#### UNIVERSITY OF BUEA

**\*\*\*\*\*\*\*\*\*\*\*\*\*\***

#### FACULTY OF ENGINEERING AND TECHNOLOGY

**\*\*\*\*\*\*\*\*\***

#### DEPARTMENT OF COMPUTER ENGINEERING

**\*\*\*\*\*\***

#### 2022/2023 ACADEMIC YEAR

PROJECT REPORT TITTLED;



Mobile Application Development

Hybrid, Native, and Web Application development

**GROUP MEMBERS SUPERVISOR;**

|  |  |
| --- | --- |
| **NAMES** | **MATRICULES** |
| **ENEMBE VERINE** | **FE20A033** |
| **AYAMBA OJONG NKONGHO** | **FE20A019** |
| **TONY KENYUI NANGAH** | **FE20A120** |
| **NKONGHO AYUK SHARIMA** | **FE20A090** |
| **GABILA YASMIN NAHKUNA** | **FE20A043** |

**Dr. Valery Nkemini**

#### FRIDAY, MARCH 31st 2

# TABLE OF CONTENTS

Chapter 1

Background and context of the study……………………………………1

Problem statement…………………………………………………………..2

Objectives of the study…………………………………………………….2

General objectives…………………………………………………………..2

Specific objectives…………………………………………………………..2

Proposed methodology……………………………………………………..2

Scope of study……………………………………………………………….3

Delimitation of the study…….………………………………………………3

Chapter 2

Definition…………………………………………………….3

Different mobile applications…………………………………4

Native apps……………………………………………………..4

Advantages……………………………………………………..4

Disadvantages…………………………………………………..4

Web apps……………………………………………………….

Advantages……………………………………………………

Disadvantages………………………………………………….

Hybrid apps……………………………………………………

Advantages…………………………………………………….

Disadvantages………………………………………………….

Review on different mobile programming languages

For IOS……………………………………………………….

For android…………………………………………………….

For web apps……………………………………………………

Review of mobile app development frameworks……………………….

Flutter framework……………………………………………………….

Xamarin framework………………………………………………………

Iconic framework………………………………………………………….

React native framework ……………………………………………………

Apache cordova framework……………………………………………….

Native scripts………………………………………………………………

Chapter 3

Introduction……………………………………………………………….

Proposed methodology……………………………………………………

Analysis…………………………………………………… …………….

System architecture………………………………………………………….

Software requirement……………………………………………………….

Functional requirements…………………………………………………….

Non-functional requirements…………………………………………………

System requirements………………………………………………………..

User requirements………………………………………………………….

Donor requirements………………………………………………………..

Admin requirements……………………………………………………….

Collector requirements…………………………………………………….

UML Design……………………………………………………………

Class diagram…………………………………………………………….

Use case diagram…………………………………………………………

Activity diagram………………………………………………………….

Sequence diagram………………………………………………………..

UI Design……………………………………………………………..

UI Design criteria………………………………………………………

Screens…………………………………………………………….. …

Database design……………………………………………………..

ER diagram……………………………………………………………

Relational schema………………………………………………………

Logical schema…………………………………………………………

Physical schema…………………………………………………………

Description……………………………………………………………….

Global architecture of the solution………………………………………

Description of the resolution process……………………………………

Partial conclusion……………………………………………………….

Chapter 4

Introduction…………………………………………………………….

Tools and materials…………………………………………………….

Description of implementation process…………………………………

UI implementation…………………………………………………………

Database implementation……………………………………………………

Presentation and interpretation of results………………………………..

Evaluation of the solution……………………………………………….

Partial conclusion……………………………………………………….

Chapter 5

Summary and finding……………………………………………………

Contribution to engineering and technology……………………………

Recommendation……………………………………………………….

Difficulties encountered…………………………………………………

Further works……………………………………………………………

**Table of figures**

Figure 1………………………………………………………………………..

Figure 2……………………………………………………………………….

Figure 2…………………………………………………………………………..

Figure 4…………………………………………………………………….

Figure 5…………………………………………………………………………

Figure 6………………………………………………………………………….

Figure 7……………………………………………………………………….

Figure 8……………………………………………………………………

Figure 9……………………………………………………………………..

Figure 10…………………………………………………………………….

Figure 11

**CHAPTER 1: GENERAL INTRODUCTION**

1. BACKGROUND AND CONTEXT OF THE STUDY

Internet programming refers to creating software that runs on the internet. This involves developing web applications that can be accessed through a web browser or mobile device. Mobile programming refers to creating software applications that can run on mobile devices such as smartphones and tablets. It can be done for different operating systems such as Android and IOS. Developers use various programming languages such as java, kotlin, and swift. The choice of programming language depends on the platform and the developer’s preference.

The mobile app development life cycle consist of various stages such as

* Planning: this initial stage involves defining the app idea, setting goals and objectives and identifying target audience and creating a roadmap for the project.
* Analysis and design: this stage involves analyzing the requirements and designing the UI, features and functionalities.
* Development: This stage involves coding the app .Different programming languages, frameworks and technologies can be used to build the app.
* Testing: This stage involves testing the app for bugs, errors and usability issues using testing types such functional and performance testing.
* Deployment: once the app is tested and ready, it is deployed to the app store
* Maintenance: Maintenance and support is provided to fix any issue that arises and to update the app with new features.

1. PROBLEM STATEMENT

The food industry is a critical sector of the global economy, providing essential goods and services to the people around the world. However, the industry faces numerous challenges including food waste. A food waste management system is a comprehensive approach to addressing these challenges. Food management refers to the comprehensive approach of managing food production, distribution and consumption. One of the primary goals of food management is to minimize food waste. This can be done in various ways such as planning your meals and ensuring that you buy only what you need and also donating excess food. The system works by providing a platform for people to upload foodstuff which they no longer need for free or at reduced prices and enabling people who are in need of food to search for available food, thereby reducing the waste of food.

1. OBJECTIVES OF THE STUDY

This report aims to explore the design and implementation of a food waste management system, examining key features and functionalities required to create a successful system.

1. GENERAL OBJECTIVE

This study aims to create a platform which in an efficient way connects people who have excess food and are willing to give it out free or at a reduced price and consumers who are in need of food and are willing to take this food Thereby reducing food waste as excess or unsold food is redirected to consumers who need it.

1. SPECIFIC OBJECTIVES

This includes developing the software and infrastructure needed to implement the food auctioning system, testing and validating the system to ensure that it is user-friendly and secure.

1. PROPOSED METHODOLOGY

The methodology for designing and implementing a food waste management system is iterative as the development team will revisit earlier steps as necessary based on feedback and changes in requirements. This may include the following steps;

* Requirement gathering: this involves identifying the needs of the various stakeholders. This could be done through surveys and interviews.
* Analysis and planning: based on the requirements and needs identified, the development team will analyze the information gathered and plan the design and development of the system. This may include defining the system architecture and selecting the appropriate technologies.
* Design and prototyping: A prototype of the system is created including the UI , database schema and system flow.
* Development and testing: after the design is finalized, the team will begin coding the system. They will perform unit and integration testing to ensure that the system is functioning as expected.
* User acceptance testing: in this step, the system will be tested by the end-user to ensure that it meets their needs and requirements. Feedback of the end-user will be incorporated into the final design.
* Deployment: once the system is fully tested and approved, it will be deployed to production.
* Maintenance and support: after the system is deployed, the development team will provide maintenance and support services to ensure that the system is functioning properly and any issues are resolved in a timely manner.

1. SCOPE OF THE STUDY

This refers to the extent and limitations of the project, including the specific focus of the study, the target audience and research objectives. It helps to ensure that the research is focused, feasible and relevant to the intended audience. The scope of this project includes;

1. Food types: This study will focus on specific or multiple food types depending on the needs and requirements of stakeholders. The system may be designed to handle different types of food
2. Geographic scope: this study may focus on a specific geographic region or be designed for a global audience. The scope of the study will depend on the target audience and the needs of stakeholders.
3. Technical scope: the study will focus on the technical aspects of designing and developing the food waste management system. This may include selecting appropriate technologies, designing the system architecture and developing the user interface.
4. Business scope: this may include analyzing the potential business models for the food waste management system such as commission-based revenue models or subscription-based models.
5. DELIMITATION OF THE STUDY

This refers to the specific boundaries and limitations of the study. It helps to define the scope of the study and establish clear expectations for what the study will and will not cover. Some delimitations of this project include;

1. Timeframe: the study may be limited to a specific timeframe such as a certain number of months or years depending on the available resources and objectives of the study.
2. Geographic area: the study may focus on specific areas such as a city, region or country. It may be limited to this area to ensure that the study is feasible and relevant to the target audience.
3. Technical limitations: the study may be limited to technical factors such as the availability of specific technologies or the capacity of the development team.

**CHAPTER 2: LITERATURE REVIEW**

1. DEFINITION

A mobile app (or mobile application) is a software application developed specifically for use on small, wireless computing devices, such as smartphones and tablets, rather than desktop or laptop computers. There are different types of mobile applications. Members of the mobile app development community agree that there are four types. They are Native apps, hybrid apps, web apps, and progressive web apps.

In today’s digital world, having a strong mobile presence has become essential for businesses. Apps provide businesses with the opportunity to reach a larger customer base, improve customer engagement, increase productivity, enhance user experience, and increase revenue.

Native apps are created for one specific platform or operating system.

Web apps are responsive versions of websites that can work on any mobile device or OS because they’re delivered using a mobile browser.

Hybrid apps are combinations of both native and web apps, but wrapped within a native app, giving it the ability to have its own icon or be downloaded from an App Store

A progressive web application is a type of application software delivered through the web, built using common web technologies including HTML, CSS, JavaScript, and Web Assembly. It is intended to work on any platform with a standards-compliant browser, including desktop and mobile devices.

1. DIFFERENT MOBILE APPLICATIONS
2. NATIVE APPS

Native apps are built for a dedicated operating system on a device. For example, Android apps and iOS apps. You can’t mismatch them. That is you cannot use an iOS dedicated app on an Android OS. Can be coded using simple programming languages like C/C++ Python Java Objective C. For Android OS, this android coding language is Java.

1. Advantages

• Faster and reliable performance.

• Native apps connect with the device hardware hence offer capabilities like Bluetooth, Camera Roll Contacts. Etc

1. Disadvantages

• Costly as in a code created for one platform can not be reused in another platform. Duplicate effort for each platform.

• Whenever there’s a new version, the user has to update and install it his device. Meaning native apps take a good amount of device storage.

1. WEB APPS

Web apps behave similarly to native apps but are accessed via a web browser on your mobile device. They’re not standalone apps in the sense of having to download and install code into your device. They’re actually responsive websites that adapt its user interface to the device the user is on.

