different elements in the web.

2.11.2 Server Side

Server-side writing is used to communicate with long life storage like databases or files. Different medium and technologies are used to communicate with the server and the server is the back-end side of application development. The server renders pages to the client and process the keyboard or any input made by the user. The Server-side scripting means the information needed to create the dynamic page is done on the web server before being sent to the user. This is done using scripting languages such as PHP, ASP.NET and Python [9]. Server-side scripting shows interactive web sites that can store and retrieve data to users and it tries to provide response according to the user's requirement.

For our system we used the following technologies:

- PHP- When the sever receives requests for php pages, the server reads the php files and works on the tasks before giving it to the user. PHP supports different databases (MySQL, Oracle…) and is compatible with Apache web server.
- We used SQL – Structured Query Language to communicate with the MySQL database both from the mobile application and the web application. SQL allows users to create databases, access, insert and manipulate data in the database [10]. Different Relational and non-relational Database Management Systems are available and can be used but for our system we chose MySQL. MySQL is a relational database management system that uses SQL for easier CRUD applications. It performs create, read, update and delete operations efficiently on relational data.
- Movement of data between the android application and the server is done through usage if JSON. JSON is part of JavaScript language. JSON models the data objects that are transferred over the network.
- Web Server- Apache HTTP Server is ability to work on many operating systems and already integrated in the Xampp software. It is also easy to use. Server Requirements; Servers need to be fast with large storage and more RAM to be able to take more data. A Server needs to have permanent internet address (IP - Internet Protocol). If the I.P. address changes, the web site would not be found and will appear offline, the browser will display a **cannot find web site** error [11].
- For the security of our application, we used Laravel middleware. These deny none admins from accessing the admin pages and non-users from accessing the system. Moreover, it denies generally non-authenticated users from accessing the system.
- Laravel (back-end): is an open-source PHP framework that is widely used for web-based or web application development known for its reliability, high performance, finer efficiency, articulated ORM systems, a multiplicity of Authentication options,

expandability, modularity, facilitates caching, secure SQL processing, dedicated SQL builder, restful controller, uncomplicated coding rules, etc. The framework offers a wide ecosystem with immediate deployment, routing, templating, ORM, DB query and listing [9].

Algorithm

Below is the algorithm

The algorithm below shows how the minimum threshold for required values of different food types is acquired for different districts. There is no food type shortage as long as the food available quantity is above the threshold, but once it goes below the threshold then such a district is experiencing a certain food type shortage

$$let\ m_1 = Minimum\ food\ type\ required\ per\ person$$

$P_1 = District\ Population$

T= $Total\ minum\ required\ food\ value\ per\ district$

C= $Current\ Food\ Quantity$

$T = m_1 * P_1 = Minimum\ Food\ Threshold$

We devise a formula for

$$L = \frac{C}{T}$$

$$if\ L > 1,\ there\ is\ no\ shortage,\ the\ status\ is\ good$$

$$else\ if\ L < 1, there\ is\ shortage,\ the\ status\ is\ action$$

$$if\ L = 0\ ,\ food\ type\ not\ available$$

Chapter 3 - Requirements and Analysis

3.1 Introduction

This chapter presents data gathering methods that were utilized for system's requirements, to be specific functional and non-functional requirements, so as to meet the project's scope already defined in chapter 1. Unified Modeling Language (UML) is used to design the system's requirements for its Object-Oriented nature and how simple it is to understand as compared to other data modelling techniques. Moreover, dashboard designs, theories and practices are also clearly illustrated.

3.2 Information gathering techniques

The data was collected from various local and international organizations that record Lesotho food security data such as Bureau of Statistics and Lesotho Vulnerability and Assessment Report (LVAC). More information on the data can be found in the appendix.

3.3 System's requirements

3.3.1 Functional Requirements

	User story	Requirement	Priority
Registration of new user	As a user I want to be able to register into the system	The system must allow registration, verification and storage of new users into the system's database	MUST
Login on phone or computer	As an existing user, I want to be able to login into my account using either computer or mobile phone	The system must allow registered users to log into their account on mobile or computer by entering their email and password.	MUST
Recover password	As an existing user, I want to be able to be able to recover my login password in case I forget it	The system must allow a registered user to recover their password by sending a reset password link to the email the user used to register into the system	MUST
Responsiveness	As a user, I want to be able to use the web application in different view modes	The system must be responsive to accommodate different views offered to the user	MUST

Insert, view and edit records	As an existing user, I want to collect, view and edit food security data of my respective district	The system must allow respective users to insert, store, modify and update data related to food security conditions of their districts	MUST
View Overall Food Security Status	As an existing user, I want to be able to view visual data on food security	The system must allow registered users to view dashboards that contains visual representations of data collected of Food security	MUST
View Specific food group data	As an existing user, I want to be able to view data on various foods that belong to the same food group	The system must allow users to view data that is related to a certain food group	MUST
Send food security alerts to respective bodies	As an existing user, I want to receive real time updates if food security levels get low	The system must send e-mail alerts to respective authorities if food availability levels drop below a certain threshold.	MUST

3.3.2 Non-Functional Requirements

- both the mobile and web application should be easy to navigate through
- Only the administrator can delete a user from the system
- Only the administrator can add the weights of different food types
- It should be easy to change the requirements of the system
- The system should always be available for dashboard viewing and registering of data
- The capacity for users stored in the database will not be problem as about close to 10 data district users will be registered each recording data for their respective districts
- The data collector in one district should not be able to enter food security data of other districts
- The system should only upload data to the database only when all the fields have been filled in correctly
- The system should view up to date information on the dashboard

3.4 System's Requirements modelling

Models aim to provide a high level and simple description of the system. In this project, various diagrams are used including use case diagrams, object relationship diagram, and use case descriptions

3.4.1 Use case modelling

The use case diagram is used to model the system's functionality and requirements by using actors and use cases. The system's use case is given below in fig 1.1

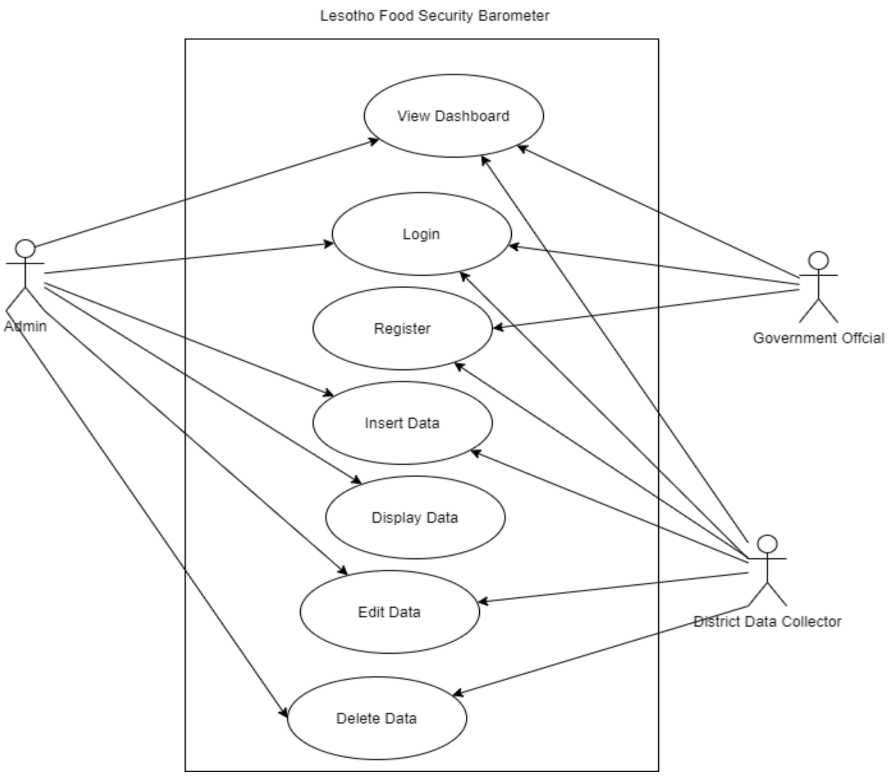


Figure 1-Use Case Diagram

Use case	Brief description
View dashboard	A registered user can be able to view the system's food data dashboard
Login	Registered users should be able to login into the system using their email and password.

