FARM DATA MANAGEMENT SYSTEM TO INCREASE EFFICIENCY AND REDUCE COSTS IN COLLECTING AND STORING DATA AND MANAGING FARMING ACTIVITIES FROM PLANTING STAGE UNTIL HARVESTING.

BCISLMR170919

December 17, 2021

Abstract

Lack of historical farming data to support the decision-making processes and over relying on outdated data has resulted in farmers making misinformed decisions that severely affects their farms and yields. This project aims to create a farm data management system that will modernize the way the farmer collects, processes, and stores farm data. It will also allow them to manage farming activities in an efficient manner. Data will be collected from all the key stakeholders and other sources to inform the design and implementation processes of the project.

1 Introduction

1.1 Background

Agriculture is one of the major sources of food for humans. For years we have relied on classical farming methods to help us produce more. The ever increasing global population which according to the UN Department of Economical and Social Affairs, is projected to grow to over 9 billion by 2050, continues to demand more and more production output from this industry.

Currently, technological advances in areas such as artificial intelligence, robotics, big data analytics, and Internet of Things (IoT), continues to cultivate and inspire innovations in agriculture. New farming practices have emerged which majorly leverage technology as the key component that would ensure that they succeed, remain sustainable, and environmentally friendly. These new practices include vertical farming, precision agriculture, and Smart farming. Vertical farming involves growing crops in vertically stacked layers in an environmentally controlled, well-regulated, self-contained structure (like a building or special-designed greenhouse). (Byjus, 2021). Microsoft Commons is an example of a vertical farm categorized as an indoor farm. It has a planting-to-harvest cycle of 28 days and an annual yield of nearly 6800kg. Vertical farms can be categorized into the rooftop, greenhouse, and urban farms among others. (Tower Farms, 2021). Precision farming sorts to use technology to increase yields and profitability whilst decreasing the dependency on traditional farming inputs like water, land, and chemicals. It encourages to use less to produce more. (Rogers, 2014). Aker Technologies Inc. has developed patented computer vision and biometric sensors, and software solutions to identify, classify, measure and predict crop diseases, pests, and pathogens. Aker aims to improve profitability and increase yields by transforming crop monitoring. (Aker, 2021). Lastly, smart farming relies on technologies like robotics, drones, sensors, AI, and Internet of Things

(IoT) to manage farms, increase the quality and quantity of goods whilst optimizing production methods. (Sciforce, 2020).

These new farming practices are the future since they encourage tracking and monitoring every farming activity to get to know better how different activities and approaches may affect the productivity of the farm. The practices require and generate a lot of data. The major problems experienced are cost-ineffective processes for data collection, unreliable data sources, poor data quality, unclear statements of who owns the data, and lack of better and easy-to-use data management tools and systems.

1.2 Problem statement

The major problem especially in Kenya, farmers lack the necessary data to inform their decisions and understand how the farm performs over a certain period. Most rely on gut instinct, inherited knowledge, and data that dates back the latest 2-4 years which might be inaccurate since they didn't or partially recorded it. What was experienced and done in the last 10 years can't be applied today since many factors can influence the output of a farm field. Changes in the type of seeds, climate and weather of a region, and soil nutrients and pH. These factors if tracked can be usefully in provide concrete and actionable data.

Another problem is the data collected on a farm doesn't benefit the owner. Big agriculture organizations use this data to determine how a particular product would perform in those farms and tweak the product for maximum benefits then present it to the farmer at high prices.

1.3 Objectives

1.3.1 General objective

The main objective was to come up with a platform that farmers can utilize to collect data, track, and manage every process and activity from the planting season up to harvesting season in an efficient manner.

1.3.2 Specific Objectives

The main objectives will be achieved by:

- Creating a page with input fields where the farmer can enter data based on the current farming stage. If the farming stage changes then, the inputs fields should change accordingly.

- Providing a page where the farmer can view the data collected in a format like a table. The farmer will filter data, for instance, based on soil type, crop type, or acreage used.

- Creating a page with the activity-viewing tab and activity-scheduling tab. In the activity-viewing tab, the farmer will view and track previous and ongoing activities with their status. In the activity-scheduling tab, the farmer can schedule oncoming activities.

- Providing an admin page where the farmer (superuser) can create, alter, or delete users (staffs). The admin page should allow the superuser to grant different access rights to various users which would define how these users interact with data.

1.4 Justification

Currently, data is a most valuable commodity than oil. Anyone who can collect and control data, has power over others. This platform benefits farmers who want to create, collect, and control their data, be the first person to benefit from the usage of that data, efficiently manage farm processes, and use it to make data-driven decisions. The farmer may sell the data to:

- Seed, fertilizer, and chemical production companies for them to maximize the quality, improve effectiveness, and output of their product.

- The government advisories agencies that can use it to predict the food output of a region and combined data from the different farmers all over the country can be used to determine whether the country has food sustainability. The agencies can also use the data to formulate crop insurance and policies on supply chain operations.

1.5 Scope

The project covered data collection and track every process and activity from the start of planting season, stages in between, up to the harvesting season. Soil quality and pH, weather, chemicals and fertilizers used, crop planted, seeds type, all expenses, and yields.

1.6 Limitations

These the limitations the project may experience:

- Resistance to change where farmers might be unwilling to change their methods of tracking and managing farming activities and adapt to the new methods. Resistance might be brought forth by fear of failure, mistrust, and lack of confidence in the new platform.