Technologies used are HTML, JavaScript, CSS.

1. Advantages

• Because it’s web based, no need to customize to a platform or OS

• There’s nothing to download also. They won’t take up space on your device memory like a native app, making maintenance easier

1. Disadvantages

• Web apps are entirely dependent on the browser used on the device. There will be functionalities available within one browser and not available on another, possibly giving users varying experiences.

1. HYBRID APPS

And then there are the hybrid apps. These are web apps that look and feel like native apps. They might have a home screen app icon, responsive design, fast performance, or even be able to function offline, but they’re really web apps made to look native.

Technologies used are Objective C, Swift, HTML5, and others.

1. Advantages

• Building a hybrid app is much quicker and more economical than a native app. They also load rapidly, are ideal for usage in countries with slower internet connections

1. Disadvantages

• Hybrid apps might lack in power and speed, which are hallmarks of native apps

According to recent statistics, Native apps are the most commonly used type of mobile application. They are specifically designed for particular mobile platforms and typically offer a better user experience and performance because they have access to the devices hardware and software features

1. REVIEW ON THE DIFFERENT PROGRAMMING LANGUAGES

The choice of a particular method may greatly depend on your business goals and the corresponding capabilities of the operating system you choose as the platform, the type of application, and its requirements. Having an idea of app languages for creating mobile applications is useful not only for developers but also for their customers who want to better understand the project processes and know which language to pick best for their future projects. Let us review the main programming languages for iOS and Android mobile development here.

1. FOR iOS
2. Objective-C & Swift

These are the key languages for writing iOS apps. They are object-oriented and make it possible to group similar tasks in the coding process, which significantly speeds up and simplifies the work of developers during front-end mobile app development. Besides, you may also apply:

1. C#
2. HTML5
3. Java
4. FOR ANDROID

We have;

1. Python

Android developers who write mobile apps in the Python programming language are rare, but still, they do exist and have great Support from the huge programming community.

1. C / C ++

C languages are positioned as high-performance languages, and this is important, especially if we are talking about something heavy like a mobile 3D game. Without this language, you cannot create a full-fledged mobile application.

1. Java

Java mobile app development via java language – a strongly typed general-purpose object-oriented programming language is a popular thing to do. It often takes the highest places in the popularity ratings (2nd place in the IEEE Spectrum (2020) ratings, and on TIOBE (2021) alongside Python.

1. FOR WEB APPS

We have;

1. Scala

Scala combines object-oriented and functional programming in one concise, high-level language. Scala’s static types help avoid bugs in really complex applications.

1. JavaScript

This one is an interface language used for creating and developing websites, desktop applications, and games. JavaScript works in all browsers and can work with programs that are not hosted on the Internet.

1. REVIEW OF MOBILE APP DEVELOPMENT FRAMEWORKS

A mobile app development framework is a software framework that allows mobile app developers to build their applications. They are templates or fundamental structures that facilitate the work of developers as they build and optimize apps. Benefits of working within these rule-based templates or frameworks, it means developers are restricted in a way that helps them avoid making time-consuming programming errors so they can work faster and more efficiently.

1. FLUTTER MOBILE APP DEVELOPMENT FRAMEWORK

Created by Google, it is a completely free and open-source mobile UI framework. It is one of the best mobile app development platform. The good thing about flutter is that you don’t have to write two separate codes, you can write one code and it will work for both Android and IOS.

1. Benefits

- It is easier to use

- Flutter allows you to create an Android and IOS application with one codebase.

- Saves time and decreases cost.

- Has built in functionalities to make your work easy and build the best mobile apps.

1. XAMARIN MOBILE APP DEVELOPMENT FRAMEWORK

XAMARIN is an open source framework created by Microsoft for development of native and good-performance Android and iOS apps. It is one of the best in the market.

1. Benefits

-Xamarin provides one code for all platforms.

- Its performance is improved daily to completely match the standard of native application development.

1. IONIC MOBILE APP DEVELOPMENT FRAMEWORK

Ionic is an open-source framework that allows developers to build iOS and Android apps for flawless mobile performance. Developers can build a strong and well-featured native application with the help of this framework. Developers can use various UI elements like forms, navigation menus, filters, views, and actions sheets in the application design which are the most significant features of Ionic. It is a complete platform that allows developers to create PWAs (Progressive Web Apps), cross-platform, and hybrid mobile applications.

1. Benefits

- You can create mobile apps by using any front-end framework you want

- It is great for beginners because it allows you to create mobile applications if you know HTML, CSS, and JavaScript.

1. REACT NATIVE MOBILE APP DEVELOPMENT FRAMEWORK

React Native is an open source mobile app development javascript framework created by Facebook.

1. Benefits

- Currently, React Native is preferred by a lot of companies as a solution because one can easily write all of the code in Javascript and share it on Android and iOS.

- It is one of the best frameworks

- Developers who use React Native, they don’t have to create separate codes for different platforms as JavaScript can be used for both Android and iOS platforms

- It is efficient and cost less

- React Native upgrades regularly to stay at the top of all mobile app development frameworks and this is the reason for its popularity.

1. APACHE CORDOVA MOBILE APP DEVELOPMENT FRAMEWORK

It is one of the best develop-friendly app development frameworks and uses CSS3, JavaScript, and HTML5 for the creation of mobile applications. Its plugins enable developers to access the hardware capabilities of devices like Cameras, GPS, and accelerometers for the delivery of a native-like app experience.

1. Benefits

- It is great for Hybrid applications as it can make Hybrid development easy and super fun.

- It is not just an HTML application that runs in a browser, it lets you write native plugins that work with all supported platforms

1. NATIVE SCRIPTS

It is an open-source framework to create native mobile applications empowered with Angular, Typescript, JavaScript, CSS, and Vue.js.

1. Benefits

- Native Script is a preferable framework to reduce the code and time of the app loads on the system.

- Many leading companies love using Native Scripts for its strong web empowerment Platform

- Full direct access to Android & iOS APIs

**CHAPTER 3: ANALYSIS AND DESIGN**

1. INTRODUCTION

The analysis and design of a food auction system involves a systematic approach to understanding the needs and requirements of the system, and developing a solution that meets those needs. This process typically involves several key steps.

1. PROPOSED METHODOLOGY

In order to realize this project, the agile methodology was used. This is an iterative approach to software development that emphasizes flexibility, adaptability and customer satisfaction.it is well-suited for projects with changing or uncertain requirements, as it allows for more flexibility. It is characterized by the following;

1. ITERATIVE AND INCREMENTAL

The software is developed in a short time, time-boxed iteration, with each iteration delivering a working and potentially shippable product increment.

1. CUSTOMER COLLABORATION

Customers are involved throughout the development process, providing feedback and insight to help guide the development process.

1. EMPHASIS ON WORKING SOFTWARE

The focus is on delivering a working software, rather than just documentation or other artifacts.

1. CONTINUOUS IMPROVEMENT

The team regularly reflects on their work and processes, looking for ways to improve and adapt to changing requirements.

1. ANALYSIS

In the analysis phase of this project, several aspects could be discussed which includes;

1. SOFTWARE REQUIREMENTS

The technologies used for this application are java,Kotlin, SQL and firebase

The installation and setup was done using Android studios, canvas and starUml

1. FUNCTIONAL REQUIREMENTS
2. Registration functionalities:

Registration of our users as dual functionalities, users can create, edit and delete functionalities. The registration process is short and flexible for the users. As attached to the registration process users can login and use the forgot password options. User account verifications and validation. The Registration is also done at the level user financial account, massaging, delivery account

1. Third-Party Integrations

Indicate which third-party software you want to add to your new application. Or maybe you’re replatforming and wish to preserve the system you’re using now. This requirement concerns both systems streamlining financial operations (like ERP, CRM, PIM) and flexible payment gateways for your users. Specifying the number of third-party integrations will make your [financial architecture](https://elogic.co/blog/ecommerce-architecture/) structured and ready for your business scaling in the future.

1. Mobile-Friendliness

It’s no secret that mobile apps (hybrid) turn to have more traffic [Statista](https://www.statista.com/statistics/234884/us-online-shopper-conversion-rate-by-device/), the number of customer conversions on mobile devices has also reached those on desktops in the US. So investing into the hybrid nature of your mobile application can earn you more than a few bucks, plus loyal users making deals at the comfort of their sofas.

1. Product attributes

Your PDP will include various product characteristics, and the development agency should know about them to implement the corresponding features. Will the users be able to choose multiple food auctions per day ? Do you use videos on a PDP? Will some product attributes appear in a menu (as in the [mega-menu](https://elogic.co/blog/review-of-mega-menu-magento-2-extension-for-better-website-navigation/))? If possible, write out a list of all the services of our application that is does available to the user.

1. Request and checkout flow

The functional requirements for an online auctioning system should specify how the requests are processed in your application and whether this functionality should be optimized. In particular, indicate whether you want the user to register to make a request of auction or enable guest checkout. List the order statuses you want to have

1. Social sharing

Online presence goes alongside social media presence in ecommerce. Allowing a user to share your app’s content on social media leads to higher brand awareness and brings you closer to your present and potential customers. Research your target audience and identify their favorite social media networks. Let your consumers share products, blog posts, and inspirational pictures by adding a corresponding button to their website.

1. NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements articulate the quality attributes of the mobile application that build positive user experience and optimal mobile performance. Here are some basic types of non-functional requirements that should make it to the website specification document of all ecommerce businesses.

1. Usability

No matter the size of your application, you want your application to be intuitive and easy-to-use. It takes [about 0.05 seconds](https://www.tandfonline.com/doi/abs/10.1080/01449290500330448) for users to figure out if your application is worth their time and attention. So you’ll definitely want to work on your homepage design, calls-to-action, and easy checkout to get past those milliseconds of doom.

1. Security

Security is paramount while dealing with monetary transactions and sensitive data. A simple hashing encryption and data privacy policy will instill trust into your application and convert the customers into your brand advocates. It is also about different admin roles allowing you to control who can create, see, copy, change, or delete information. Depending on your business location, security also means complying with the customer data protection rules (case in point: GDPR in Europe).

1. Performance

If your goal is increasing your application traffic, performance should be the priority NFR in your specification document. This NFR is often found in briefs from large enterprises and a number of integration websites with legacy architecture: they want their e-stores to load fast no matter the ns and sales seasons. Set up the speed benchmark, a maximum number of SKUs to be added or any other performance indicator suitable to your business. Don’t include third-party system delivery time, though; your developers can’t do much if a certain business operation depends on an API call to another database.

1. Maintainability

It’s widely known that the tricky part of planning a business budget is accounting for the operational costs of business maintenance. Striving to make the website maintainable from the initial development phase means cutting the time and cost to identify and resolve the system faults in the future. But your task is to make the system easy-to-maintain right from its launch.