Register	All the actors will be able to register into the system and the admin approves registration of other actors.
Insert data	The admin and the District Data collector are the only ones that have to authority to insert food security data into the system
Display data	The admin and District Data Collector are the only ones who can access the system’s food security data using the mobile or web app
Edit data	The admin and District Data Collector are the only ones who can edit the system’s food security data using the mobile or web app
Delete data	The admin and District Data Collector are the only ones who can delete the system’s food security data using the mobile or web app

3.5 Dashboard design theories and practices

3.5.1 Benefits of dashboards

Dashboards provide a high-level visual presentation of performance without entangling the user into a mesh of numbers and figures. They also offer the following benefits:

- Optimization of the decision-making process – Dashboards consolidate reporting into one location. This saves time compared to running multiple reports, moreover, it becomes possible to make decisions in an agile an efficient manner, drastically reducing chances of an errors.
- Optimizing time and resources - The use of dashboards aids in the improvement of daily routines. It reduces the time spent on manual and complex operations like updating

spreadsheets and reports, allowing those resources to be devoted to the most important tasks.

- Intuitive data presentation – dashboards do not require lengthy and complicated training to use. Dashboards are designed with the user in mind. The graphic representation allows for an easy navigation throughout the information
- Mobile device accessible – the majority of dashboard software is designed to work on any mobile device. Making it easy to access key metrics anytime, anywhere
- Monitoring and oversight - dashboards can deliver customized information and analytical capabilities to each group of users based on their job.

3.5.2 Dashboard design principles

- All essential information should be immediately accessible within 5 seconds – when designing a dashboard, stick to the five second rule: this is the amount of time the relevant stakeholder should spend analyzing the dashboard to get the information they need [13]
- Logical Layout: the inverted pyramid- The most important insights should be displayed at the top of the dashboard, with trends in the middle and granular data at the bottom [14]. That is, keep the most important and high-level insights at the top, the trends which give context to these insights and greater granularity details at the bottom that need more studying
- Each dashboard should contain no more than 5-9 visualizations – It may be tempting to cram as many details into a single dashboard but this can be counter intuitive as cognitive psychology says that a human brain can only comprehend around 7+-2 images in one-time, visual clutter can be reduced by simply breaking a dashboard into two or more separate dashboards or use filters
- Keep the design simple and easy to understand– make it as easy as possible for users to be able to analyze the information on the dashboard

- Select the right type of data visualization – Avoid cluttering the screen with a variety of charts and tables when choosing visual options. Maintain a uniform look throughout the dashboard by 3.5.3 Existing food security dashboards

There is a variety of food security dashboards that offer different functionalities and their examples are given below:

- Food systems dashboard – the food systems dashboard combines data from multiple sources to give users a complete view of food systems using interactive data visualizations [14]. It enables comparison of food systems around the world

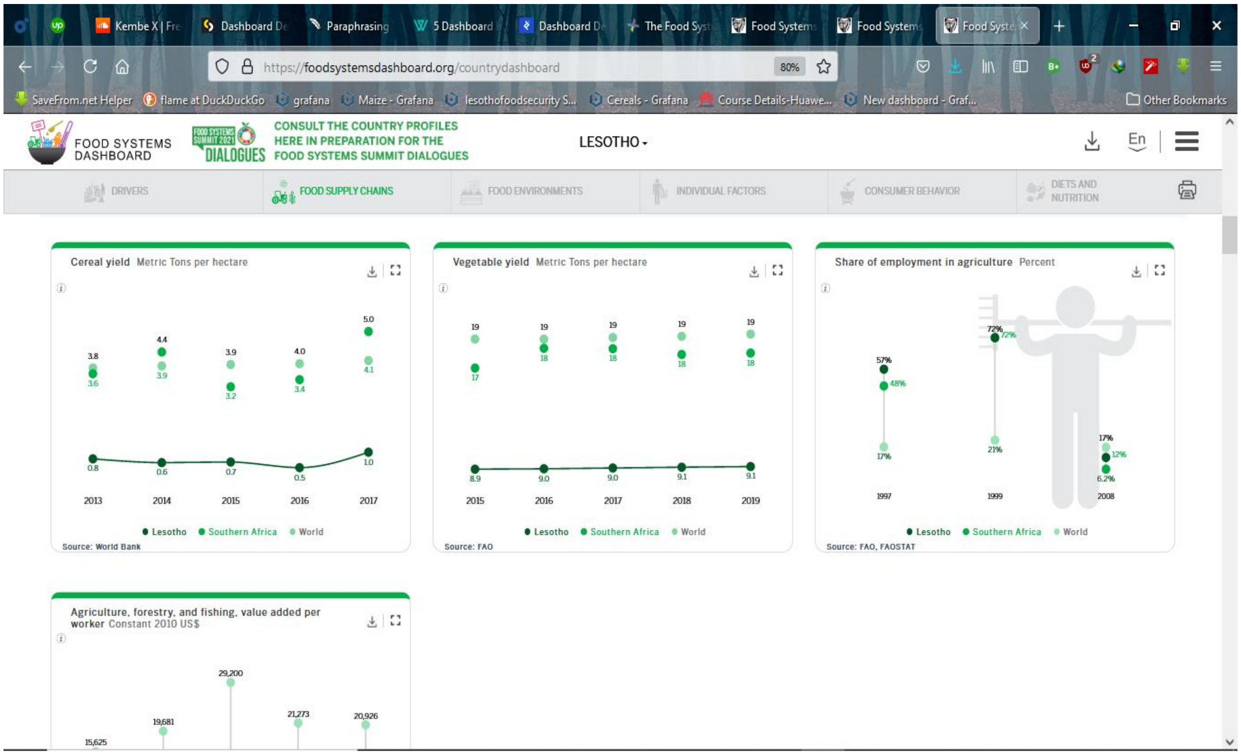
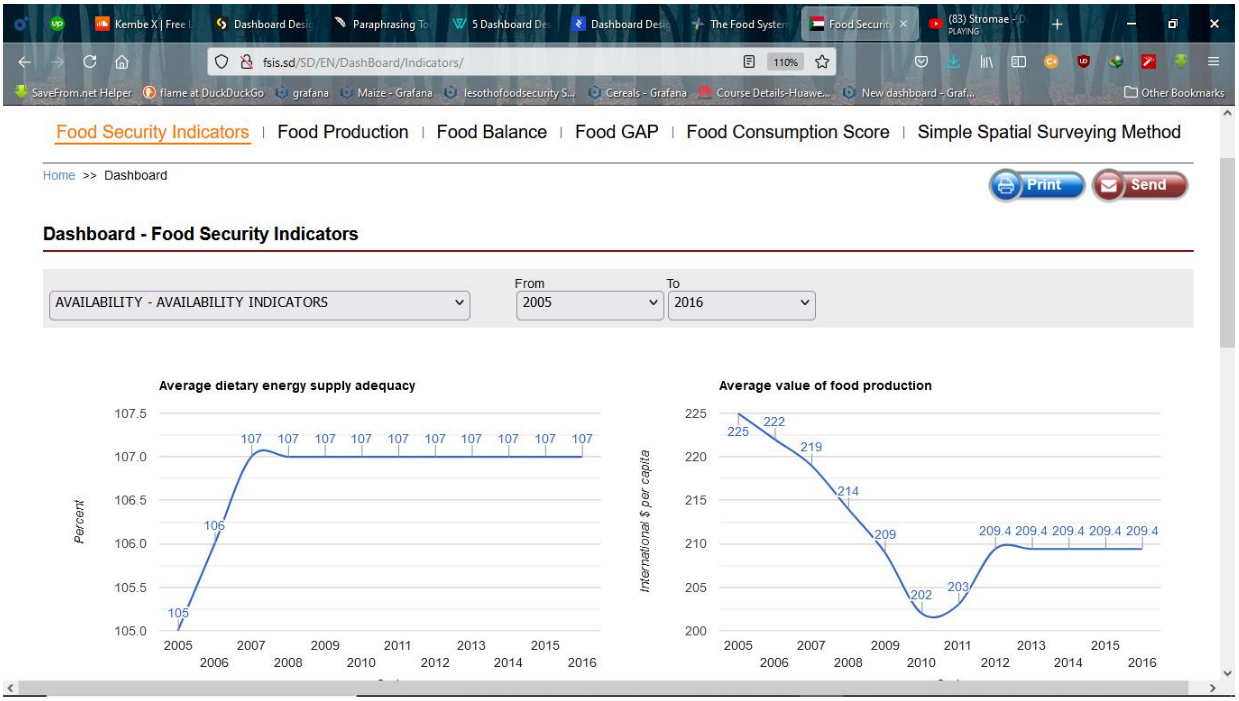


Figure 2- Dashboard of Food Systems Dashboard

- Sudan food security dashboard – a dashboard that focuses on presenting food security levels in Sudan using the defined pillars of food security namely availability, accessibility, utilization and stability [15].



