#### 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;**

**Dr. Valery Nkemini**

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

#### FRIDAY, MARCH 31st 2

**Mobile application programming**

Project description for the project of waste food management. The project here is to manage food, to food management system but something like food auctioning system. Users should be able to give out food free or at reedy-made prices to propel who need it. These prices are known as auction limit.

c. This is maintain the agender of free food of giving out free waist food. But in any giving the is need for some kind of stimulation which referred to as an auctioning.

Our system supports giving out of food as well as taking food free or on auction (paying a compensation). Let us look at the functional and nonfunctional requirement of the system.

Responsible for a native food auctioning system behavior

Here are a few must-haves sets of functional requirements applicable to all for our hybrid mobile application.

1). Registration functionalities:

Registration of our users as dual functionalities, users can create, edit and delete functionalities. The registration process is short and flexibility of 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

2). Third-Party Integrations

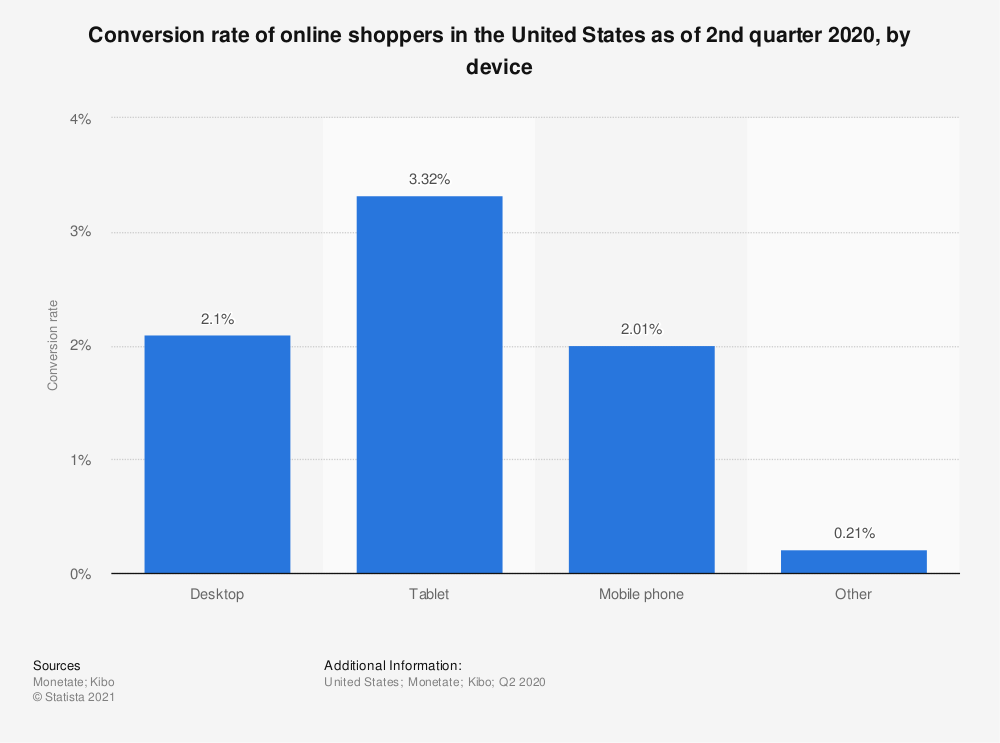
Indicate which third-party software you want to add to your new application. Or maybe you’re replatforming and wish to preserve the systems you’re using now. This requirement concerns both systems streamlining financial operations (like ERP, CRM, PIM) and flexible payment gateways for your users. Specifying 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.

*FR #1 examples*:

* The application shall be integrated with Odoo [ERP system](https://elogic.co/blog/best-magento-erp-solutions-for-small-businesses/) and Svea payment solutions.
* Orange money, or mobile money are added our software as APIs which

3). 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 a 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.



Study your target audience and inquire about their devices. Specify how the position of the essential buttons and options on mobile application should change for a better usage experience (insider tip: place the checkout button within your user’s thumb’s reach, for most people prefer surfing the internet with only one hand from mobile devices+

). Your mobile-first functional requirement should be precise so as not to confuse the developers.

*FR #2 example*: PDP should be adapted to the screens of the Apple iPhone 6s and above.

4). 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.



Product attributes name, bargaining, transportation, quantity and more

*FR #3 examples*:

* We support multiple food donation .
* Product Images on the product detail page should have the option to zoom in.

**Name:** Burger

**Auctioning:** $0.00

**Transportation:** $0.00

**Location**: Buea, long street

**Post owner:** Patu Teke

**Contact:** 687897879

**Bagian:** closed

**Available Quantity:** 34

**Takeout per person:** 2

**Description:**

This I garbage test for the food available here

Takeout

5). Request & Checkout Flow

Your functional requirements for an online auctioning system s should specify how the request 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 (visible both to the customer and the store admin).

*FR #4 examples*:

* The mobile shall show the following order statuses: confirmed, processing, shipped, returned.

6). 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.