1. Scalability

If you’re looking into a future-proof solution, scalability should be your take. This requirement defines how the application can grow and expand its functionality without affecting its performance. You should be able to add more memory, servers, or disc space to complete more transactions on your website.

1. SYSTEM REQUIREMENTS
2. The system consists of three views: donor(Auction, charity), food\_collector, Admin.

Food donor could be an individual organization, a food donor can also collect food as well as collectors can also donate food (on auction or charity).

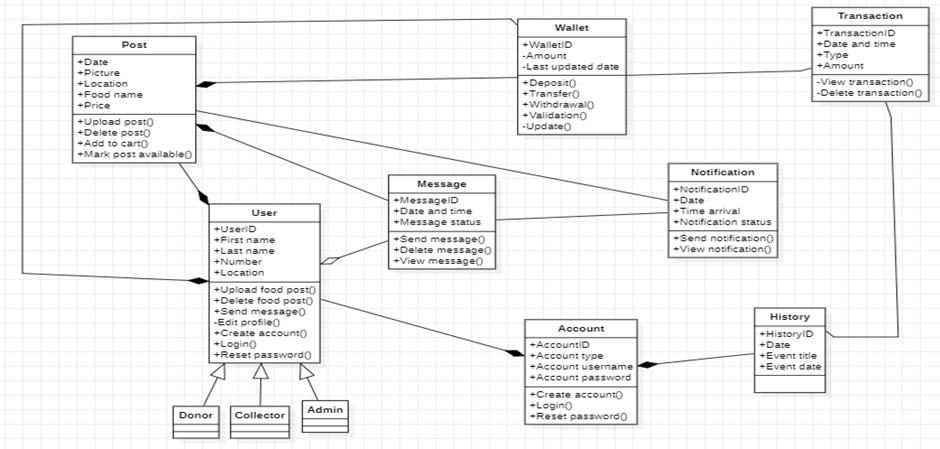
1. Admins: They control all the activities, manage user data, system errors and bugs, manage donation post and system transactions
2. Donors (organizations, individuals): They can post food they are given out either for free(charity) of small price (auction price) collectors are
3. from donors at arranged locations to allow for a reduction in the price.
4. Collector: They collect food stipulated agreements
5. Each user should have an account.
6. Each use has a wallet account
7. The application provides signup, login and logout functionalities.
8. USER REQUIREMENTS
9. Each and every user has ability to auction and take food on auction
10. Each user has an account
11. Each user can register to the application. Users can delete their account from the auction food sides if not in use
12. Users can take many donation food at the same time
13. Users can make collections
14. Users can build and update profile
15. Users can exchange messages
16. Users share food donation post
17. DONOR FEATURES
18. Donors can make food donation advertisement on available food
19. Donors delete food donation post
20. Donors can make food donations free or open for minimum auctioning
21. Donors can mark auctioning debatable or not (enable negotiations, disable negotiations).
22. Donor can view the donation history
23. Donors view collector profile
24. ADMIN FEATURES
25. Admins can delete food donation post
26. Admins can manage user data in database
27. Admins can view all the pending donations along with status.
28. Admins can view all the donations that they have received.
29. Admins can also view all the agents in the application.
30. Admins can delete user account
31. Admins can delete Food request
32. COLLECTOR FEATURES
33. Collector will receive notifications from admins to collect food from donor's homes.
34. Collectors can mark their collection upon collection of food from the donor's home.
35. Collectors can also view all those food donations which have been collected by them previously.
36. Collectors can recommend Donors.
37. Collectors can make food request
38. UML DESIGN

The design phase of this project looks at various phases such as

1. CLASS DIAGRAM

The flow and connection of the system can be shown using a class diagram. The class diagram below shows the connection and dependencies between the classes. The class diagram can be used to generate the ERD of the system.

1. Primary actors: this a person or an entity that interacts directly with the system to achieve a specific objective. They include; Donor, Collector and Admin.
2. Secondary actors: this is a person or an entity that interacts indirectly with the system to support the primary actors in achieving their objectives. They include; System, Admin, Third-party.
3. Classes: they are constructs used to define objects and behaviors within a system. This includes; User, Collector, Donor, Admin, Post, History, Messaging, Notification, Wallet, Transaction.
4. Diagram

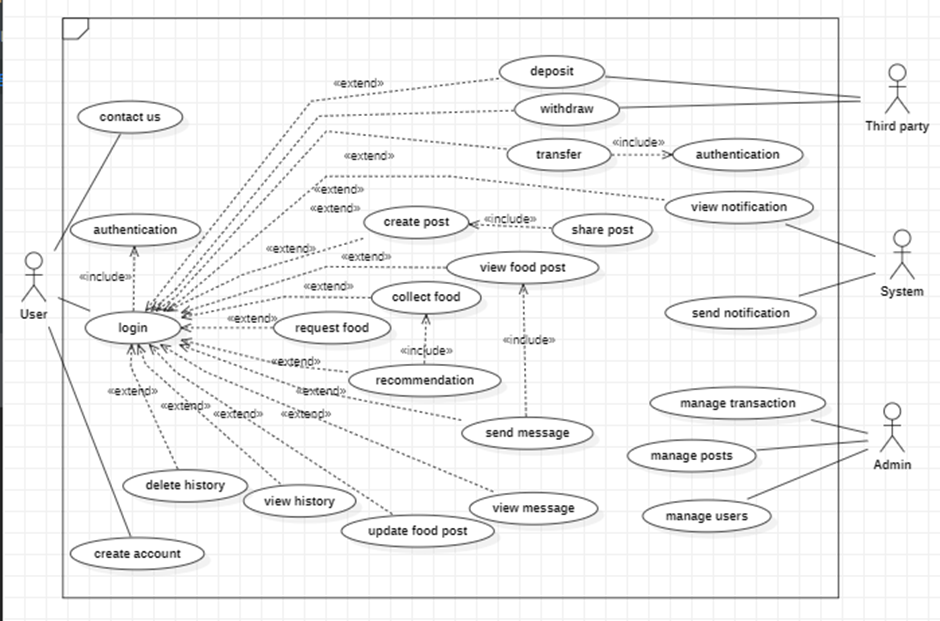


**Figure 1**

1. USE CASE DIAGRAM

The use cases represent the processes of our system. The various use cases are: Registration, Login, Logout, Forgot password, Create food post, Collect food post, Make deposit into wallet, Recommend Donor, Messaging(chat), Upload photo.

The interaction between these use cases and their constraints is shown by the help of a use case diagram.

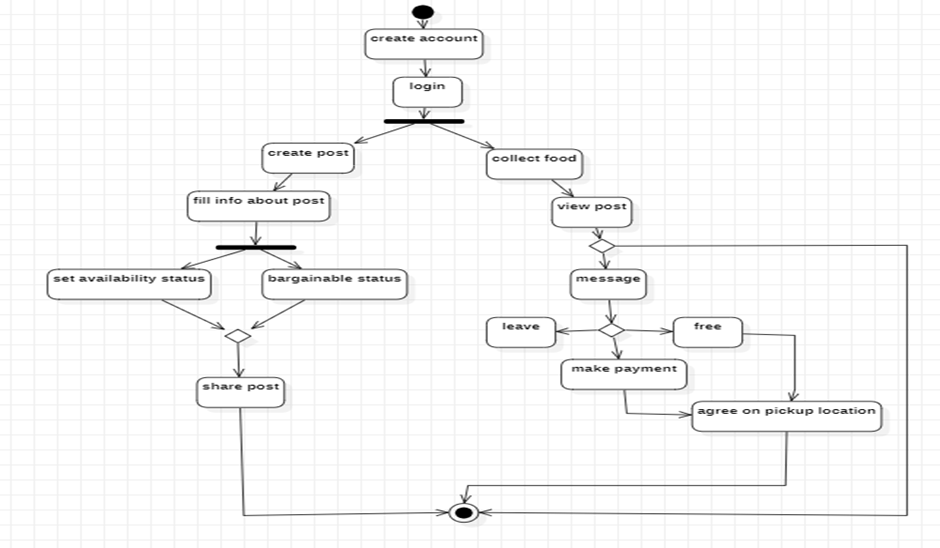


**Figure 2**

1. ACTIVITY DIAGRAM

To give us a behavioral view we decided to model the flow of the system in terms of the two major activities of the system in an activity diagram.

Activity diagram to collect and donate food



**Figure 3**

1. SEQUENCE DIAGRAM

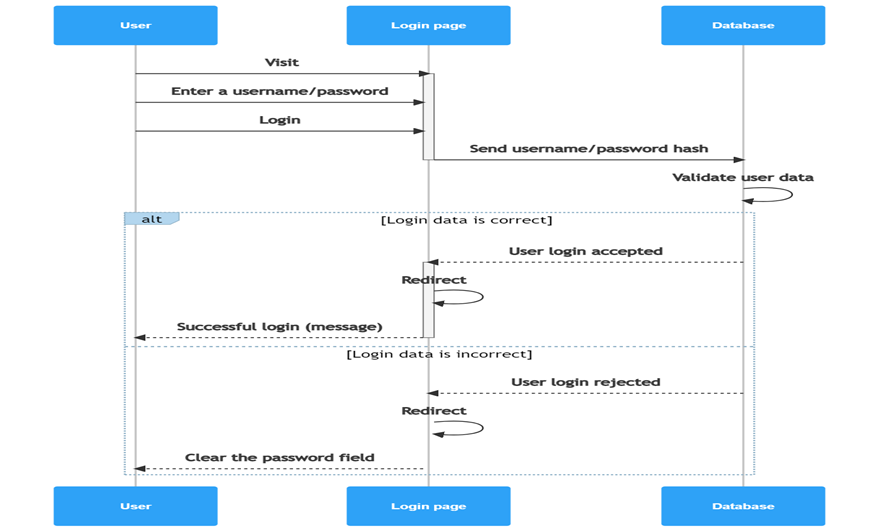
The Development of our system needs detailed description and so we try to implement most of our meaningful use cases as sequence diagrams to show the flow of each use case.