- Lack of computer literacy. Most farmers are literate where they can read and write but not all of them have the basic computer skills needed by this platform.

- Lack of access to affordable, fast, and reliable internet. Many farmers live in rural areas where access to the internet is a huge problem. Some don't have access to a cellular network while others have weak cellular network signals.

2 Literature Review

2.1 Theoretical Review

This section reviews what has been done by different authors on sustainability, collection of data and application of digital technologies in agriculture. The review was guided by the question: how data and digital technologies can help achieve sustainable agricultural practices?

2.1.1 How data and digital technologies can help to achieve sustainable agriculture

Sustainable practices are practices that aim to assist us to meet the needs of the present without affecting the future generation's ability to meet their own needs by creating conditions where there is harmonious coexistence between humans and nature. (EPA, 2021). Many industries have been experiencing more and faster modernization of their processes whilst the agriculture industry has been lagging behind. According to the UN Department of Economic and Social Affairs, the food sector (agriculture) accounts for 22 per cent of greenhouse gas emissions due to the conversion of forest to farmland and the use of synthetic chemicals and outdated farming techniques to increase output. (UNDESA, 2021). The agriculture sector acts as the source and the sink for greenhouse gases, by changing how different things are carried out and how it utilizes the available and scarce resources. This sector can have a green footprint and become sustainable. (Schahczenski & Hill, 2008).

Agriculture has gone through many evolutions like the use of tools during Agrarian history and the use of tractors during the mechanisation of the agricultural processes. The current evolution is called Agriculture 4.0, which involves precision farming that is highly encouraged by areas like cloud computing, big data analytics, cheap and better microprocessors, sensors, and actuators, fast internet and increased cellular network coverage, and advancements in automation. (Kovács & Husti, 2018). All the previously mentioned areas depend on or generate data. Data is the key to pushing agriculture to this new frontier, transforming farmers into advocators of sustainability and productivity, helping in the dissemination of information, knowledge and best practices. (UjuziKilimo, 2021).

Data collected from different sources, such as operations, transactions, images and videos from sensors, are large, complex, and unstructured. The conventional data processing and management tools and methods can't be applied to the data. For this reason, new and advanced tools and techniques have are been developed, evaluated, and continue to be perfected to ensure new and more insights are being extracted from the data at a faster rate. (Sharma, 2020). Data mining is one of the advanced techniques used to mine useful information or find patterns in data. This technique is at the intersection between machine learning, statistics, and database systems. (Majumdar, Naraseeyappa, & Ankalaki, 2017). Insights extracted from the data collected can be used in:

- Boosting productivity through better management of fertilizers, seeds, agrochemicals, and other resources.
- Risk management involves reducing crop failure as a result of factors like changes in climate and weather patterns, low-quality seedlings, and others.
- Yield prediction where predictive models are used to obtain crop production estimates and insights on weather, chemicals, and biomass index of the soil. (Sharma, 2020).
- Companies involved in agriculture in planning supply chain decisions like scheduling of production and marketing activities. (Majumdar, Naraseeyappa, & Ankalaki, 2017).

Africa has over 33 million smallholder farmers and around 60 per cent of the world's uncultivated arable land (O'S, 2013). To ensure that everyone is on board with the plan to change the agriculture sector and ascertain food security. These smallholder farmers are the future since the future of agriculture will be highly focused on small farms rather than large ones. Hence, the need to empower and educate them on new ways of doing things that prevents them from over-relying on outdated, inaccurate, and inherited knowledge while making decisions on the food they produce. Digital technologies can help in bridging gaps between what the first-world countries and third-world countries in Africa when it comes to food production, provide faster delivery of services and monitoring progress, better yields that consume fewer resources like water, and most importantly improve the social-economic conditions of where the farmer is located resulting to poverty eradication. Smart and solar-powered irrigation kits, like the ones made by SunCulture, are some of these digital technologies. These kits are cheap and affordable helping smallholder farmers who are even in rural areas to irrigate their crops in an efficient way thereby, allowing water conservation and saving of money that would have been spent on expensive and wasteful irrigation systems. Additionally, dissemination of correct, timely and accurate information, knowledge, and best practices can leverage the revolution and increased penetration of mobile phones in Africa. (UjuziKilimo, 2021).

2.2 Related Projects

2.2.1 Apollo Agriculture

Apollo assist small-scale farmer maximize their profits by accessing customized credit package, quality farm inputs, and advice on how to increase yields. Apollo assesses credit risk and customized each package to farmer's specific location by collecting different types of data such as satellite, soil, farmer behavior and crop yield data. (Cruchbase, 2021). Machine learning models are applied to the data to assist in making credit decisions. If the farmer is approved for a loan, they receive a voucher code which he or she can redeem in one of the 250 Apollo approved agriculture retailers in Kenya. (Bosilkovski, 2020).