Home >> Dashboard

Dashboard - Food Security Indicators

AVAILABILITY - AVAILABILITY INDICATORS

From: 2005

To: 2016

Figure 3-Dashboard of Food Security Indicators

Chapter 4 – Implementation and design

While chapter 3 focused on the logical model of the system, this chapter focuses on the conceptual as well as the physical design of the system. This includes the architectural design, database design, user interfaced design as well as the physical design of the system, which includes both software and hardware requirements and security in the system.

4.1 Methodology

The system was developed using agile methodologies that integrated the scrum and extreme programming methods. Agile development process generally has the following steps: establishing a few functional requirements, design, development, test, deploy, evaluate, collect feedback, establish new requirements and repeat the process until the final desired product is achieved [10]. The iterative nature of agile methodology allows us to discover the project's features and requirements in an iterative way since we are creating a new system with constantly changing requirements. Agile methodologies, when compared to other traditional methodologies such as waterfall offer improved benefits such as flexibility in that the priorities and requirements can easily be changed throughout the course of the project.

4.2 System design

The general flow of the food security barometer system can be described as follows: a district data collector collects data on the quantity of selected food types available in a given period of time by using either the system's mobile or web application. The data collected by all district data collectors is then stored in the national database, Grafana then uses the data to form graphical reports in the form of dashboards. The dashboards continuously show the current food levels of different food types in a given period of time, this helps any concerned party to be able to study, analyse and monitor food availability levels. The quantity of each food type should not be below a certain defined threshold, if this happens, then Grafana will send a notification to a responsible government official so that they can be able to make quick decision about such a shortage. Fig 4.1 shows the conceptual model of the system

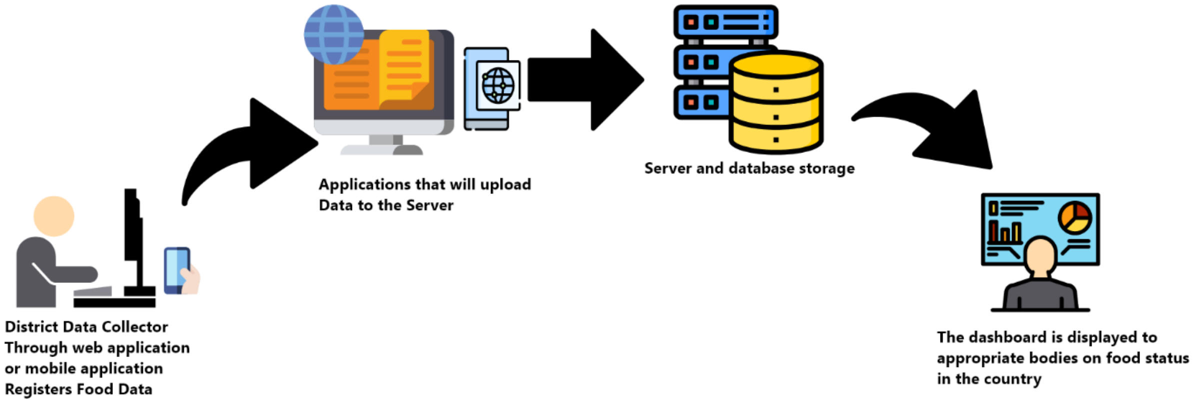

Applications that will upload
Data to the Server
Server and database storage
District Data Collector
Through web application
or mobile application
Registers Food Data
The dashboard is displayed to
appropriate bodies on food status
in the country

Figure 4- Conceptual Design

4.3 The System Architecture

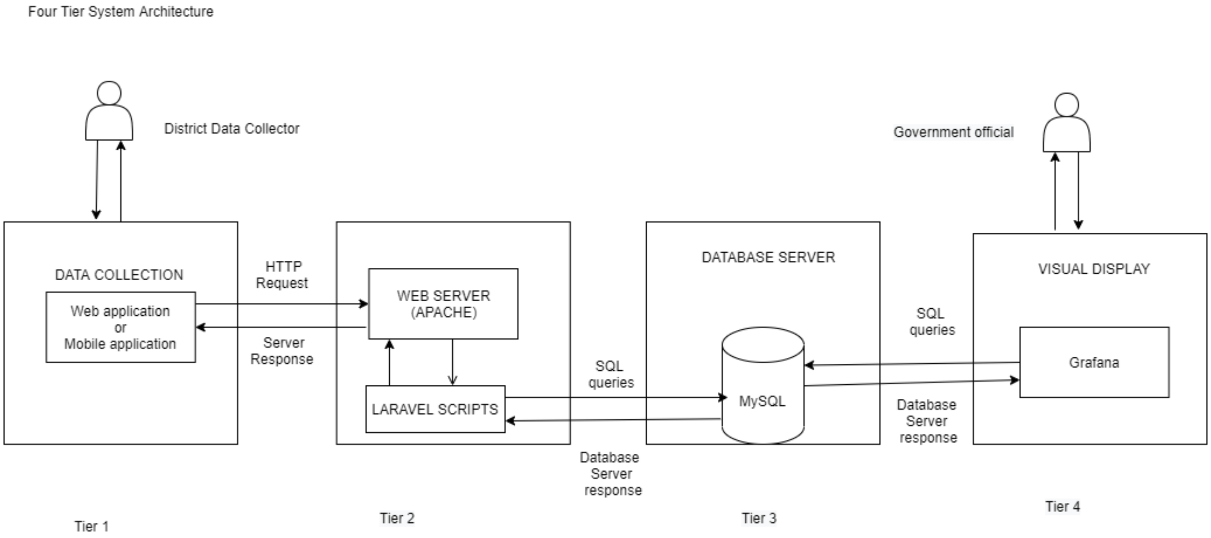


Figure 5- Architectural Design with 4 tiers

Figure 1 show that the Food Security Barometer is divided into four tiers: the user interface in the web or mobile application, the web server, the database server and the visual display on the Grafana application

Tier 1-Data collection

There are two methods of inserting data into the system, either through a mobile or web application and each method along with its design tools and languages will be described in the section that follows.

Mobile application:

Language used: Java

Development tool: Android studio

A mobile application was designed to make it easy for district data collectors to insert food availability data into the system. One benefit of mobile apps is that they work faster than their counterpart web applications [11]. In a mobile application the user interface is made up of a couple of elements, namely Views, View groups and layouts. **View** is the base class for all UI components in android, **view group** is a subset of view and a base class for layouts and a **layout** defines the visual structure for a UI, such as UI for insert cereals activity or application widget [12].

We opted to use android studio instead of other IDE's such as Eclipse because of the following reasons: Android studio is the official IDE for developing android applications, it offers advanced code completion, system stability as it less likely to crash, uses Gradle to build its systems which is faster and more stable than Eclipse's Apache Ant.

Android applications can also be written using Kotlin but we opted for Java as Kotlin has a steep learning curve, is constantly being updated meaning the developer always need to update his libraires and it has a strict syntax. Moreover, java has a wider community support

The frond end of the application was designed using Html and CSs

The backend of the web application was designed using Laravel, which is an open-source PHP web framework that follows the model-view-controller (MVC) architectural pattern.

- Model – structures data in a reliable form and prepares it based on controller’s instructions. Also, it helps the controller to retrieve all of the information it needs from the database.
- View – Displays data to the user in an easy-to-understand format, based on the user’s actions
- Controller - Takes and responds to user commands, sends commands to the model for data updates, sends instructions to view to update interface. It is essentially the brain of the application.

How it works:

- A user requests to view a page by entering the URL
- The controller receives the request
- The controller the uses Models to retrieve all of the necessary data organizes it and sends it off to the
- View, which uses that data to render the final webpage presented to the user in their browser.

The benefit of using frameworks is that they usually have good documentation and community support, moreover frameworks eliminate the need to write repetitive code hence saving the developer time to develop the system [13].They also have a strong level of security as they are tested and developed by different developers. There are a wide variety of frameworks such as Django and Ruby on rails, while they are good alternatives, Laravel is good and fast to learn and deploy.

Tier 2-Web Server

Apache Web Server – is a free and open-source cross platform web server software that delivers web `   content through the internet. It processes requests and serves web assets and content via HTTPS. In general, its job is so establishing a connection between the server and the browser (Mozilla Firefox, Chrome) a client uses so that it can distribute data between them interchangeably.

PHP functions as the open-source middleware that sits between the database server (MySQL) and the Web server (Apache) and provides significant functionality for both [14]. It can work as an Apache integrated module, which provides performance boost to the handling of PHP scripts. PHP works like this: When a request is made for a page with PHP code that has a *.php* extension, Apache hands it over to the PHP scripting engine to handle. Usually, PHP is used to provide access to a database via its code, so the scripting engine is set up to talk to the local database.