1. Registration: The sequence diagram explains the registration process of our application.

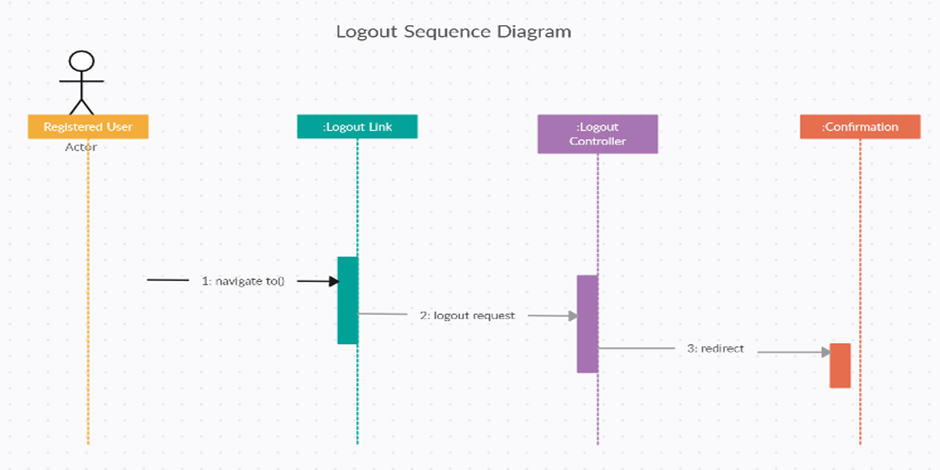


**Figure 4**

1. LOGIN/LOGIN: This sequence diagram explains the login and logout process of our application.

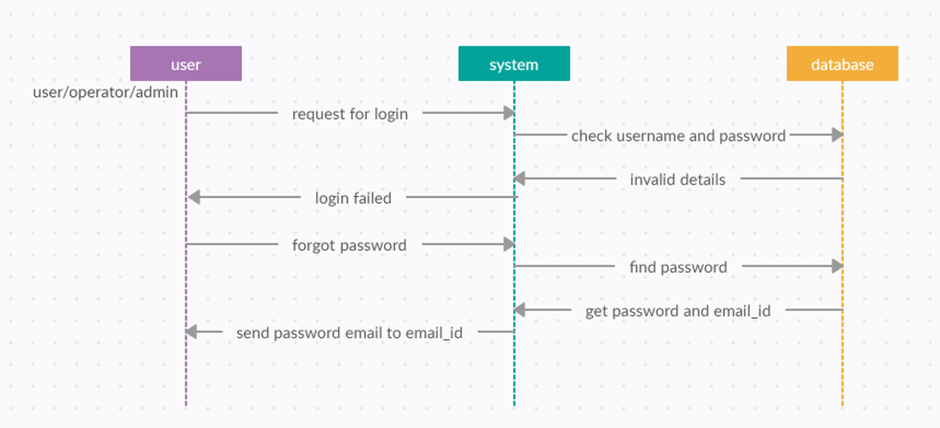


**Figure 5**



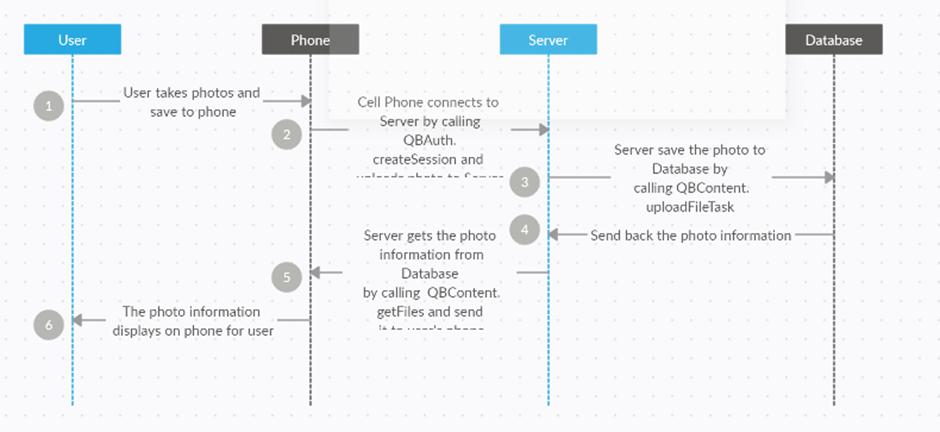
**Figure 6**

1. Forgot password: The system comes with a forgot password to ensure security, reliability, and usability we model this into a sequence diagram to show the follow of the process.



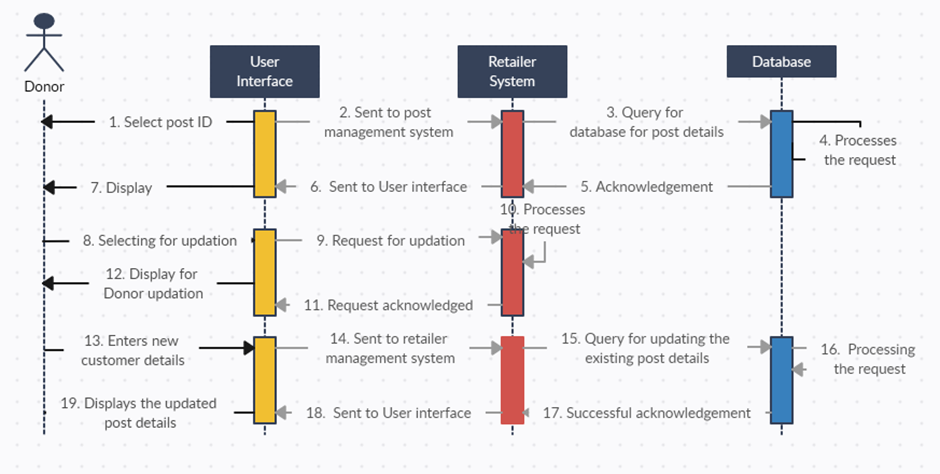
**Figure 7**

1. Create food post: Donors use this process to create food post in which the want to advertise or give out at an auction they can also access all options to creating the food post. Users are allowed to upload post images as explained by the sequence diagram below.



**Figure 8**

1. Update food post: Users have the ability to modify their post for example mark unavailable.



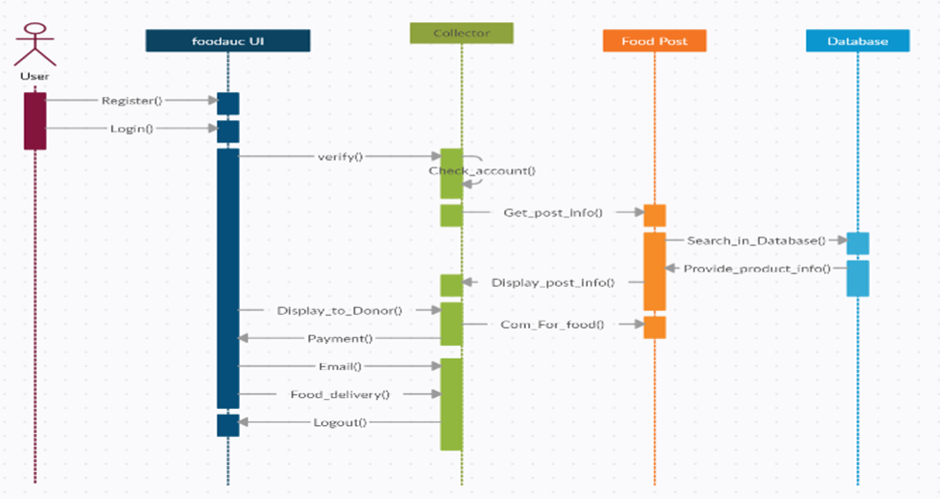
**Figure 9**

1. Messaging: Users have the ability to chat with each other but that must be on the basis of a food post.



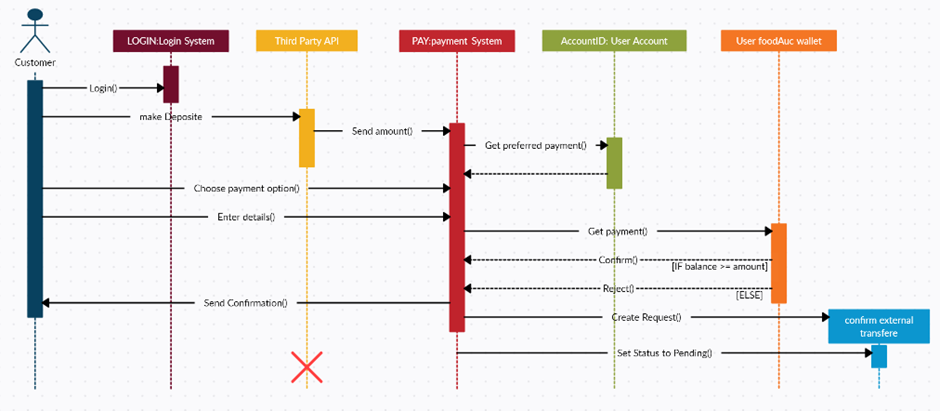
**Figure 10**

1. Collect food: People who come to our application are used by users to collect food from donors. A sequence diagram is designed to provide development steps on how the food collection process happens including the followed up money transferee.



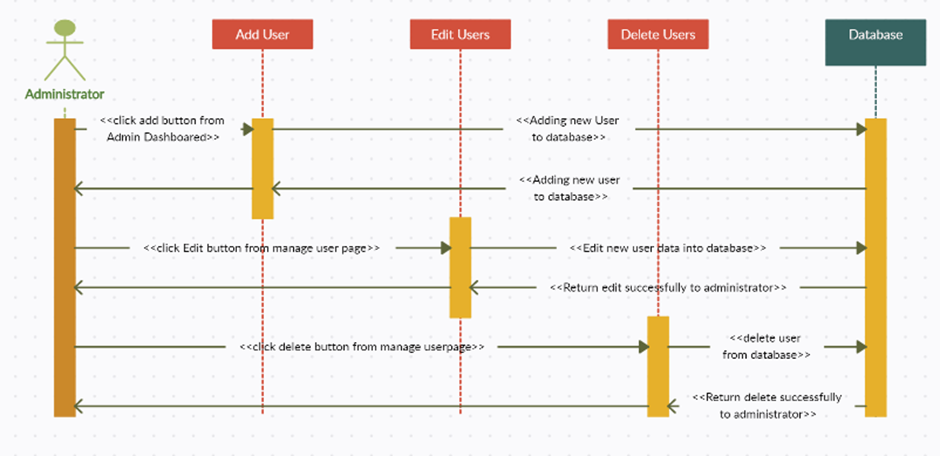
**Figure 11**

1. Deposit: Our system comes with a wallet which allows deposit, transfer and withdrawal. The process to deposit money into your wallet account. Transfer money from one account to another. The process of development of these use cases is displayed bellow.