2.2.2 NavFarm Diary

The platform helps in collection and monitoring data in diary farming . It also maintains milk production records for entire cattle herd. Dialy, weekly, and monthly summaries are maintained which can include yield per day. Herd or diary management module is responsible for registration of herd performance, record sales with multi-user support. Milk Parlour links is used to transfer breeding information to and from computerized milking parlours. Cattle breeding is used to set breeding reminders, register new calves, and diary herd performance including pregnancy and submission rates. Feeds module tracks supplements, feeds growth ratios, body fat and water consumption

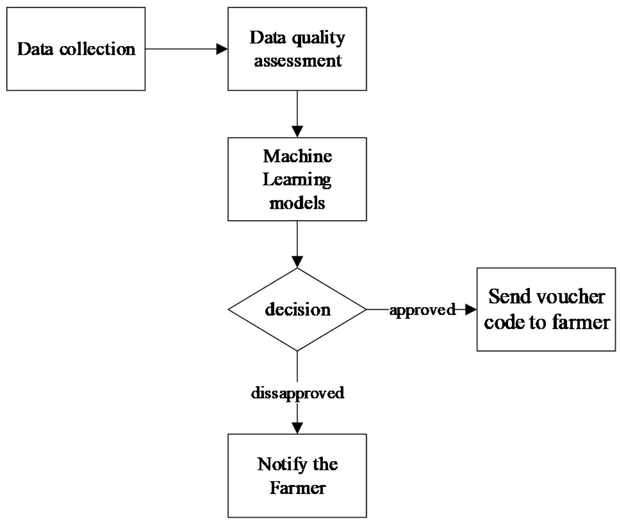

Figure 1: Apollo Agriculture

in different breeds and weather conditions. All the outputs from the named modules are processed in the processing module which has real-time yield control, tracking of milk production, and others in order to produce reports and summaries. (NavFarm, 2021).

2.2.3 Conservis

Conservis is a platform that unifies multiple streams of data to allow farmer to track farm activities in real-time. (SoftwareConnect, 2021).

Planning and budgeting module allows creation of operational and financial plans, track actuals to budget and respond to changing conditions. It also includes a submodule for purchasing and inventory, which is used to consolidate input shopping list and real-time purchasing orders and keeps track of the inputs at hand. Production and work order module is responsible for production tracking and reporting where it keeps track of the operations and their cost to boost efficiency, accuracy, and accountability. Harvest and reporting module is made up of harvest management and loading and delivery tracking submodule. Harvest management allows condition of harvest data collection on harvest and coordination of crews and equipment, and keeps track of problems for resource accountability. Loading and delivery tracking is responsible for monitoring yields in realtime, trace loads from the farm location to destination, and payment handling. (Conservis, 2021).

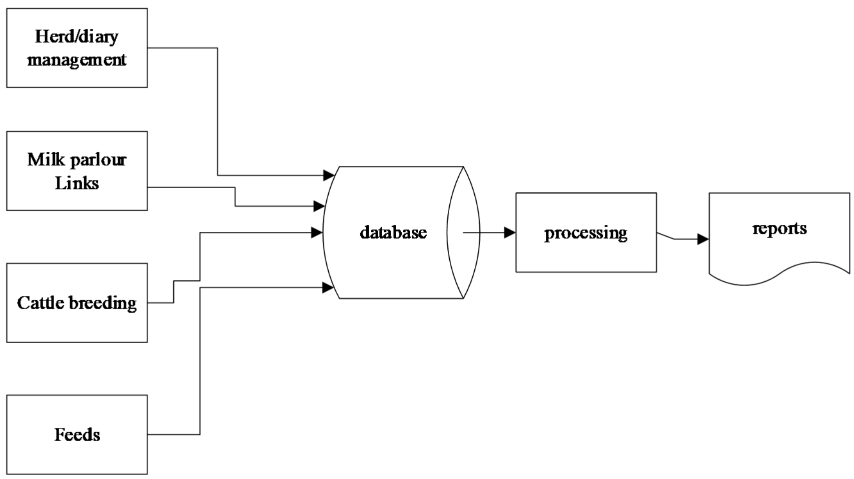

Figure 2: NavFarm

The reporting functionality in Conservis, generates reports effortlessly at time-saving pace. The reports can be used by regulators, insurers, and stakeholders.