An Apache web server is used with the php module to support the back ends in tier 3. The PHP scripts will process user input, the generate SQL queries so as to access the database while the Apache web server will compile and execute the output in dynamically generated HTML pages.

Tier 3-Database Server

MySQL is an open-source relational database management system with a client server model. A database management system is a set of programs that controls the creation, maintenance and use of database and its end users. A database server is a computer system that provides other computers with services related to accessing and retrieving data from a database.

The reason we choose MySQL is that it open source, flexible and quite easy to use, it is also secure

Tier 4- Visual Display

Dashboards

Grafana - a multi-platform open-source analytics and interactive visualization web application. It provides charts, graphs and alerts for the web when connected to supported data sources. SQL queries are used to generate output to be displayed on the dashboard, Grafana continuously request data from the SQL database and displays it in a form that is more visually appealing.

We choose Grafana as compared to Kibana because Grafana supports multiple database management systems. This is because it could be possible that the government uses distributed databases.

4.4 Database Design

We utilized UML diagrams in order to enforce more clarity on the architectural design of the system. Below is the Extended Entity Relationship diagram of the system which shows the development process of the models and tables of the database in the server.

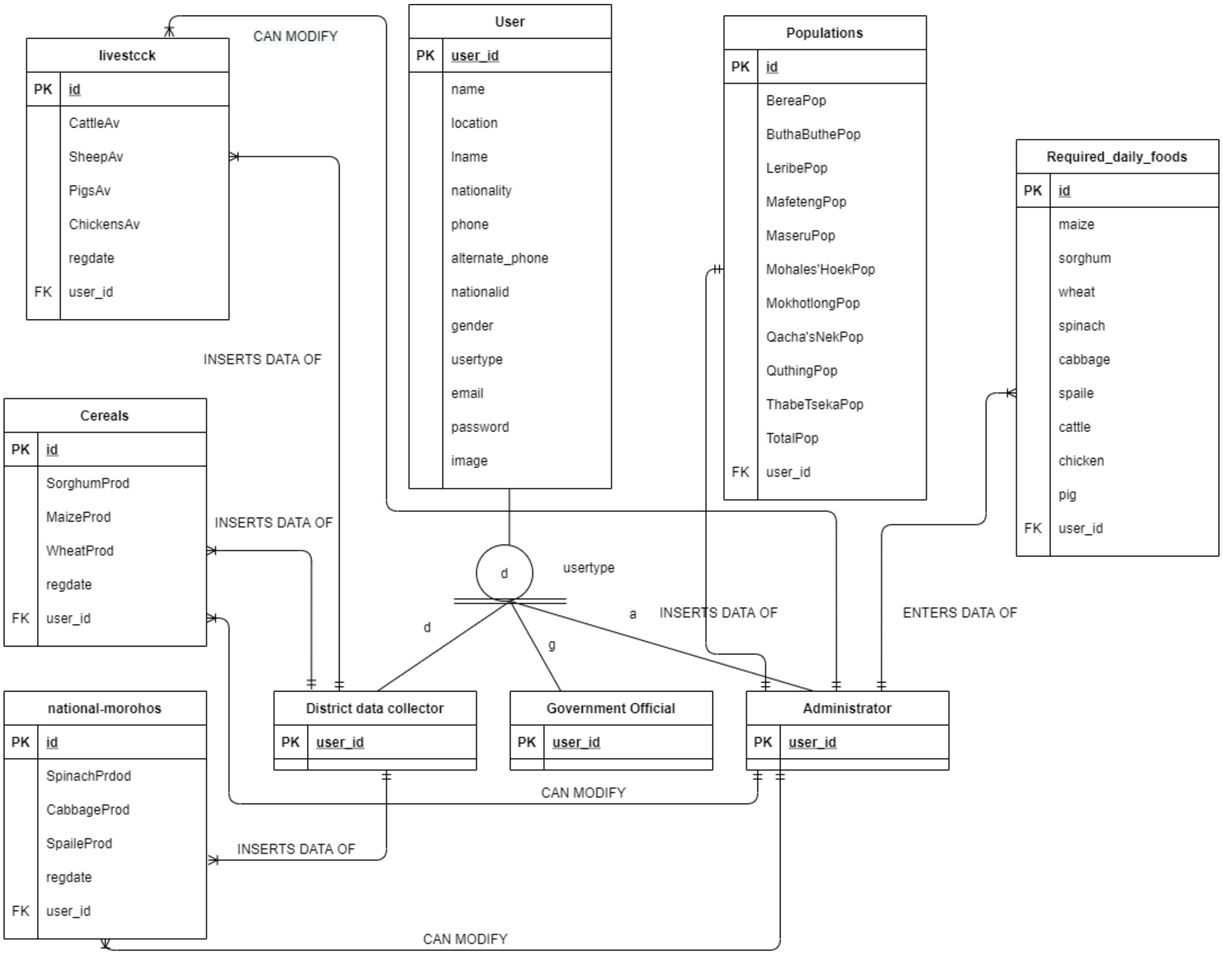


Figure 6- Extended Entity Relationship Diagram

With each link to the tables there is a verb that explains as per how they relate. For example, between administrator and required_daily_foods, the link connecting the tables is One to many relationships whereby one and only one administrator can (Enters Data Of) add one or many data of required_daily_foods.

4.6 How the System is visually designed

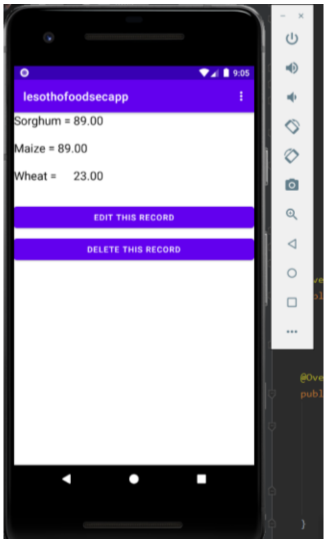
4.6.1 The registration:

A district data collector or admin can register on the mobile application and/or web application as per their choice of platform. The screenshots of the platforms are put in the appendix B section. For making the system improve security a google reCAPTCHA was added so that users will need to be validated for every change in their network. The reason why ReCAPTCHA is a prevalent tool in online submission forms is to prevent spam and abuse from entering the site [15]. Moreover, in improving the security of the system, the forms on the site use Laravel classes that utilize CSRF Protection/Cross-site request forgery. In addition, the mobile application forms also are secure as they don't allow the registrar from submitting a registration form with empty fields and need to log in once they leave or exit the application.

4.6.2 Data CRUD:

The system follows HCI- Human Computer Interaction concepts of navigation as it is simple for users to easy navigate the system as they perform data manipulation. They can find easy to go back and forth in the system as the buttons have colours that separate and enhance alerts, save, and delete and as well alerts for every operation done. The colour picking of our system is great because, the system in general uses simple colours that are consistent throughout the system. This does not confuse the user because users can right ahead be able to know which colours mean what without even reading the buttons. The mobile application and web application cereal registration is shown in the appendix B section.

The system web application comes with bootstrap. Not only does bootstrap provide fonts and CSS for page layouts it also helps to design websites faster and easier [16]. Therefore, the fonts chosen for the system are readable and users do not need to strain their eyes form usage of the system. Below is a screen shot of cereal values being manipulated



Sorghum = 89.00

Maize = 89.00

Wheat = 23.00

Figure 7- Shows the mobile application view

District Data Collectors and Administrators can both update their profiles and shown in the appendix.