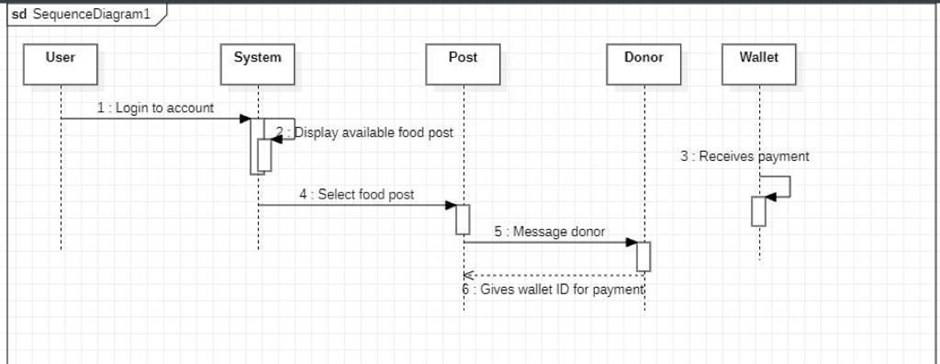
**Figure 12**

1. Manage user: The admin does monitor the system from users, post, and transactions, it in this capacity that the admin works as a secondary actor to the system. The sequence diagram bellow shows how the admin manages user of the system by performing add, delete edit actions on the user I information.



**Figure 13**

1. Recommend user: In our system collectors can recommend donors as a form of encouragement.



**Figure 14**

1. UI DESIGN

In this phase of our project we will be looking at the UI design and implementation. This entails designing the visual layout and interaction of the application. The UI design includes a range of elements such as the color scheme, icons, buttons, menus and other visual components.

1. UI DESIGN CRITERIA

In order to come out with the UI design there are various criteria that need to be taken into consideration.

1. Colors

When deciding on the colors for an application, we need to take into consideration the target audience, trends, user experience and what the application is all about. In this case we are designing an application that deals with food so the colors chosen should be aligned with the main goal of the application.

1. Font style

With regard to the font style used in an application, there are various criteria that need to be considered such as; the font style should be easy to read, has clear distinct letterforms and good contrast between the letters and the background. We need to ensure that the font used is appropriate for the target audience. Also, the font style should establish visual hierarchy and guide the user’s attention to important elements on the screen.

1. Page size

Choosing the right page size for a UI design is important to ensure that the content is presented in a clear and organized manner and that the user can easily navigate the interface. So to determine the page size of an application we need to take into consideration the content of the page, the type of device this application will be used on and ensure that the page size chosen will enhance user experience by making the UI easy to navigate. We also took into consideration the screen size of the trending mobile devices.

1. Device type

Choosing the right device type for a UI is important to ensure that the design is optimized for the device on which it will be viewed. It is important to consider factors such as the technology used, screen size, and the interaction methods. We also need to consider the context in which the application will be used, that is if the application will be used in an office setting or on-the-go.

1. Number of pages

Determining the number of pages in the UI design is important to ensure that the content is organized in a clear and logical manner, and that the user can easily navigate between different sections of the interface. The number of pages should match the amount of content that needs to be displayed, it should be appropriate for the device on which it will be viewed and also enhance the overall user experience.

1. Content

Determining the content to use in UI design is important to ensure that the information presented is clear, relevant and useful to the user. The content should meet the user’s needs, clear and easy to understand, accessible to all users. It should also be organized in a logical and intuitive manner.

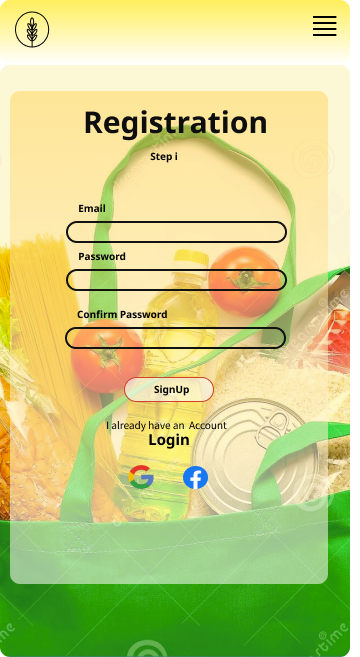
1. Structure

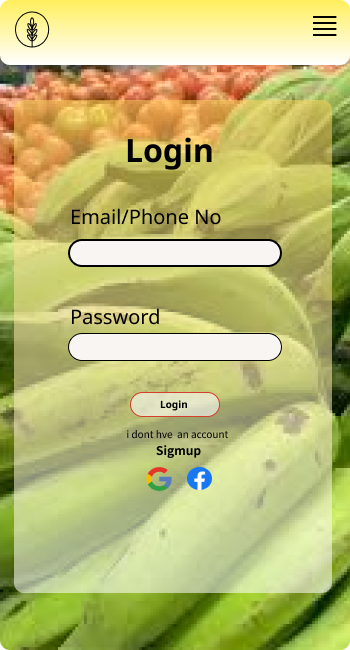
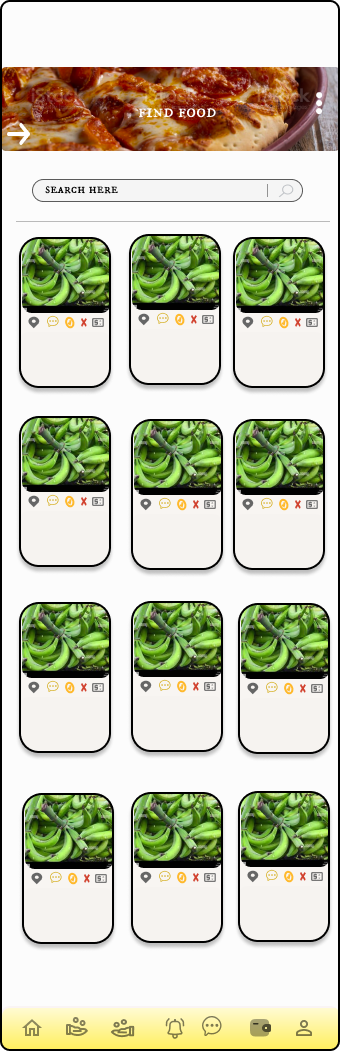
When looking at the structure to use in UI design, it is important to ensure that the information presented is organized in a clear and logical manner. We also need to take into consideration the content, user experience and navigation of the system.

1. SCREENS

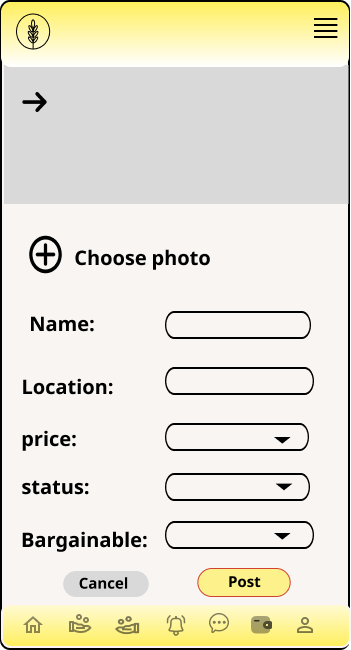
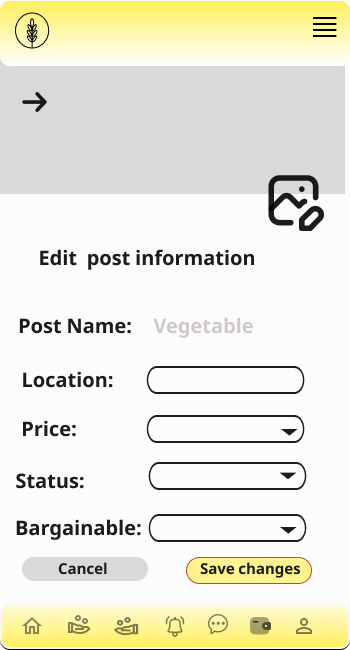
These are the various screens in our system which includes;

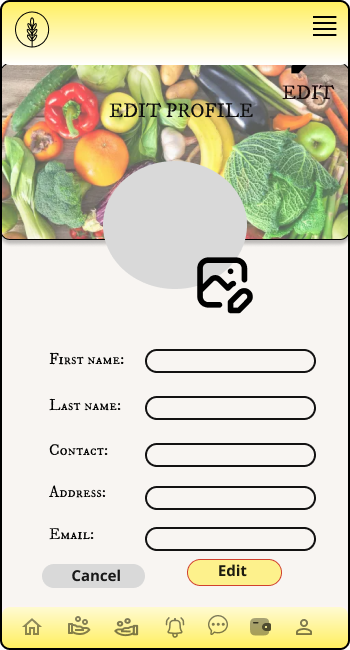


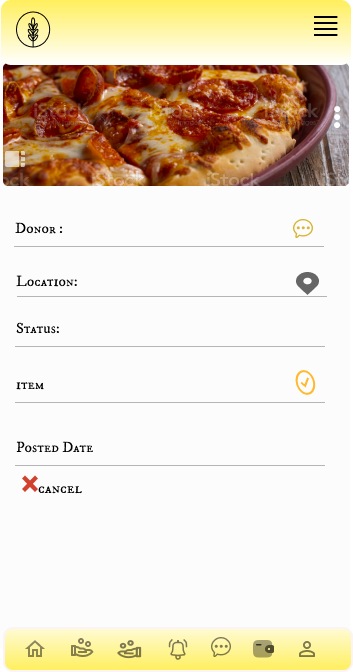
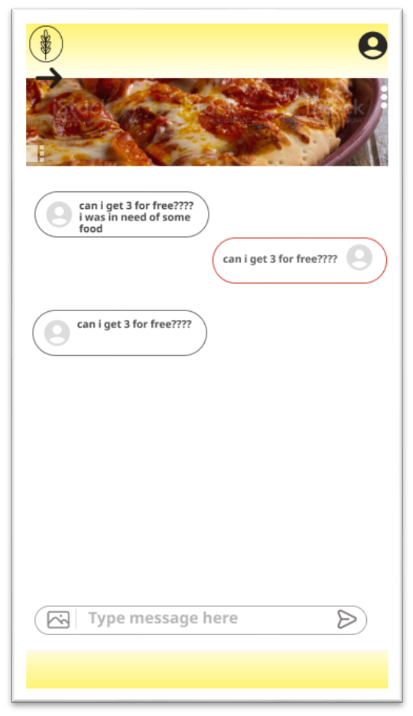
 