*FR #5 examples*:

* Our product info should be shared on Facebook, Instagram, Pinterest, and Linkedin.
* User comments from Facebook should appear on the product detail page in the “Customer Reviews” page.

**We’re following the social sharing practice ourselves at Elogic. If you’re feeling generous (and a little smarter after reading this article), why not share it on social media!**

Non-Functional Requirements: Your Website Flair

As mentioned earlier, non-functional requirements articulate the quality attributes of the mobile application that build positive user experience and optimal mobile performance. The Bible of business analysts — [BABOK](https://www.iiba.org/career-resources/a-business-analysis-professionals-foundation-for-success/babok/?gclid=Cj0KCQjwi7yCBhDJARIsAMWFScOAoU-glykSiFuxN2uF6QdqzgpVAijll-tblm-Epo4vMecOcXrc5k4aAlKFEALw_wcB) — distinguishes between NFRs for merchants (e.g. maintainability, scalability, reusability) and for users (e.g. usability, security, accessibility). In our opinion, they are all equally important at different stages of your business journey: as your application scales, your non-functional requirements may add up.

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 appication 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. Website usability is also defined by

* how easily a user can achieve their goal in a single page visit;
* how quickly they perform the tasks in the app;
* how memorable and intuitive the design is;
* number and time of errors users make.

*NFR #1 examples*:

* A customer should easily find the right item for them, understand what problems it solves, and make a purchase without contacting us.
* No multistep checkout: users must reach the “add to cart” button in one step from PDP.

2). 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 our application 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).

There are many factors at play when it comes to security; specifying this non-functional requirement means taking the first step to [ecommerce fraud prevention](https://elogic.co/blog/ecommerce-fraud-prevention-10-best-practices-for-online-merchants/).

*NFR #2 examples*:

* Only the system data administrator can assign roles and change access permissions to the system.
* The website must be resilient to any kind of attacks, including DDoS and XSS attacks.

3). 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 or websites with legacy architecture: they want their e-stores to load fast no matter the number of integrations 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.

*NFR #3 example*: The website’s homepage should load in less than 4 seconds on iOS 10+, Safari on 4G.

Testing performance will help you understand whether you’ve achieved your KPIs set out in non-functional requirements. Prepare for Magento performance testing following our in-depth guide on the matter!

[Performance optimization](https://elogic.co/category/performance-optimization/)

Magento Performance Testing: A Step-by-Step Guide to Supercharging Magento

4). 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. As saddening as it may seem, there’s no escape from the future issues and you can see many cues on [how to maintain an ecommerce website](https://elogic.co/blog/how-to-maintain-an-ecommerce-website-tips-to-keep-your-business-afloat/). But your task is to make the system easy-to-maintain right from its launch.

*NFR #4 example*: Because we are looking to grow, the website shall remove all the back-end complexities for in-house engineers to make changes to the system in the future.

5). 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.

On the server-side, you might want to add localization features in case you plan to enter new markets and [sell products internationally](https://elogic.co/blog/how-to-sell-products-internationally/). Overall, this NFR accounts for painless business expansion and has both hardware and software implications.

*NFR #5 examples:*

* Our main goal for the next two years is internationalization, so the application shall have multiple user-views for each country we’re selling to.
* The application shall expand to support 500k+ SKUs on a single server without a negative impact on the website load speed.

**Reequipment Specification**

**System Reequipment**

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

Food donor could be an individual organization, food donor can also collect food as well as collectors can also 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 at 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.

**User Reequipment**

1. Each and every user has ability to auction and take food on auction
2. Each user has an account
3. Each user can register to the application
4. Users have notifications about available auctioned food
5. Users can make compensation or pay gifts to auctioned food
6. Users can delete account their account from the auction food sides if not in use
7. Users can take many donation food at the same time // due I don’t know If I should not my be users should only be allowed to take one auction in a day.
8. Users can make collections
9. Users can build and update profile
10. Users can exchange messages
11. Users share food donation post

**Donor Feature**

1. Donors can make food donation advertisement on available food
2. Donors delete food donation post
3. Donors can make food donations free or open for minimum auctioning
4. Donor can mark auctioning debatable or not (enable negotiations, disable negotiations).
5. Donor can view the donation history
6. Donors view collector profile

**Admin Features**

1. Admins can delete food donation post
2. Admins can manage user data in database
3. Admins can view all the pending donations along with status.
4. Admins can view all the donations that they have received.
5. Admins can also view all the agents in the application.
6. Admins can delete user account
7. Admins can delete Food request

**Collector Features**

1. Collector will receive notifications from admins to collect food from donor's homes.
2. Collectors can mark their collection upon collection of food from donor's home.
3. Collectors can also view all those food donations which have been collected by them previously.
4. Collectors can recommend Donors.
5. Collectors can make food request

**Technologies used**

* 1. Java
  2. SQL
  3. Firebase

**Installation and Setup**

* 1. Android Studio
  2. Canvas
  3. StarUml

**For running the application:**

1. java must be installed on the system.
2. You should have a Firebase database.
3. You should have a code editor (preferred: Android Studio)

**Installation and Setup**

1. Download the source code in the desired location on your system.
2. Open the code in your code editor.