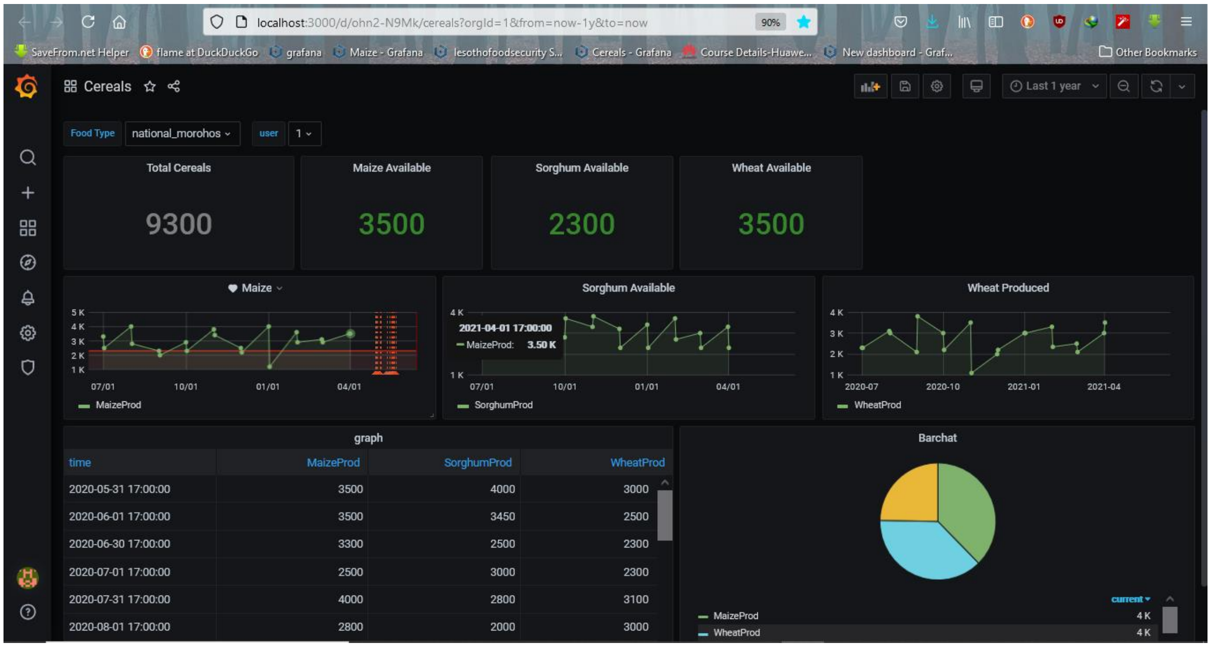
The system has also the administrator panel where the administrator can do the following;

- Edit and give user roles
- Add data on required minimum amount of food on different food types
- Register populations of the districts

The design of the administrator panel page is shown in the appendix B section.

System Dashboard Design

This is the panel that shows the cereals dashboard, it shows the total cereals available at a given period, their time series data and a detailed comparison graph of the three food types Maize, Sorghum, Wheat



Food Type national_morohos

user 1

Total Cereals	Maize Available	Sorghum Available	Wheat Available
9300	3500	2300	3500

graph

time	MaizeProd	SorghumProd	WheatProd
2020-05-31 17:00:00	3500	4000	3000
2020-06-01 17:00:00	3500	3450	2500
2020-06-30 17:00:00	3300	2500	2300
2020-07-01 17:00:00	2500	3000	2300
2020-07-31 17:00:00	4000	2800	3100
2020-08-01 17:00:00	2800	2000	3000

Barchat

Chapter 5 – Experiments and Results

Introduction

In this Chapter we show the system performance and evaluation under several factors. The factors and methods to test our system include unit testing, integration testing, system testing and usability testing.

The table below presents how the different methods of testing were carried out.

Unit Testing	Integration Testing	System Testing and Acceptance Testing	Usability Testing
This involves testing single methods for errors before they can be integrated with others. In these part methods need to work as needed IDE such as Android Studio and Visual Studio have built features to support the above. TDD	Individual modules are combined and tested as a group. Combining methods in isolation for resources conflicts errors. It focuses on checking data transfer amongst modules. Communication between login.java class and the activity_login.xml file in order to show UI in mobile application.	We test the system as a whole to check whether user requirements are met. This Testing checks the complete end to end scenarios, as the users of the system would use the system.	This is about getting real people to interact with the product built and observing their behavior and reactions to it.

We visited The Agricultural department and asked some students to try and add data of basic cereals to the database. The test was done on the register cereals part of the mobile application. However, the application provided an “error class DatabaseConfig not” not found because the app could not connect to the database. The php file referenced in Cereals.java file was misplaced. The tested code is in the appendix	In the development of our mobile application, it carried it out by the us, the developers and testers.	The testing was done in 2 ways through alpha testing and beta testing.	This was carried out by different students at ICT lab. This is where students navigated through both the web application and the mobile application. At first the system was not user friendly and new people found it difficult to use, thereby it had to be refined for better navigation.

Network Speed

Testing the network speed: The System does not work at full speed because it is not being run on a live server but rather a local server. The local sever is constrained by a number of factors including hard drive speed of a local PC or laptop, RAM, and processor speed. But being implemented on a real dedicated server it will be much faster.

Responsiveness

Testing the responsiveness: The responsiveness of the web application is important when the website will be viewed on the tablets or smart phones.

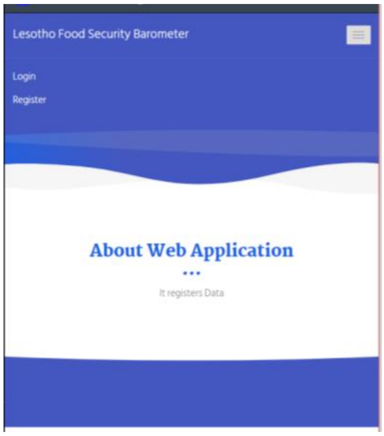


Figure 8- Screenshot of the landing Page

Notification

Testing the notification sending: The barometer should alert the government official once the food is below par.

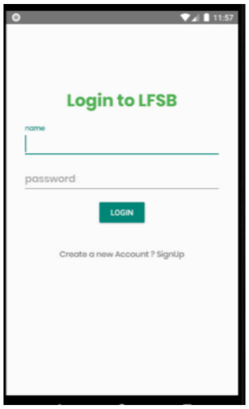
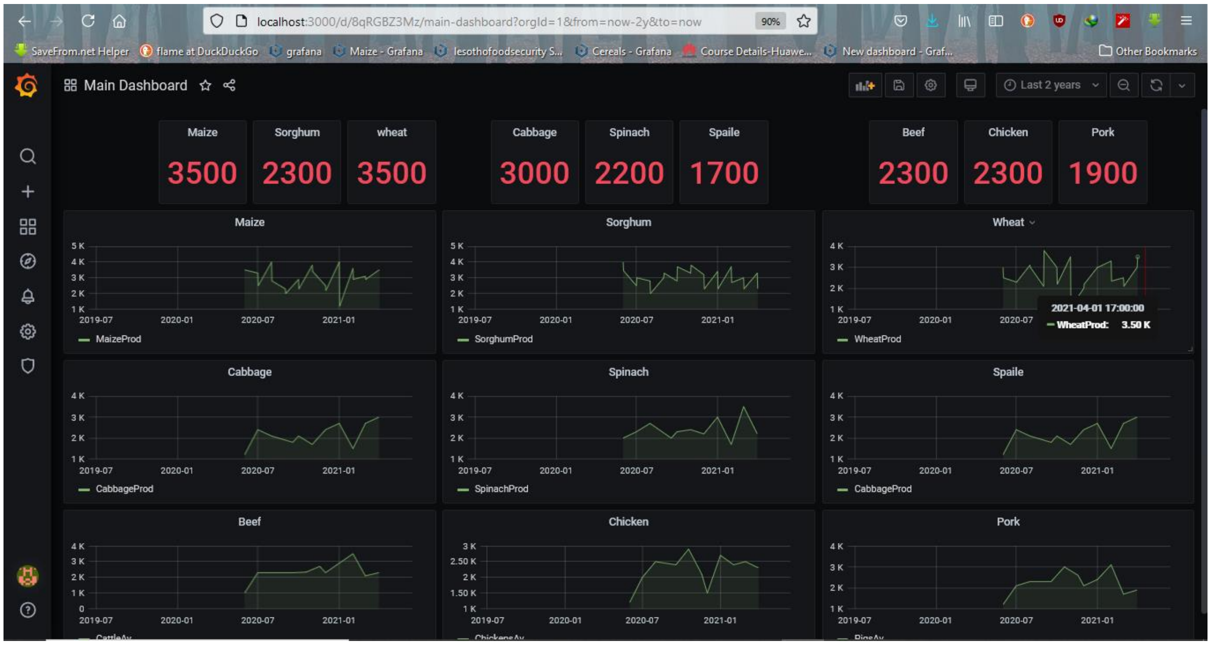


Figure 9- Show the Welcome layout of the mobile application

What the system reports:

Once the highlighted green values of Maize available in district Maseru, goes below the defined pie

The Results are thereby displayed to the user on the Grafana. For example below, the highlighted red colors indicate shortage.



Chapter 6 Future Work

The chapter outlines the functionalities that could have been implemented if we had more time to continue developing the project.

The original requirements that would have been implemented are:

- The system must send e-mail alerts to respective authorities if food availability levels drop below a certain threshold.