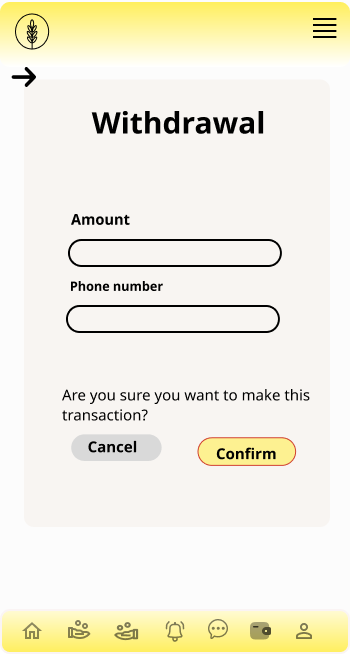
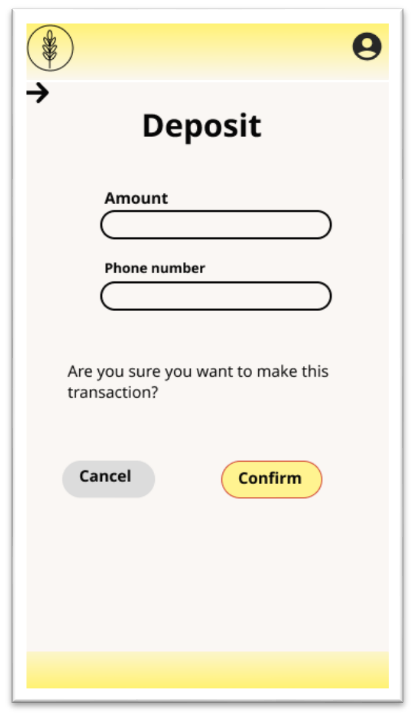
 

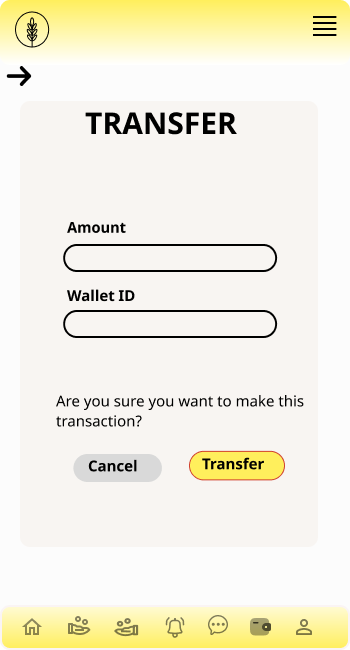
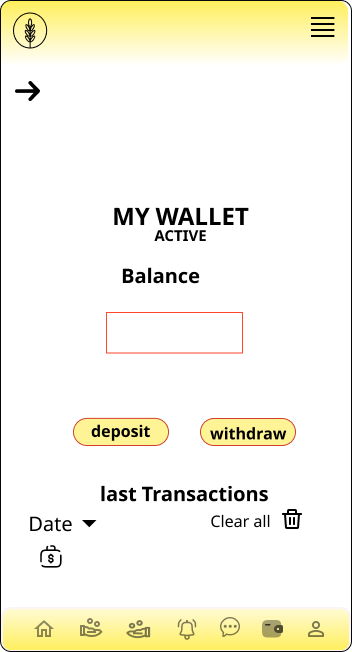
 

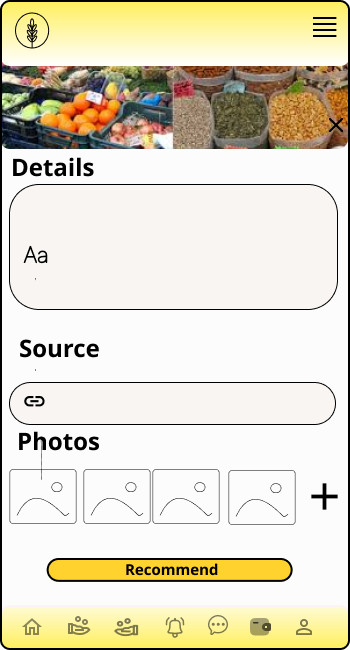
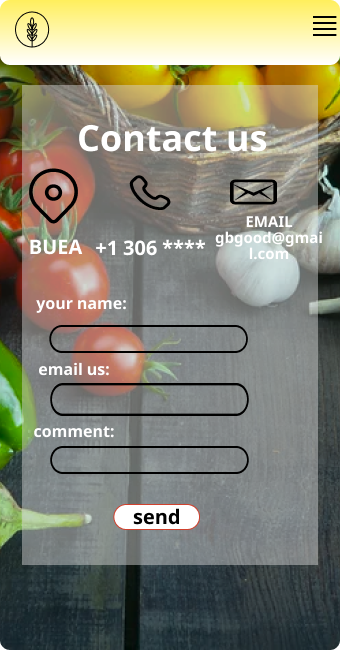
 

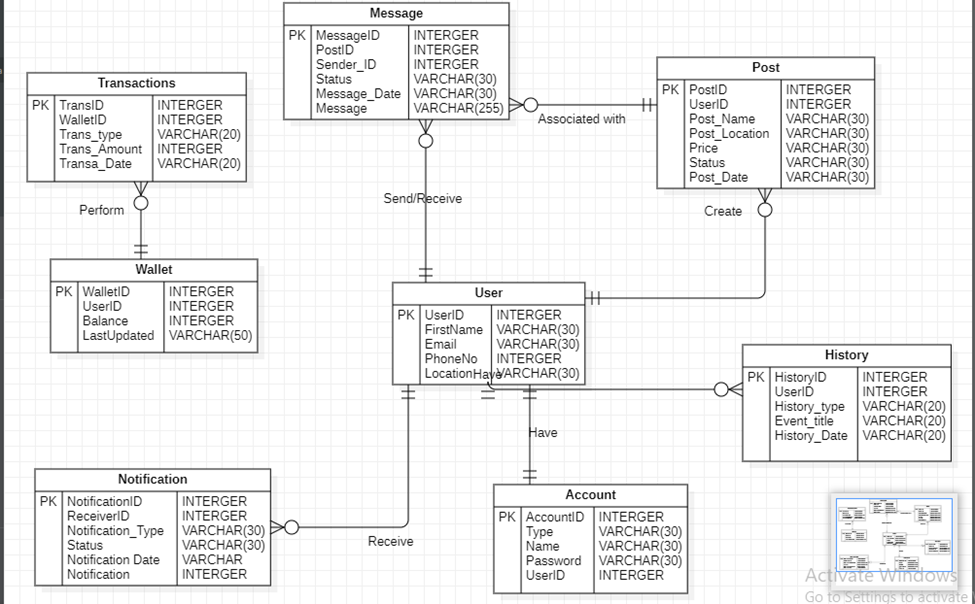
 

1. DATABASE DESIGN

Database design is a collection of steps that help create, implement, and maintain a business's data management systems. The primary purpose of designing a database is to produce physical and logical models of designs for the proposed database system. In order to design a database for our food waste management system (FoodAuc) some relevant information is based under different categories of Database concepts.

1. ENTITY RELATIONSHIP DIAGRAM(ER Diagram)

Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. An entity-relationship diagram, or ER diagram, is essential for modeling the data stored in a database. Below illustrates the ER diagram of our system



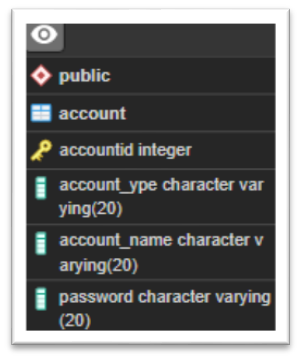
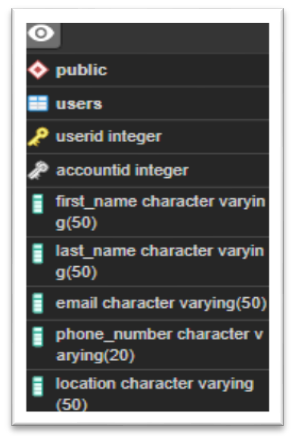
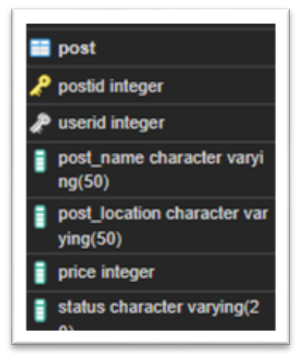
**Figure 15**

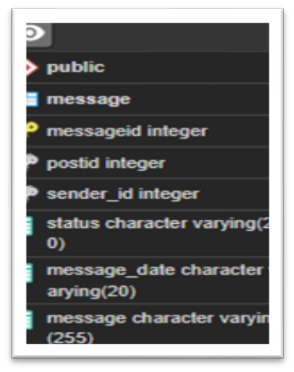
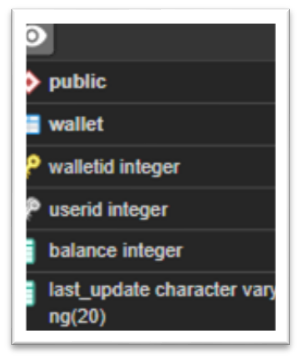
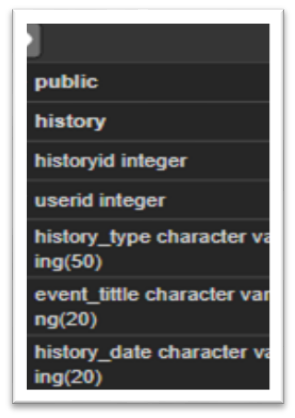
1. RELATIONAL SCHEMA

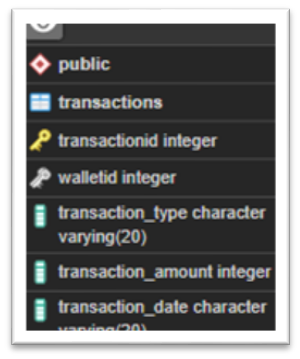
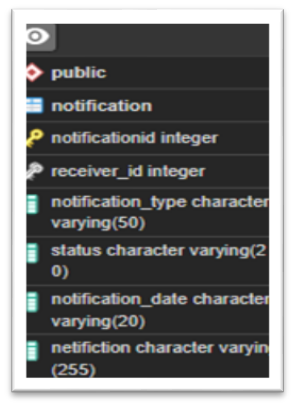
A relational schema is a set of relational tables and associated items that are related to one another. There are 2 levels of the Relational schema.

1. Logical or conceptual level

How users interpret the relation schemas and the meaning of their attributes.