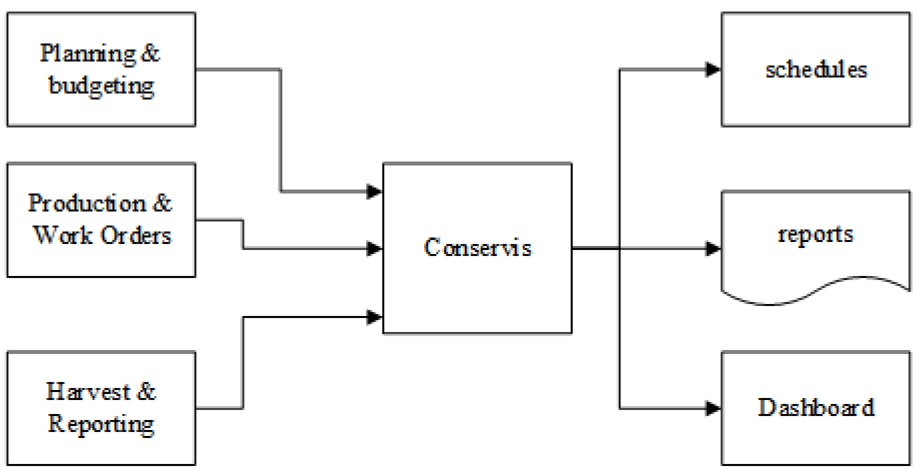

Figure 3: Conservis

2.3 Conceptual Framework

This subsection introduces and describes different parts (Inputs, processes, and outputs) of the conceptual framework of the system. The input section contains user data, data collection, and tasks. User data includes name, national ID number, staff ID, and other information that are to be verified, processed, and stored in order to create a user profile. Farm data collection represents all the types of data that will be collected. Farm data to be collected include farm stages (tillage, planting, spraying, irrigation, cultivation, and harvest), weather, resources used, farm code, crop name, and seed type, and expenses. Data collected is loaded directly to the database while others like images must be processed (resizing and renaming). The data is cleansed to ensure quality and establish consistency. The data is stored in a manner that makes it simple to retrieve, manipulate, and modify. A well-formatted data is presented to the user. The user can filter, order, modify, and manipulate data as he or she wishes. The task section is where the user is able to create farm tasks that must be accomplished. Attributes of a task are considered inputs to the system. These attributes include but are not limited to name, start date, deadline, created by, status, notes, and list of objectives. List of objectives actions that must be completed individually to set the overall task's status as complete. Complete, ongoing, halted, incomplete, and not started are statuses a task (one status at a time) can have. Task notes allow extra but concise information to be added to a task. Once the task information is stored, it is retrieved with others and presented to the user as a schedule.

3 Methodology

3.1 Data Sources

The farmers that the platform aims to benefit are the main source of data. Data collected was used to understand how the farmer needs the system to perform in order to consider this platform as a viable solution to the problems it aims to solve. The data was used to understand how the farmers work, what they produce, how they use existing systems, and how they may change to accommodate a new system. (Sommerville, 2016). Data collection methods described in section 3.2 were utilized to gather the data.

Secondary sources of data included the internet and print media like case studies, books, and journals. Data collected from these sources was used to supplement the information collected by other methods and on how to approach the project implementation. These sources helped to answer this major question; what are the current and efficient software development methods and designs that ensure accessibility, scalability and security of the platform?

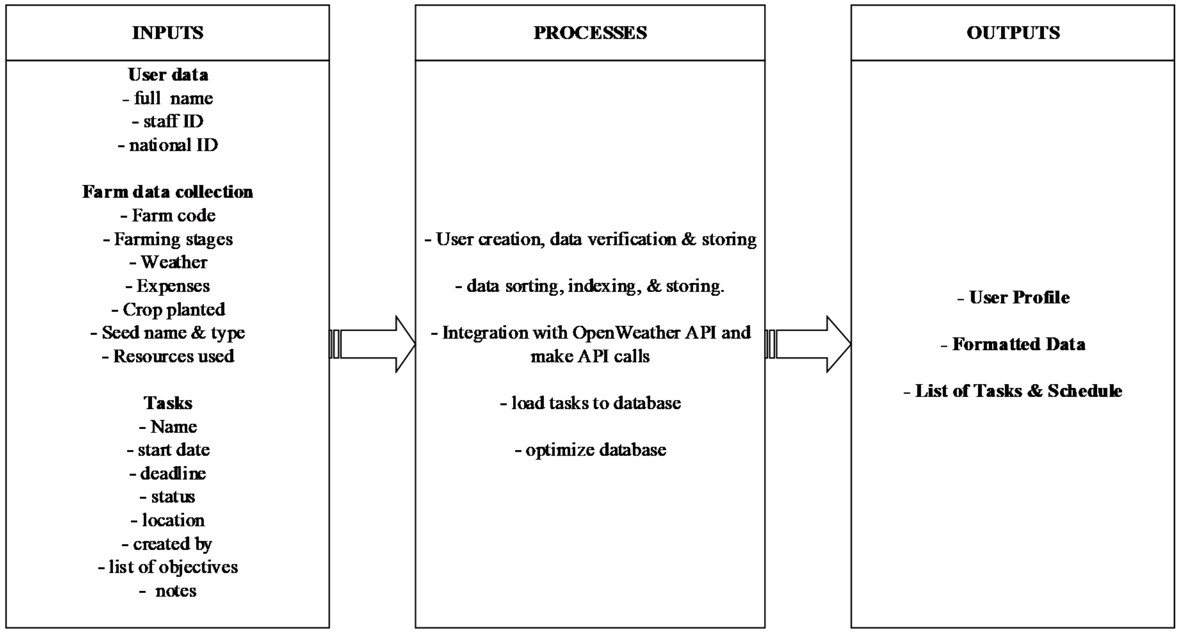

Figure 4: Conceptual Framework

3.2 Data Collection

3.2.1 Observation

It involved observing farmers as they go about their work and documenting all the steps. Documentation helped in eliciting processes the system must support and potential mistakes and risks. As an external observer, it was easier to identify inefficient process and come up with optimal solution. The observation was carried out in two forms:

- Field observation - involved being on the location with the farmer and documenting all the processes and operational procedures that were being carried out.

- Participant observation - required the observer to be unbiased and actively perform and learn the procedures of the farmers while questioning unclear and complex operational procedures to gain domain experience. This form of observation helped to understand the domain language hence, simplifying further data collection using other methods, discovering implicit requirements, and understanding how the farmers work rather than the way in which farming processes definitions say they should work.

3.2.2 Interviews

Interviews were used in conjunction with observation to get more and essential information that observation may have missed. The interview provided first-hand information, chance to clarify topics as they come up, and flexibility to adapt to the situation and get as much information as possible. The interviews were carried out in two forms; closed and open interviews. In a close interview, a set of predefined questions were prepared whilst in an open interview, a wide range of topics and issues were explored. Topics and issues like what they particularly do, difficulties faced with current systems in the market, and how they might interact with the new platform. The aim was to make the stakeholders the focus and since people love to have discussions about their work, it was easier to create rapport and make them more open to answering questions. Hence, elicit more meaningful data and system requirements.