Along with the following new requirements:

- To improve the system's usability, A map could be added to the dashboard to show specifically the villages with low food quantities so that the government officials can know exactly which villages in a certain district require the most food assistance. This is because a district can be seen as having enough food to sustain itself yet most of the food in the district in held by few villages.
- More food types can be added into the system, this will help carter for people who might not eat the food already defined in the system. This can help the government to recognize which foods its people eat.
- Currently, the system is hosted on a localhost, it would be good if the system was hosted on the cloud, this is because cloud computing offers better security as compared to the traditional methods of storing data, moreover, cloud computing offers always on and available data accessibility, which means that the system will always be available.
- Improve the system by preventing district data collectors from accessing the records of other district data collectors.

Chapter 7 Conclusion and Recommendations

7.1 Introduction

This chapter outlines the results obtained in the project which sums the achievements and shortcomings.

7.2 Achievements

By the end of the schedule, the following could be reached and worked as defined by the scope if the project.

Users were able to;

- Register-
- Login
- Add Food types
- Display Food types
- Edit Food types
- Delete Food types
- Edit their information through user dashboards both admin and other

The system was able to;

- Display Dashboard to Government Official
- Display customized feedback to Government Official

7.3 Shortcomings and Evaluations

Although the system performed as was planned

- Due to the time constraint, we were not able to fulfil the following requirements: The system must send e-mail alerts to respective authorities if food availability levels drop below a certain threshold.
- Majority of the system's requirements were met, only the email notification feature could not be implemented due to time constraints

- Even though the system is not fully complete, at the stage it is now, it can be implemented into the outside world as it contains most of the features it needs

7.4 Conclusion

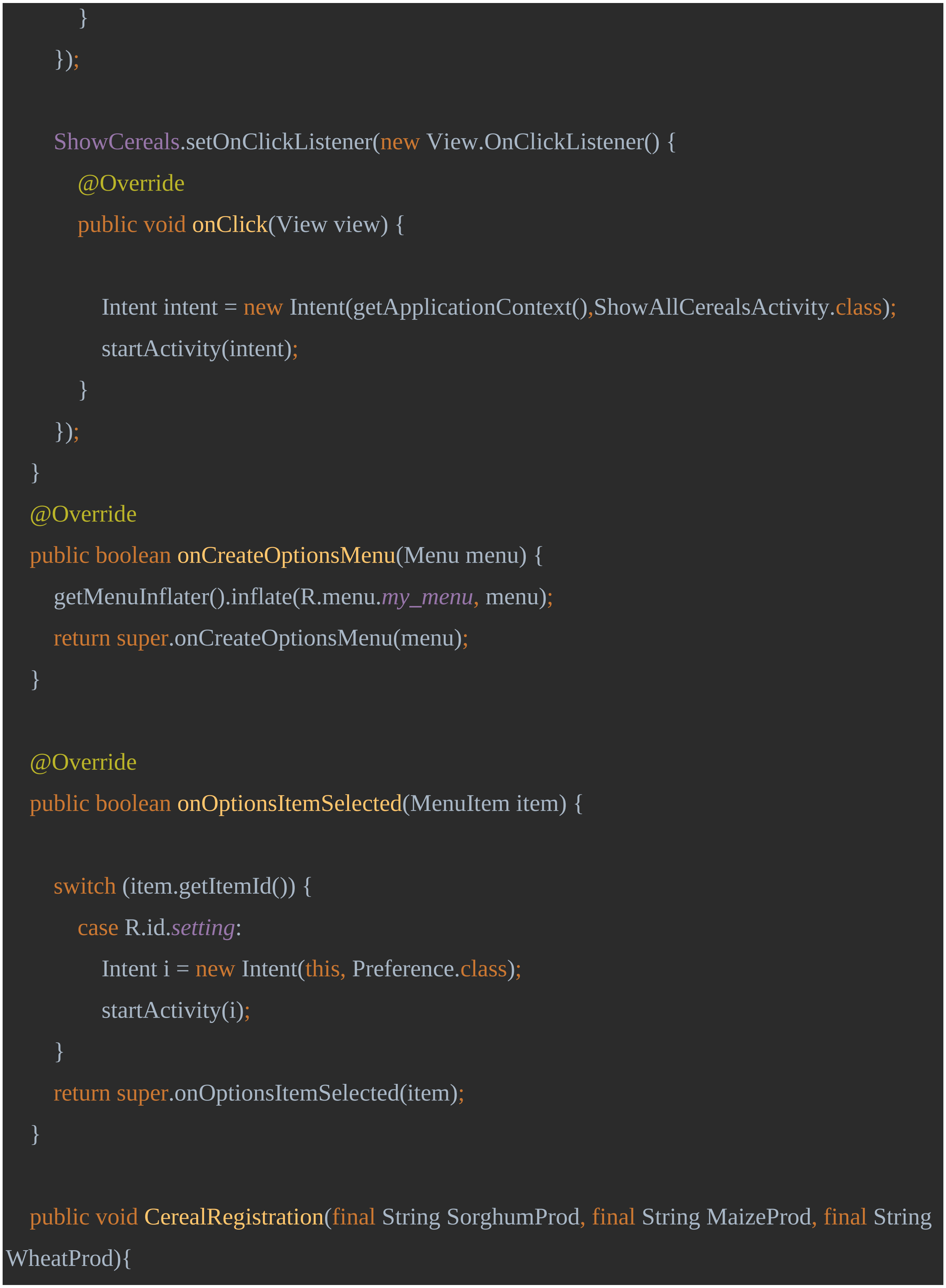
The design of a system that able to collect data on both national and district level food requirements as well as each district's harvest using either a mobile or web application, then store the data in a database and use a dashboard which collects its data from the database to periodically display information about food security status nationwide was achieved. The existence of such a system in Lesotho could be beneficial in the long term as it can help the country to plan its food security policies and strategies. In future more features could be added to the system to make it adaptive to be used in neighboring countries. Working on this project provided us with a fantastic learning experience. This project took us through the many stages of project development and provided us with real-world experience in creating projects. The challenges and problems gave us experience and a feel of how the developer's industry looks like.

Appendix

Appendix A: Code Snippets

The below is code for CerealsActivity.java that was tested

```
package com.ls.lesothofoodsecapp;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.util.HashMap;

public class CerealsAcivity extends AppCompatActivity {

    EditText SorghumProd, MaizeProd, WheatProd;
    Button RegisterCereal, ShowCereals;
    String CerealSorghumHolder, CerealMaizeHolder, CerealWheatHolder;
    Boolean CheckEditText ;
    ProgressDialog progressDialog;
    String finalResult ;
    HashMap<String,String> hashMap = new HashMap<>();
    HttpParse httpParse = new HttpParse();
    String HttpURL = "http://192.168.137.1/loginRegisterFOODSEC/registerCereals.php";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cereals_acivity);

        SorghumProd = (EditText)findViewById(R.id.editSorghum);
        MaizeProd = (EditText)findViewById(R.id.editMaize);
        WheatProd = (EditText)findViewById(R.id.editWheats);
        RegisterCereal = (Button)findViewById(R.id.buttonSubmit);
        ShowCereals = (Button)findViewById(R.id.buttonShow);

        RegisterCereal.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // Checking whether EditText is Empty or Not
                CheckEditTextIsEmptyOrNot();
                if(CheckEditText){
                    // If EditText is not empty and CheckEditText = True then this block will execute.
                    CerealRegistration(CerealSorghumHolder,CerealMaizeHolder,
CerealWheatHolder);
                }
                else {
                    // If EditText is empty then this block will execute .
                    Toast.makeText(CerealsAcivity.this, "Please fill all form fields.",
Toast.LENGTH_LONG).show();
                }
```