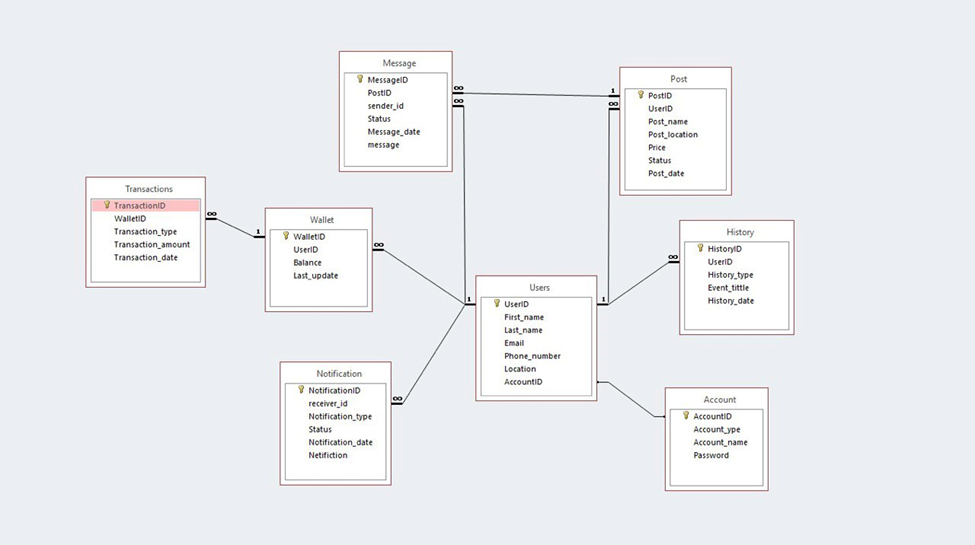
  

1. Physical schema

Physical schema is a term used in data management to describe how data is to be represented and stored (files, indices, et al.) in secondary storage using a particular database management system (DBMS). Below shows the physical schema being used for the FoodAuc application.



**Figure 16**

1. DESCRIPTION
2. Account table

It is the first table of our system. This table holds information about the user login credentials. This table is queried whenever the user requests access to the system, it generates the first id of the user known as the account id. This table gives access only to the registration, login, forgot password page and user table.

1. User table

This is the second table of our system that holds personal information about our user. This table takes the user account id as a foreign key and generates its own primary key. It gives access to the registration, user profile, update profile page, post table, history table, Notification, Message table and Wallet table.

1. Post table

This table holds information about a food post table and is also frequently accessed since information about post is frequently updated. This post stores each post under a unique id. The Create post, Update post, My post, message pages, user table, and message table have access to this table

1. Message table

This table holds chat for particular food post which donor message collectors, it assigns each post message a particular message id. Each message must identify a post, a sender and receiver. The message page, notification page, user table post table have access to this table.

1. Wallet table

The wallet table holds information about the finance, it holds the wallet id required to make transactions on the system either paying for food deposit withdrawal. The transfer, deposit, withdraw pages, transaction table, user table have access to this table.

1. History table

The history table holds past transactions of the user for example last post or last wallet transaction. The history page and user table have access to this page. Each history event is identified by a unique history id.

1. Transaction table

Our system has a lot of transactions going on in it. We keep track of all user transactions from all users. Each transaction is identified by a unique transaction id. Only the wallet table and transaction management page of the administrator.

1. Notification table

Handling notification form food post generated by the system sent to different users.

1. GLOBAL ARCHITECTURE OF THE SOLUTION
2. DESCRIPTION OF THE RESOLUTION PROCESS
3. PARTIAL CONCLUSION

**CHAPTER 4: IMPLEMENTATION (or REALIZATION) AND RESULTS**

1. INTRODUCTION

Implementation refers to the process of putting a plan or idea into action. In the context of software development, implementation typically involves writing code to create a software product or feature.Results, on the other hand, refer to the outcome of a process or activity. In software development, results may be measured in terms of the functionality, performance, or user satisfaction of the software product.The implementation and results of a software project are closely related, as the quality of the implementation can directly affect the results achieved. A well-designed and well-implemented software product is more likely to produce positive results, such as increased efficiency, improved user experience, or higher customer satisfaction.

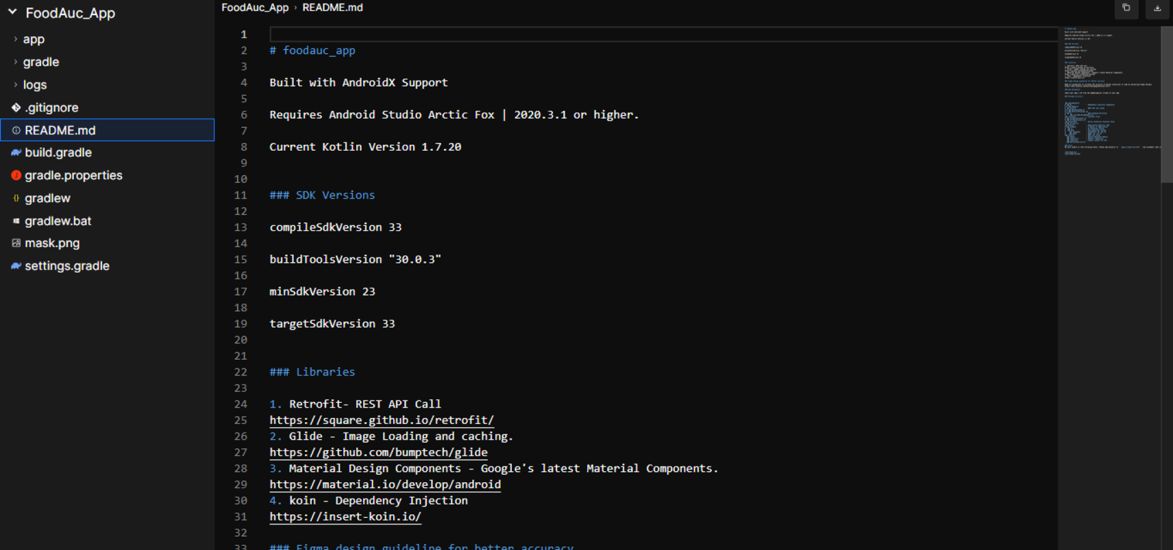
1. TOOLS AND MATERIALS

We made use of front end and back end tools. For the front end we used Kotlin, Java, model support and XML for structuring. The technologies used for back end are SQL and firebase.

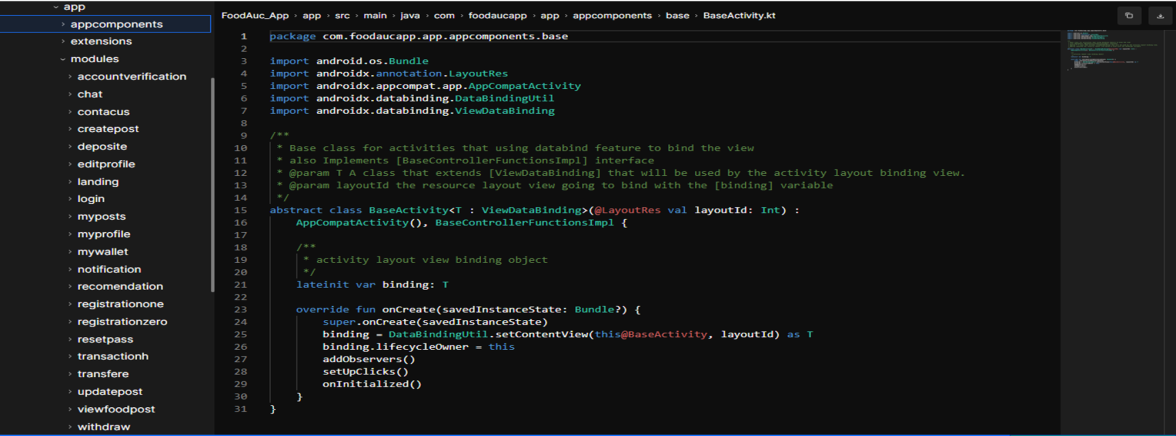
The installation and setup was done using Android studios, canvas and starUml.

1. DESCRIPTION OF THE IMPLEMENTATION PROCESS
2. UI IMPLEMENTATION

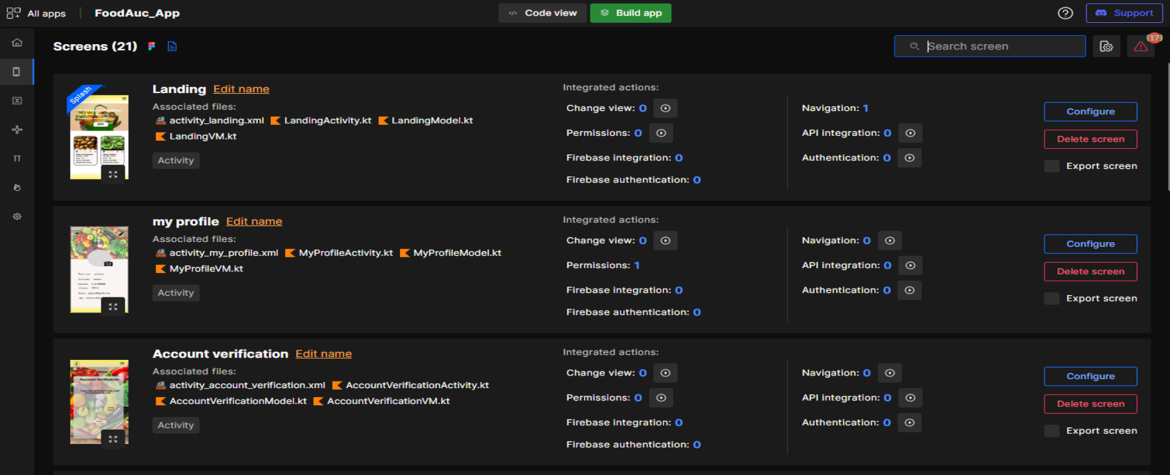
For our implementation we are going to use kotlin class and model support and XML for structuring this is the structure of our application file.



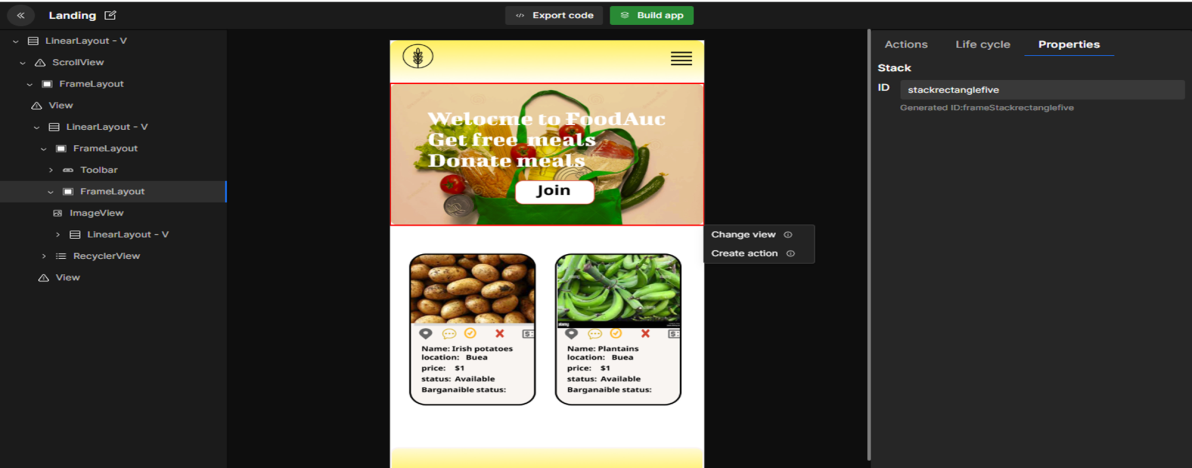
Here are all the models developed, our system comes with 30 pages which we model as different models.