3.3 Architectural Framework

The architectural framework of the platform followed the Model-View-Controller (MVC) architectural pattern. MVC is the best option because it allows separation of concerns thus, one part of the system can be modified without affecting other parts. The views (mainly deals with the user interface) or modules of the platform are responsible for data presentation, querying and displaying the state of the system, and sending user events to the controller. The views include User registration and login, data collection, data viewing, task scheduling and viewing, and admin dashboard. The Controller is responsible for processing HTTP requests, data processing and validation, user creation and authentication, mapping user events to model updates, and making API calls to OpenWeather API for the latest weather data as per the user request. The Model provides an abstraction layer (the "models") for structuring and manipulating data and is also responsible for every interaction with the database.

When the user sends a request, the request is sent to the Controller. The Controller examines the request for user action. User actions are indicated by the HTTP methods (GET, POST, PUT, DELETE) in the request headers. The controller checks whether the user is requesting data or just navigating. If the user is navigating from one view to another and these views don't

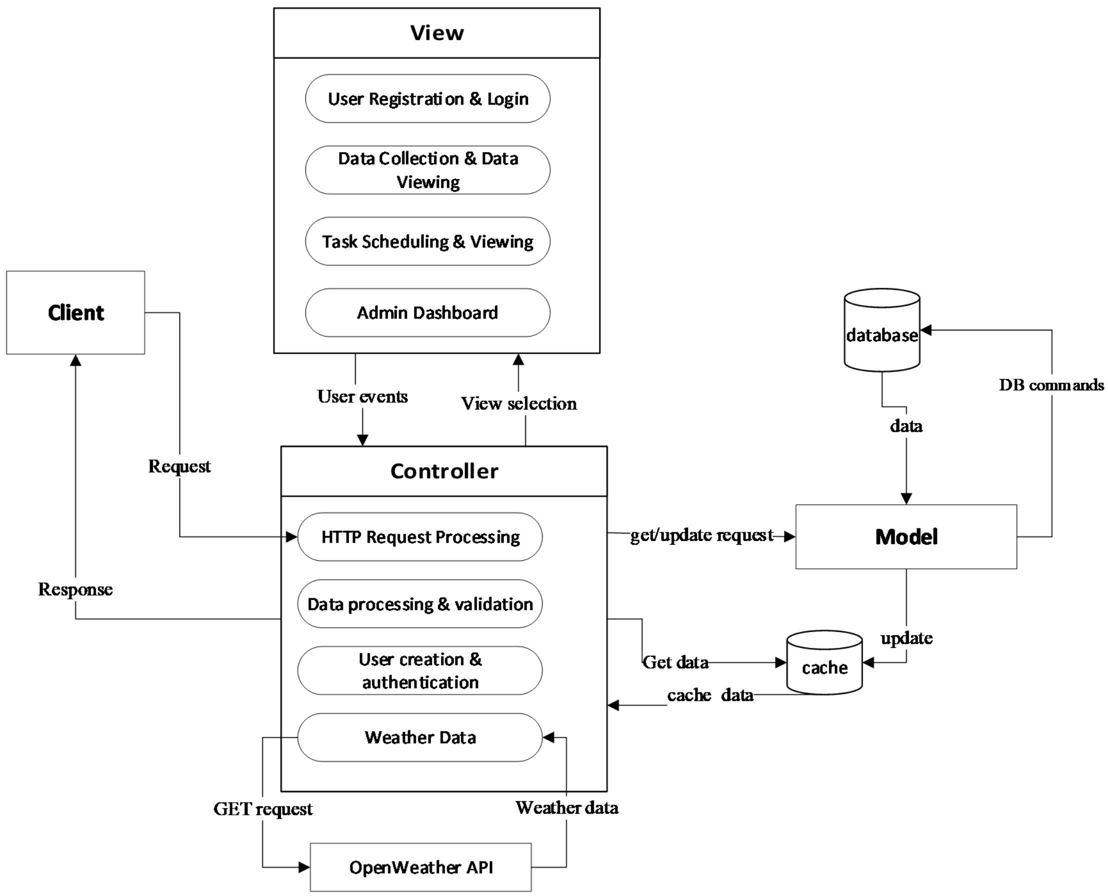

Figure 5: Architectural Framework

require data; views like Data collection and Task scheduling don't require data since they are meant to create one. The controller will search for the view needed based on the current URL and present it as a response. Views like Data Viewing, Task Viewing, and Admin panel require data. When these views request data, the controller will first check for it in the cache. Frequently accessed data will be cached to reduce network calls, avoid re-computation, and reduce database loading. If the data is not present in the cache, the request is forwarded to the Model. The Model delegates with the database and once the data needed is ready, it is sent back to the part of the Controller that requested it. The controller selects the View that requested the data and forwards the data to it. The View formats and integrates the data as part of the user interface (UI) which is sent to the user as a response.

The Admin dashboard view allows the superuser or users with permissions to view the state of the system at any time and manage users and the whole platform.

3.3.1 Relationship between Conceptual Framework and Architectural Framework

Inputs in the conceptual framework namely; User data, farm data collection, and tasks are the views or frontend of the architectural framework; User registration and login, data collection, and task scheduling respectively corresponds to the inputs in the conceptual framework. Processes in the conceptual framework are performed by different parts of the Controller and the Model. Processes make up the backend of the architectural framework. Conceptual framework's Outputs; formatted data and list of tasks and schedule, are diplayed by the data viewing and task viewing modules/views. The user profile, one of the outputs in the conceptual framework, is used by the Controller in the architectural framework to authenticate users and by the Model to allow them to create, alter, update, and select farm and task data.