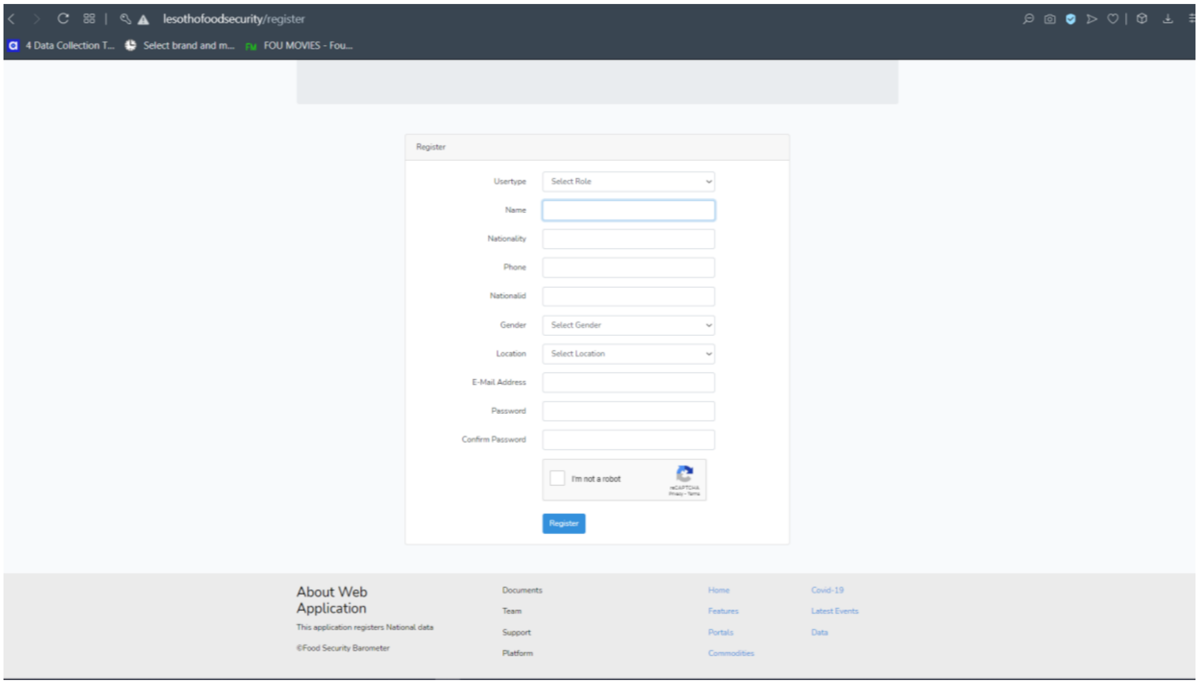


```
            }
        });

        ShowCereals.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                Intent intent = new Intent(getApplicationContext(),ShowAllCerealsActivity.class);
                startActivity(intent);
            }
        });
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.my_menu, menu);
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {

        switch (item.getItemId()) {
            case R.id.setting:
                Intent i = new Intent(this, Preference.class);
                startActivity(i);
        }
        return super.onOptionsItemSelected(item);
    }

    public void CerealRegistration(final String SorghumProd, final String MaizeProd, final String
WheatProd){
        class CerealRegistrationClass extends AsyncTask<String,Void,String> {
            @Override
            protected  void onPreExecute() {
                super.onPreExecute();
                progressDialog =
ProgressDialog.show(com.ls.lesothofoodsecapp.CerealsAcivity.this,"Loading
Data",null,true,true);
            }
        @Override
        protected void onPostExecute(String httpResponseMsg) {

            super.onPostExecute(httpResponseMsg);

            progressDialog.dismiss();

            Toast.makeText(CerealsAcivity.this,httpResponseMsg.toString(),
Toast.LENGTH_LONG).show();

        }

        protected String doInBackground(String... params) {

            hashMap.put("SorghumProd",params[0]);
            hashMap.put("MaizeProd",params[1]);
            hashMap.put("WheatProd",params[2]);

            finalResult = httpParse.postRequest(hashMap, HttpURL);
            return finalResult;
        }
    }
```

```
    CerealRegistrationClass cerealRegistrationClass = new CerealRegistrationClass();
    cerealRegistrationClass.execute(SorghumProd,MaizeProd,WheatProd);
    }

    public void CheckEditTextIsEmptyOrNot(){
        CerealSorghumHolder = SorghumProd.getText().toString();
        CerealMaizeHolder = MaizeProd.getText().toString();
        CerealWheatHolder = WheatProd.getText().toString();

        if(TextUtils.isEmpty(CerealSorghumHolder) || TextUtils.isEmpty(CerealMaizeHolder) ||
TextUtils.isEmpty(CerealWheatHolder))
        {
            CheckEditText = false;
        }
        else
            {
            CheckEditText = true ;
        }
    }



}
```

Appendix B: Models

Register Page



lesothofoodsecurity/register

Register

Usertype	Select Role
Name	
Nationality	
Phone	
Nationalid	
Gender	Select Gender
Location	Select Location
E-Mail Address	
Password	
Confirm Password	

□ I'm not a robot

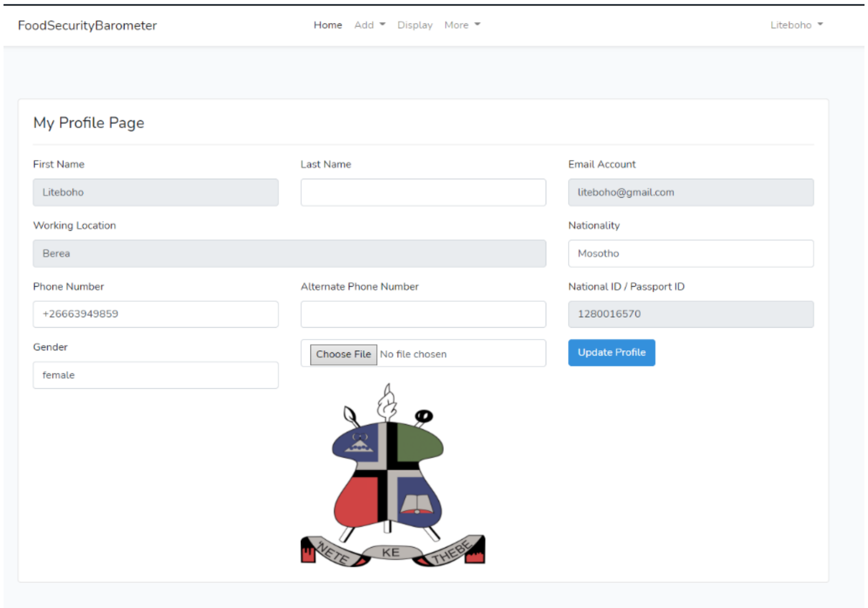
About Web Application

This application registers National data

©Food Security Barometer

Figure 10- Register Page

Profile Page



My Profile Page

First Name	Last Name	Email Account
Liteboho		liteboho@gmail.com
Working Location		Nationality
Berea		Mosotho
Phone Number	Alternate Phone Number	National ID / Passport ID
+26663949859		1280016570
Gender	Choose File No file chosen	Update Profile
female		

Figure 11- Profile Page

Signup Mobile

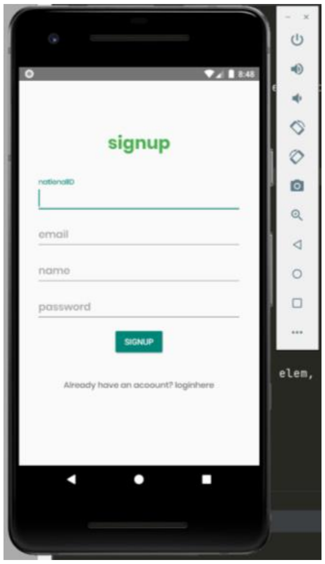
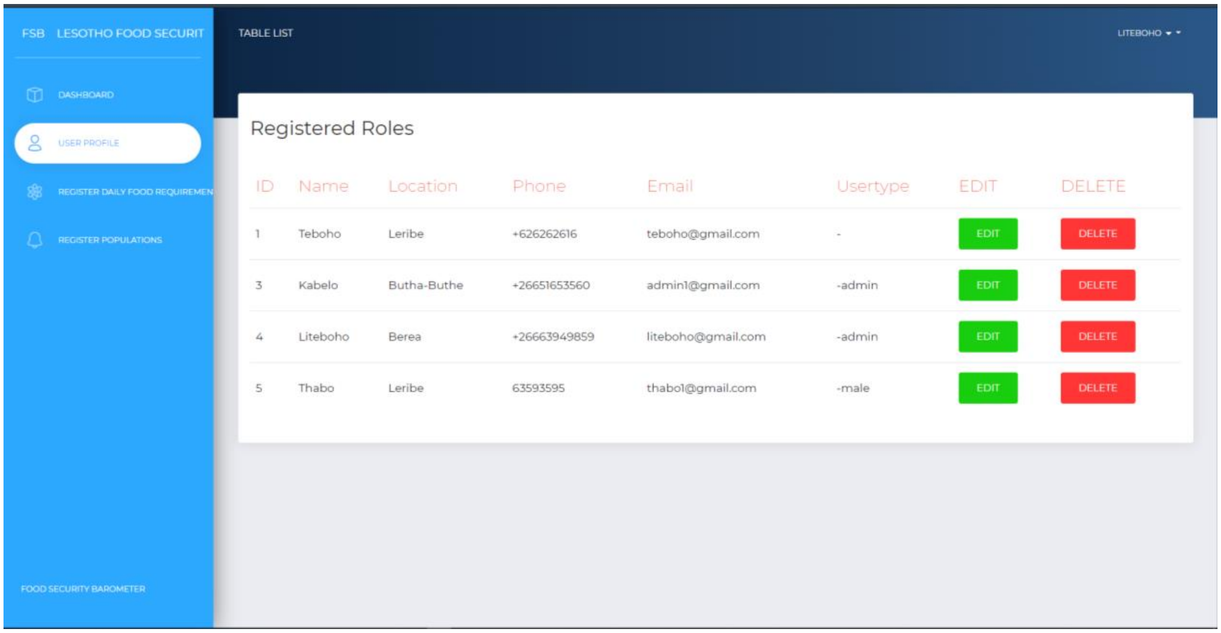