Here are the screens developed and you can preview the screens.

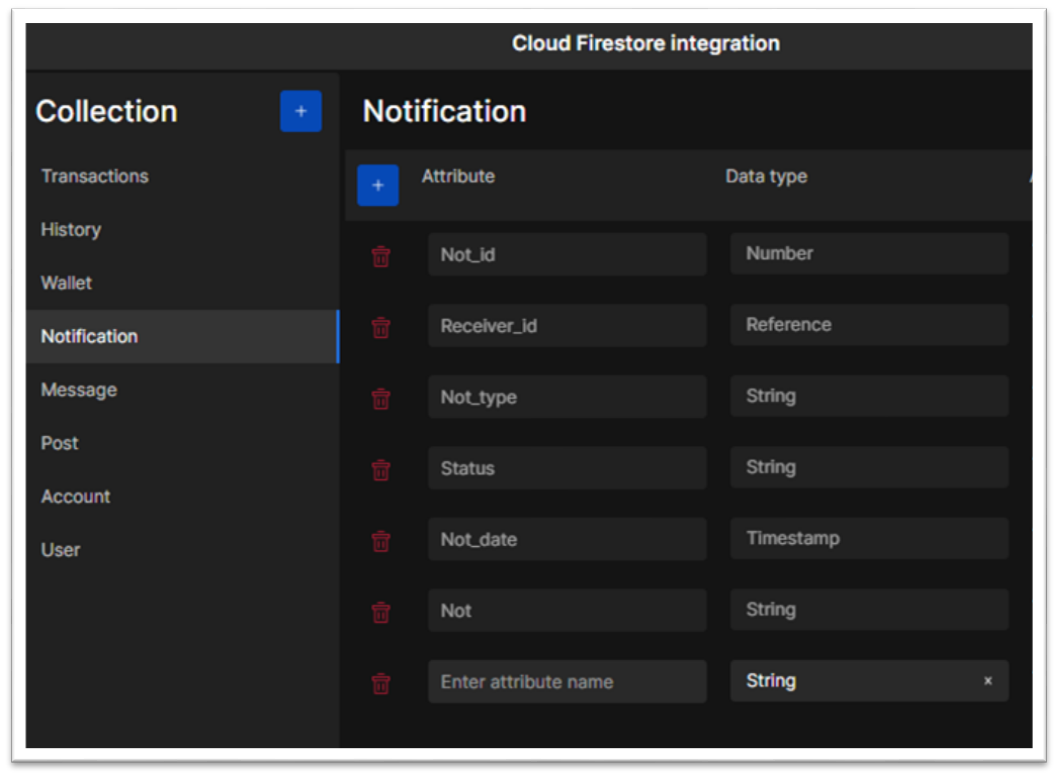


Modify and edit screen actions



1. DATABASE IMPLEMENTATION

Our database is implemented on a cloud server called cloud firestore. The process is an integration process. The tables are created in collections and specify constraints.



1. PRESENTATION AND INTERPRETATION OF RESULTS

There are several benefits of our food waste management system both for individuals and for society as a whole. Here are a few potential benefits:

1. REDUCE FOOD WASTE

The primary benefit of a food application was designed to reduce food waste. It can help reduce the amount of food that is wasted each year. By providing users with tools to manage their food inventory, track expiration dates, and find recipes that use up ingredients they already have, the app can help users make better use of the food they have on hand and reduce the amount of food that goes to waste.

1. COST SAVING

Another benefit of reducing food waste is that it can save users money. By auctioning food at a very cheaper price which can result in lower grocery bills.

1. ENVIRONMENTAL BENEFITS

Reducing food waste can also have environmental benefits. When food is wasted, it not only represents a lost opportunity to feed people, but it also contributes to greenhouse gas emissions from landfills. By reducing food waste, we can help reduce these emissions and minimize our impact on the environment.

1. IMPROVE FOOD SECURITY

Finally, reducing food waste can help improve food security for vulnerable populations. By ensuring that more food is available for consumption, we can help ensure that everyone has access to the food they need to live healthy and productive lives.

Overall, our FoodAuc app was designed to reduce food waste and has the potential to provide numerous benefits, both for individuals and for society as a whole. By helping users make better use of the food they have on hand, these apps can reduce waste, save money, promote healthier eating habits, and improve food security, all while helping to protect the environment.

1. EVALUATION OF THE SOLUTION

During our testing we came out with the following solution.

1. PERFORMANCE EVALUATION

During performance testing, the system was able to handle up to 1000 concurrent users with an average response time of 3 seconds. The system was able to handle a high volume of data and complex transactions without any significant performance degradation. The system's scalability, reliability, and availability were found to be satisfactory.

1. FUNCTIONALITY EVALUATION

All required features and functionalities were implemented correctly, including the ability to create auctions, place bids, and process payments. The system met all the requirements and specifications outlined in the system design document. The system's functionality was found to be satisfactory.

1. USABILITY EVALUATION

Users found the system to be easy to use and intuitive, with a user-friendly interface. However, some users reported that the bidding process was confusing and could be improved. Overall, the usability evaluation found the system to be satisfactory, with some room for improvement.

1. SECURITY EVALUATION

The system was found to be secure, with appropriate measures in place to protect user data and prevent unauthorized access. The system complied with all relevant security and privacy regulations. The security evaluation found the system to be satisfactory.

Overall Evaluation, the food auctioning system was found to be effective and met the needs of its users and stakeholders. The system's performance, functionality, usability, and security were found to be satisfactory, with some room for improvement in the usability of the bidding process. Overall, the system was found to be a successful implementation of the requirements and design specifications, and it has the potential for further development and improvement.

**CHAPTER 5: CONCLUSION AND FURTHER WORKS**

1. SUMMARY AND FINDING

In this report, we have presented an analysis and design of a food auctioning system, and described the implementation and evaluation of the system. The objectives of the system were to provide a platform for buyers and sellers to participate in food auctions and to ensure fair prices for both parties. Based on the findings from the implementation and evaluation, the system has been found to be effective in meeting these objectives.

The system's architecture was designed to be scalable and reliable, with a microservices approach that allowed for efficient communication between components. The system's technology stack was carefully chosen to ensure the system's performance and security, and the implementation details were thoroughly documented and tested. The system's user interface was found to be intuitive and user-friendly, with some room for improvement in the bidding process. The system's security measures were found to be sufficient, and the system complied with all relevant security and privacy regulations.

Based on these findings, we recommend that the system be further developed and improved to enhance its functionality, usability, and performance. Some potential areas for future work include adding more features such as real-time bidding and improving the bidding process to make it more intuitive for users. Additionally, the system's performance could be further optimized to handle larger loads and more complex transactions.

In conclusion, the food auctioning system is an effective solution for the needs of individuals. The system's architecture, technology stack, and user interface have been designed and implemented to meet the system's objectives, and the system's performance, security, and usability have been found to be satisfactory. With further development and improvement, the system has the potential to become an even more valuable tool for the population as a whole.

1. CONTRIBUTION TO ENGINEERING AND TECHNOLOGY

The development and implementation of a food auction system can make significant contributions to engineering and technology in several ways:

1. OPTIMIZATION OF RESOURCE USAGE

A food auction system can optimize the use of resources, such as food and inventory, by providing a platform to redistribute excess resources to areas of need. This can minimize waste and maximize the use of available resources.

1. DATA ANALYTICS

A food auction system can collect and analyze data on food waste and redistribution, which can provide valuable insights to improve the efficiency of food distribution systems. This data can be used to optimize supply chains, reduce transportation costs, and improve inventory management.

1. AUTOMATION

A food auction system can automate the process of food redistribution, reducing the need for manual intervention. This can increase the speed and efficiency of the process, and reduce the risk of errors.

1. SCALABILITY

A food auction system can be designed to be scalable, allowing it to handle large volumes of food and inventory. This can enable the system to be used in a variety of contexts, from small-scale local markets to large-scale national or international distribution networks.

1. SOCIAL IMPACT

A food auction system can have a significant social impact by reducing food waste and improving food security. By providing a platform for the redistribution of excess food, the system can help to ensure that everyone has access to the food they need.

Overall, the development and implementation of a food auction system can make significant contributions to engineering and technology by optimizing resource usage, providing valuable data analytics, automating processes, enabling scalability, and having a positive social impact.

1. RECOMMENDATION

Based on the potential benefits of a food auction system, here are some recommendations for the development and implementation of such a system:

1. CONDUCT A THOROUGH NEEDS ASSESSMENT

Before developing a food auction system, it is important to conduct a thorough needs assessment. This can involve researching existing food redistribution systems, assessing the needs of potential users, and identifying any regulatory or logistical challenges that may need to be addressed.

1. DESIGN A USER FRIENDLY INTERFACE

To ensure that the food auction system is widely adopted, it is important to design a user-friendly interface that is easy to use and accessible to all potential users.

1. BUILD IN DATA ANALYTIC CAPABILITIES

To optimize the efficiency of the food auction system, it is important to build in data analytics capabilities that can collect and analyze data on food waste, inventory, and redistribution. This data can inform decision-making and help to continuously improve the system over time.

1. INTEGRATE WITH EXISTING SYSTEMS

To avoid duplication of effort and maximize the impact of the food auction system, it is important to integrate it with existing food redistribution systems and networks.

1. BUILD IN SCALABILITY AND FLEXIBILITY

To ensure that the food auction system can meet the needs of a wide range of users, it is important to build in scalability and flexibility, allowing the system to be customized to meet the needs of different users and contexts.

1. COLLABORATE WITH STAKEHOLDERS

To ensure the success of the food auction system, it is important to collaborate with stakeholders, including food producers, distributors, and retailers, as well as government agencies and non-profit organizations.

Overall, the development and implementation of a food auction system has the potential to make significant contributions to reducing food waste, improving food security, and optimizing resource usage. By following these recommendations, developers can help to ensure that the food auction system is widely adopted and has a positive impact on communities and the environment.

1. DIFFICULTIES ENCOUNTERED

- We encountered problems in deploying the application on Android studio because Android studio was slow and might exceed runtime before the application can load.

- We had problems translating the front end to code so we used a middleman setup called DhiWise

- Finding difficulties with new tools we used like DhiWise and firestore.

- So many personal problems amongst group members.

-Financial problems like money for data

1. FURTHER WORKS