3.4 Implementation

3.4.1 User Registration and Log in Views

These pages are to be used to create and for logging in users. The forms displayed on these pages will be rendered from the backend. The forms represent the user model and only relevant fields are rendered since fields needed by the registration page are more than what is needed to log in the user. Each model field represents a form input. Bootstrap CSS will be used to style the pages and JavaScript will be used to carry out basic client-side data validation before the form is submitted. To ensure security and prevent Cross-Site Request Forgery (CSRF) attacks. A hidden input field containing a CSRF-token will be rendered together with the form on each page. CSRF-token is a unique, random, and secret value generated by the server-side application. When the form is submitted, the server validates the token first and rejects it if the token is missing or invalid. When the user wants to log out, the logout request will only be received by use of a POST request. This is to prevent unknowingly logging out of users.

3.4.2 Farm Data Collection and Task Scheduling Views

Farm Data Collection and Task Scheduling views provide the user with input fields that allow them to enter farm data and task data respectively. Farm data include crop planted, farm soil data, weather data, farming stage, and any other relevant information needed. Task scheduling views allow the user to create tasks that will be displayed on the Task Viewing page. Task data include name, created on, current status, created by, list of activities to be accomplished and a note. As mentioned before these two views will contain CSRF-token. The Farm data collection page will have multiple form input fields which represent different fields from multiple database tables. HTML, CSS, Bootstrap, and JavaScript will be used to implement these views.

3.4.3 Farm Data Viewing and Task Viewing

These views are responsible for displaying data created by the farmer on Farm Data Collection and Task Scheduling Views. In farm data viewing, data will be presented in a tabular form. This is to make it easier for the farmer to view the data. While in Task Viewing, each task will be a card that shows all the necessary information about a specific task. In farm data Viewing, the farmer will be able to modify, update and filter data, for instance, based on crop type, soil type, farm code, or acreage used. In Task Viewing, the farmer can filter data based on status or created by. Data sent from the Controller will be populated in these views using Django Template Language. Activities like updating and filtering of data will be implemented in JavaScript. JavaScript will listen for these events (For instance, when the farmer clicks a button), and make HTTP requests to the backend with necessary parameters using the JavaScript built-in Fetch function. The backend will respond accordingly. For presentation and styling, CSS and Bootstrap will be used.

3.4.4 Controller and Model

These are the backend.

The Controller will be implemented in Python language, specifically using Django and Django-Rest-Framework. Django-Rest-Framework will be used to handle RESTful requests while the Django framework will be responsible for handling the heavy lifting. Since Django is a framework, it provides a means to implement all the three layers of the project's architectural framework. The project's architectural framework follows the MVC architectural pattern. MVC is a language and framework independent. Django follows the MVC pattern with its own interpretation of the pattern. Django interprets the MVC to MVT (Model-View-Template). According to the Django documentation, Django has concepts of "views" that encapsulate the logic responsible for processing user's requests and for returning the response, concepts of "model" that provide an abstraction layer for structuring the data, and template which provides a designer-friendly syntax for rendering the information to be presented to the user.

Django will provide functionalities that the Controller in the project's architectural framework is supposed to perform. Functionalies like user authentication, URL routing and configuration,middleware that hooks into Django's request and response processing, object-relational mapper and templating system used to render HTML files to the user.

Django allows the creation of custom class-based and function-based views where each view is mapped to a specific URL. The view can handle all the requests made to that URL. It can also be written to handle all the HTTP methods or only the allowed ones and reject the rest. A view can be restricted to only allow logged-in users. Data processing and validation will be implemented as function-based view to handle processing and validating the data before it is stored in the database.

A model represents a single table in the database. PostgreSQL will be used as the database. All the models will be Python class that subclass Django Model class (django.db.models.Model). Each class will have attributes that represent the fields of a table in the database. The Model provides an automatically-generated database-access API. Connection and communication with the database will be handled automatically by Django via *psycopg2*. Psycopg2 is a python adapter for the PostgreSQL database.

When the weather data is requested, an API call will be made to OpenWeatherMap API using a Python module called *requests.* The request module will receive a response and forward the response data to the user and to the Model to save it to the database.

3.4.5 Cache

Caching will be implemented in two forms: Database caching and FileSystem caching. In database caching, all the cache will be stored in the database which requires a fast, well-indexed database server. In this form of caching, I will use Django's *DatabaseCache Backend* and point it to the Cache table in the database. While in Filesystem caching, I will use Django's *FileBased-Cache Backend* and provide it with the filesytem path to where the cache will be stored. The backend serializes and stores each cache value as a separate file.

3.4.6 Tools and Resources

These are the tools and resources that were used in this project:

- Browser.
- Visual Studio Code as an IDE.
- PgAdmin.
- Computer.
- Insomia to test the OpenWeatherMap weather API.
- Django and Django-Rest-Framework.
- PostgreSQL.

3.5 Testing

3.5.1 Accessibility and Usability Testing

These tests were carried out on all views. Accessibility testing is meant to ensure that the project can be used by everyone and ascertain equivalent user experience for people with disabilities and people with mobile phones. Screen readers should be able to read the website. All images should have a proper *alt* text, the flow of information should make sense since screen readers navigate the pages in DOM order, ensure all the interactive elements like links and buttons state their purpose and state and ensure that every website page follows a semantic structure to enable efficient navigation by the assistive technology. (Dodson, 2021).