Figure 12- Signup of Mobile

Admin Page



Registered Roles

ID	Name	Location	Phone	Email	Usertype	EDIT	DELETE
1	Teboho	Leribe	+626262616	teboho@gmail.com	-	EDIT	DELETE
3	Kabelo	Butha-Buthe	+26651653560	admin1@gmail.com	-admin	EDIT	DELETE
4	Liteboho	Berea	+26663949859	liteboho@gmail.com	-admin	EDIT	DELETE
5	Thabo	Leribe	63593595	thabo1@gmail.com	-male	EDIT	DELETE

Figure 13- Admin Page

References

[1] J. Barnhart, "Feed The Future," 2020. [Online]. [Accessed 06 April 2021].

[2] "worldinfozone," 2021. [Online]. [Accessed 1 April 2021].

[3] "World Health Organization," 2021. [Online]. Available: https://www.who.int/data/gho/data/indicators/indicator-details/GHO/population-using-at-least-basic-drinking-water-services-(-). [Accessed 3 June 2021].

[4] "Sustainable Development," [Online]. Available: https://sustainabledevelopment.un.org/content/documents/2076Consolidated%20Technical%20inputs.pdf. [Accessed 3 March 2021].

[5] A. Choudhry, "The Importance of Node.js in the Responsive Web Era," 19 February 2016.

[6] S. D. M. Shafirov, "JetBrains," JetBrains, 14 August 2000. [Online]. Available: https://www.jetbrains.com/phpstorm/features/#:~:text=PhpStorm%20helps%20you%20get%20around,or%20search%20for%20its%20usages.. [Accessed 6 June 2021].

[7] "Developers," 2021. [Online]. Available: https://developer.android.com/studio/intro. [Accessed 6 June 2021].

[8] "Grafana," [Online]. Available: https://grafana.com/docs/grafana/latest/getting-started/. [Accessed 3 June 2021].

[9] T. Swati, "Educba," 2020. [Online]. Available: https://www.educba.com/what-is-laravel/. [Accessed 5 May 2021].

[10] P. Tim, "When to Use Waterfall vs. Agile," 17 May 2019. [Online]. Available: https://www.macadamian.com/learn/when-to-use-waterfall-vs-agile/.

[11] K. Black, "Sooperarticles," 21 May 2021. [Online]. Available: https://www.sooperarticles.com/technology-articles/mobile-application-development-articles/reasons-why-mobile-apps-better-than-websites-2021-1809978.html.

[12] S. Andujar, "Askinglot," 13 May 2020. [Online]. Available: https://askinglot.com/what-is-a-layout-in-android-studio.

[13] S. Team, "Syndicode," 21 August 2017. [Online]. Available: https://syndicode.com/blog/5-open-frameworks-advantages-for-web-app-development/.

[14] L. Scott, "Tech Republic," *Step-By-Step: Integrate Apache and MySQL with PHP on Linux,* 14 November 2001.

[15] "whispir," 2020. [Online]. Available: https://www.whispir.com/blog/recaptcha-preventing-spam. [Accessed 3 June 2021].

[16] S. Nathan, "HTMLGOODIES," May 10 2016. [Online]. Available: https://www.htmlgoodies.com/html5/10-common-uses-of-bootstrap/. [Accessed 1 June 2021].

[17] WOCATpedia, "Definition and Dimensions of Food Security," 2020. [Online]. Available: https://wocatpedia.net/wiki/Definition_and_Dimensions_of_Food_Security#Availability. [Accessed 29 11 2020].

[18] William Craig, "WebFX," 2010. [Online]. Available: https://www.webfx.com/blog/web-design/5-fundamental-steps-to-deploying-a-website/. [Accessed 21 12 2020].

[19] Wikipedia, "Java(programming language)," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Java_(programming_language). [Accessed 1 12 2020].

[20] Wikipedia, "Grafana," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Grafana. [Accessed 1 12 2020].

[21] "Artificial Intelligence," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Artificial_intelligence. [Accessed 05 November 2020].

[22] WHO, "International Food Policy Research Institute," 2020. [Online]. Available: https://www.ifpri.org/topic/food-security#:~:text=Food%20security%2C%20as%20defined%20by,an%20active%20and%20healthy%20life.. [Accessed 29 11 2020].

[23] "Web Developers Notes," 2021. [Online]. Available: http://www.webdevelopersnotes.com/what-is-web-server. [Accessed 6 June 2021].

[24] tutorialspoint, "Sql tutorial," 2020. [Online]. Available: https://www.tutorialspoint.com/sql/index.htm. [Accessed 1 12 2020].

[25] Tobii VR, "Tobii," 2018-2020. [Online]. Available: https://vr.tobii.com. [Accessed 09 11 2020].

[26] Richard H. R. Harper, "The Roleof HCI in the Age of AI," *International Journal of Human-Computer Interaction,* vol. 35, no. 15, pp. 1331-1344, 2019.

[27] R. Nagare, "Scale Dynamix," 27 June 2018. [Online]. Available: https://scaledynamix.com/blog/what-is-php-composer/#:~:text=Composer%20is%20basically%20a%20dependency,a%20project%20to%20project%20basis.. [Accessed 06 June 2021].

[28] PIYUSH PATHAK, "Human-computer interaction: An AI revolution," [Online]. Available: https://medium.com/@piyushpathak03/human-computer-interaction-an-ai-revolution-76d967786468. [Accessed 09 11 2020].

[29] Nielsen, "Usability Government," 2020. [Online]. Available: https://www.usability.gov/how-to-and-tools/methods/prototyping.html. [Accessed 10 11 2020].

[30] K. Moeketsi, *UML.*

[31] Linkedin, "How Artificial Intelligence is tranforming Human-Computer Interaction, and its implications for Design," 2017. [Online]. Available: https://www.linkedin.com/pulse/how-artificial-intelligence-transforming-interaction-its-ganapathy. [Accessed 09 05 2020].

[32] Laura Busche, "Smathing Magazine," *The Skeptic's Guide to Low-Fidelity Prototyping,* 2014.

[33] kinsta, "kinsta," 2021. [Online]. Available: https://kinsta.com/knowledgebase/what-is-laravel/. [Accessed 06 May 2021].

[34] Indeed, "7 Steps of the Design Process," 2020. [Online]. Available: https://www.indeed.com/career-advice/career-development/design-process. [Accessed 05 11 2020].

[35] "eukhost," 14 December 2015. [Online]. Available: https://www.eukhost.com/blog/webhosting/server-side-scripting-pros-and-cons/#:~:text=As%20the%20scripting%20takes%20place,membership%20and%20social%20media%20sites.. [Accessed 07 June 2021].

[36] Elisa Giaccardi, "Association for Computing Machinery," 2019. [Online]. Available: https://dl.acm.org/doi/10.1145/3322276.3322351. [Accessed 06 November 2020].

[37] "Computer Hope," 1998. [Online]. Available: https://www.computerhope.com/jargon/p/programming-language.htm. [Accessed 6 June 2021].

[38] Bureau of Statistics, "Lesotho Agricultural Census Volume V: Technical Report," 2019.

[39] D. Austin, "What is the importance of client-side scripting to an application lifecycle? How it can be implemented," p. 1.

[40] AREND HINTZE, "Government Technology," 2016. [Online]. Available: https://www.govtech.com/computing/Understanding-the-Four-Types-of-Artificial-Intelligence.html. [Accessed 05 November 2020].

[41] "W3schools," 1999. [Online]. Available: https://www.w3schools.com/sql/sql_intro.asp. [Accessed 1 June 2021].

[42] W. F. Programme, "World Food Programme," [Online]. Available: https://docs.wfp.org/api/documents/WFP-0000120932/download/. [Accessed 3 June 2021].