Usability testing will be carried out to ensure that the platform is effective, efficient, and satisfying to the farmer's needs. Usability will be carried out with actual users who would identify areas that need modification or improvement. Site called *nibbler (https://nibbler.silktide.com)* will be used to carry out this test.

3.5.2 Performance Testing

This test was carried out to primarily test the Controller and the database. A performance test evaluated the speed with which the Controller and database respond, scalability, where I checked for variation in response time/scale as activities in the platform varies, reliability and stability where I validated whether the platform is reliable and stable during prolonged periods where there is high user traffic. This test will be extended to evaluate whether the platform is production-ready and how it performs along with various system configurations. This test will be carried out using Lighthouse found in the browser.

3.5.3 OpenWeatherMap API testing

This test was carried out when the project is integrated with the OpenWeatherMap API. This test was meant to ensure that the API work exactly as expected and the user can consistently connect to the API and get consistent results. Load testing was also performed to ascertain that the API can handle a large number of calls. Insomia was the tool that was used to carry out this test.

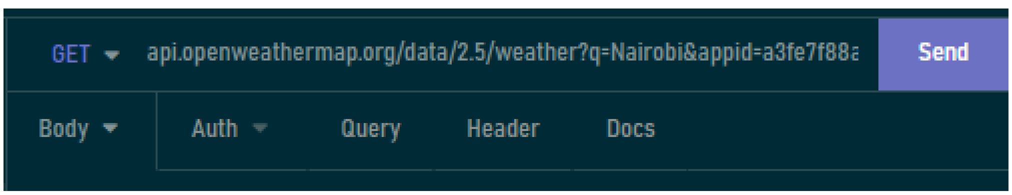

Figure 6: API GET request

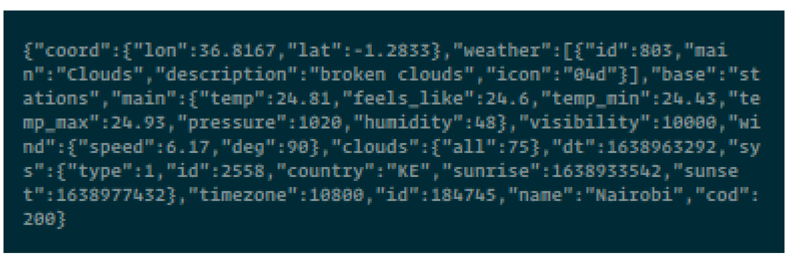
{"coord":{"lon":36.8167,"lat":-1.2833},"weather":[{"id":803,"main":"Clouds","description":"broken clouds","icon":"04d"}],"base":"stations","main":{"temp":24.81,"feels_like":24.6,"temp_min":24.43,"temp_max":24.93,"pressure":1020,"humidity":48},"visibility":10000,"wind":{"speed":6.17,"deg":90},"clouds":{"all":75},"dt":1638963292,"sys":{"type":1,"id":2558,"country":"KE","sunrise":1638933542,"sunset":1638977432},"timezone":10800,"id":184745,"name":"Nairobi","cod":200}

Figure 7: API response

3.5.4 Security Testing

It involved ensuring that the project met security requirements like authentication, permissions, and access control. Penetration testing was carried out using a site called *Pentest-Tools (https://pentest-tools.com)*. This site tested website fingerprinting, SQL-injection, Cross-Site Scripting, remote command execution, the discovery of sensitive files, configuration issues, use of Secure Socket Layer (SSL) that enable the use of the HTTPS protocol, and whether the platform's cookies have *HttpOnly* flag that ensures that session hijacking can't be performed. When Pentest-tools site found any security vulnerabilities, it provided their descriptions and ways to patch them or prevent them.

4 Implementation

4.1 Database Tables

Login

- Login_id (PK)
- Login_username
- Login_password
- Login_rank

Farmer (Owner)

- Farmer_id (PK)
- Farmer_first_name
- Farmer_last_name
- Farmer_national_id
- Farmer_phone_no
- Farmer_login_id (FK)

Farm

- Farm_id(PK)
- Farm_name
- Farm_location
- Farm_area (acre/ha)

- Farm_Farmer_id

Worker

- Worker_id (PK)
- Worker_first_name
- Worker_last_name
- Worker_national_id
- Worker_phone_no
- Worker_login_id(FK)

Soil

- Soil_id(PK)
- Soil_pH
- Soil_color
- Soil_texture
- Soil_structure
- Soil_depth
- Soil_testing_date
- Soil_last_testing_date
- Soil_next_testing_date
- Soil_status
- Soil_notes
- Soil_farm_id(FK)

Input_category

- Input_category_id (PK)
- Input_category_name

- Input_category_desc

Input_product

- Input_product_id (PK)
- Input_product_name
- Input_product_desc
- Input_product_unit_price
- Input_product_total_units
- Input_product_total_cost
- Input_product_unit_weight
- Input_product_unit_measurement (kg, g, liter, bag)
- Input_product_unit_rate (kg/acre, L/acre)
- Input_product_Input_category_id(FK)

Input_Inventory

- Input_Inventory_id(PK)
- Input_Inventory_name
- Input_Inventory_desc
- Input_Inventory_ref_code
- Input_Inventory_created_on
- Input_Inventory_updated_on

Input_Inventory_item

- Input_Inventory_item_id (PK)
- Input_Inventory_item_Input_product_id(FK)
- Input_Inventory_item_quantity
- Input_Inventory_item_total_cost

- Input_Inventory_item_Input_Inventory_id (FK)

Crop

- Crop_id (PK)
- Crop_name
- Crop_desc
- Crop_variety
- Crop_maturity_duration

Farming_stage

- Farming_stage_id(PK)
- Farming_stage_name
- Farming_stage_desc

Farming_season

- Farming_season_id (PK)
- Farming_season_name
- Farming_season_desc
- Farming_season_start_date
- Farming_season_end_date
- Farming_season_Farm_id (FK)
- Farming_season_Crop_id (FK)
- Farming_season_yields

Farm_Task

- Farm_Task_id(PK)
- Farm_Task_name
- Farm_Task_start_date
- Farm_Task_deadline

- Farm_Task_created_by
- Farm_Task_objectives
- Farm_Task_notes
- Farm_Task_status
- Farm_Task_expected_expenses
- Farm_Task_Farm_id(FK)
- Farm_Task_Input_product_id(FK)
- Farm_Task_Input_product_units_used
- Farm_Task_Farming_season_id(FK)
- Farm_Task_Worker_id(FK)

References

[1] Rogers, N. (2014, January 10). WHAT IS PRECISION AGRICULTURE? Retrieved October 8, 2021, from https://sustainableamerica.org: https://sustainableamerica.org/blog/what-is-precision-agriculture/

[2] Byju's. (n.d.). Vertical Farming. Retrieved October 8, 2021, from https://byjus.com: https://byjus.com/free-ias-prep/vertical-farming/

[3] Aker. (2021, October 8). Home: Our Mission. Retrieved October 8, 2021, from https://aker.ag: https://aker.org

[4] Sciforce. (2020, June 22). https://iotforall.com. Retrieved October 8, 2021, from Smart Farming: The Future of Agriculture: https://www.iotforall.com/smart-farming-future-of-agriculture

[5] Tower Farms. (n.d.). Microsoft Commons. Retrieved October 8, 2021, from https://towerfarms.com: https://www.towerfarms.com/us/en/possibilities/indoor-farming/microsoft-commons

[6] United States Environmental Protection Agency (EPA), U. S. (2021, October 23). Learn About Sustainability. Retrieved October 23, 2021, from https://epa.gov: https://www.epa.gov/sustainability/learn-about-sustainability#what

[7] Goal 12 | Department of Economic and Social Affairs, Sustainable Development. (2021). Retrieved 23 October 2021, from https://sdgs.un.org/goals/goal12

[8] Kovács, I., & Husti, I. (2018). The role of digitalization in the agricultural 4.0–how to connect the industry 4.0 to agriculture?. Hungarian Agricultural Engineering, (33), 38-42.

[9] Schahczenski, J., & Hill, H. (2008). Agriculture, Climate change and Carbon Swqestration. (H. Michels, Ed.) Retrieved October 22, 2021, from http://www.attra.ncat.org/: http://www.attra.ncat.org/attra-pub/PDF/carbonsequestration.pdf

[10] O'S, J. (2013, September 4). Cold comfort farms. Retrieved October 29, 2021, from https://www.economist.com/: https://www.economist.com/baobab/2013/09/04/cold-comfort-farms

[11] UjuziKilimo. (2021). How data can help smallholder farmers adopt sustainable practices. Retrieved October 19, 2021, from ujuzikilimo.com: https://www.ujuzikilimo.com/article/HOW-DATA-CAN-HELP-SMALLHOLDER-FARMERS-ADOPT-SUSTAINABLE-PRACTICES

[12] Sharma, R. (2020, April 28). Big Data Applications in Agriculture: Role, Importance & Challenges. Retrieved October 22, 2021, from https://www.upgrad.com/: upgrad.com/blog/big-data-applications-in-agriculture/

[13] Majumdar, J., Naraseeyappa, S., & Ankalaki, S. (2017, July 5). Analysis of agriculture data using data mining techniques: application of big data. Journal of big data, 1-2.

[14] Cruchbase. (2021). Apollo Agriculture. Retrieved November 1, 2021, from https://cruchbase.com: https://www.crunchbase.com/organization/apollo-agriculture.

[15] Bosilkovski, I. (2020, May 29). Meet The Agritech Entrepreneurs Who Just Raised $6 Million To Help Farmers In Kenya Grow Their Business. Retrieved November 1, 2021, from https://www.forbes.com/:

https://www.forbes.com/sites/igorbosilkovski/2020/05/29/meet-the-agritech-entrepreneurs-who-just-raised-6-million-to-help-farmers-in-kenya-grow-their-business/

[16] NavFarm. (2021). HOW NAVFARM WORKS FOR. Retrieved November 1, 2021, from https://navfarm.com: https://www.navfarm.com/dairy

[17] SoftwareConnect. (2021). Conservis | Farm Management Software 2021 Reviews, Pricing. Retrieved November 5, 2021, from https://softwareconnect.com/: https://softwareconnect.com/farm-management/conservis/

[18] Conservis. (2021). Plan for the long haul. Retrieved November 5, 2021, from https://conservis.ag/: https://conservis.ag/farm-management-software/permanent-crop-management-software/

[19] Sommerville, I. (2016). Software Engineering. Edinburgh Gate: Pearson Education Limited.

[20] Dodson, R. (2021). Retrieved December 30 2021, from https://developers.google.com/: https://developers.google.com/web/fundamentals/accessibility/how-to-review

[21]