# A RECOMMENDER SYSTEM FOR A WEB BASED E-COMMERCE PLATFORM USING K-MEANS CLUSTERING


**ODUOR, JOSEPH OTIENO**

**093084**

**NJOMO, JOSHUA MAINA**

**138065**


An Information Systems Project Documentation submitted to the Strathmore Institute of Management and Technology in partial fulfilment of the requirements for the award of the Diploma in Business Information Technology.


STRATHMORE UNIVERSITY

Nairobi, Kenya

**10 January 2022**

## Declaration

We declare that this work has not been previously submitted and approved for the award of a Diploma by this or any other University. To the best of our knowledge and belief, this research documentation contains no material previously published or written by another person except where due reference is made in the document itself.

093084 Joseph Otieno

Signature: J. Oduor ........................................................................................

Date: 10/01/2022 ........................................................................................

138065 Joshua Maina

Signature: ........................................................................................

Date: 10/01/22 ........................................................................................

## Approval

The Information System Project documentation of Joshua Maina Njomo and Joseph Otieno Oduor was reviewed and approved for examination by:

Name: Tiberius Tabulu

Signature: ........................................................................................

Date: ........................................................................................

## Acknowledgement

We would like to express our gratitude to our supervisor and lecturer, Tiberius Tabulu, for providing the guidelines necessary to undertake this project. We are also grateful to the staff and fraternity of Strathmore University for providing a conducive environment for learning. Finally, we give our appreciation to the Software Development community for posting common issues and their solutions on easy to search for and find sites such as stack overflow.

## Table of Contents

## Table of Figures

# List of Tables

## List of Equations

## List of Abbreviations

| | |
|---|---|
| AI: | Artificial Intelligence |
| API: | Application Programming Interface |
| CPU: | Central Processing Unit |
| CSS: | Cascading Style Sheets |
| DB: | Database |
| DFD: | Data Flow Diagram |
| DIMM: | Dual In-line Memory Module |
| ERD: | Entity Relationship Diagram |
| GMM: | Gaussian Mixture Model |
| GPU: | Graphics Processing Unit |
| IDE: | Integrated Development Environment |
| ML: | Machine Learning |
| NVMe: | Non-Volatile Memory express |
| OOAD: | Object Oriented Analysis and Design |
| OOP: | Object Oriented Programming |
| OS: | Operating System |
| PCIe: | Peripheral Component Interconnect express |
| PHP: | Hypertext Pre-processor |
| RAD: | Rapid Application Development |
| RAM: | Random Access Memory |
| SSD: | Solid State Drive |
| WCSS: | Within-Cluster Sum of Squares |
| Wi-Fi: | Wireless Fidelity |

## Abstract

The advent of the internet has led to the rise of online shopping platforms that allow the modern consumer to buy items from the comfort of their home and vendors to connect with customers way beyond their local vicinity. This, however, brought to light the issue of vendors having difficulty connecting with appropriate customers in the wide customer base provided by the internet and bringing forward comparable products likely to be purchased based on consumer purchase patterns.

This project thus made use of machine learning to help recommend related products to potential buyers, using K-Means algorithm, to cluster related items based on purchase history. Using Rapid Application Development, a user-friendly web application was built that reduces user browsing time and does automatic advertisement of vendor goods to relevant customers improving the quality-of-service.

The culmination of the project was a mock web application that emulates an e-commerce platform and uses the K-Means ML algorithm to recommend items to customers based on transaction history.

# Chapter 1: Introduction

## 1.1 Background of the Study

Online shopping, as defined by Master Card Worldwide Insights, is the process of purchasing goods and services from merchants who sell over the internet (Mastercard Worldwide Insights, 2008). Over the recent years, shopping online has gained popularity among consumers. In addition to offering a wide variety of goods to potential consumers, this innovation also gives businesses a huge market potential. These days, social media not only facilitates presentation of its users on the internet, but it also provides a platform for vendors to market their products online. Because of such advantages, more people prefer online shopping to conventional shopping these days.

In the modern world, online shopping has become the norm and is being widely adopted by consumers. From the consumers' perspective, online shopping provides low and transparent prices, comprehensive assortment of goods and services and a much more convenient shopping alternative that has eliminated the need for brick-and-mortar shops with inconveniences such as squeezing through crowds, being stuck in long waiting queues at the counter, and battling for parking spaces. Consumer purchase behavior in online shopping and in traditional shopping is vastly different. Both differ in social, cultural, personal, and psychological aspects (Wang et al., 2008).

Machine learning encompasses computer algorithms which automatically improve upon themselves through experience and the use of data (Mitchell, 1997). This is a new field in computer science that makes inferences and can learn patterns based on historical data or make association decisions between two datasets and decide whether they are similar or not, and to what degree. Through clustering algorithms, which are a subset of unsupervised machine learning algorithms, data observations are partitioned into clusters each belonging to the cluster with the nearest mean. Algorithms, such as K-Means clustering algorithm, the simplest unsupervised learning algorithm, prove useful in characterizing customer segments for a more personalized marketing delivery (Priy, 2021).

Recent advances in the subfield of machine learning, in the field of artificial intelligence, have made it possible to analyze large datasets that are impossible to make inferences from by a team of human beings in a lifetime. Such has proved useful in the field of statistics to provide useful insights when presented with large datasets as well as make informed decisions based on the summary of the trend of data (Advani, 2021).

Emerging tech companies leverage on the machine learning aspect of computers to be successful in online marketing and advertising.

From the purchase history data collected from e-commerce platforms, more personalized advertisements can be pushed to users increasing sales revenue by recommending what is most likely to be bought by a specific customer, instead of going by the traditional approach of showing advertisements to anyone, even if they are most likely to never be interested in the product or service.

## 1.2 Problem statement

Vendors have difficulty connecting with appropriate customers in the wide customer base provided by the internet. Most buyers do not know what they want until they are shown an example or hinted to a certain product by mostly lucky advertising, advertisements that found the right person at the right time.

Therefore, this project modelled a solution based on clustering using the K-Means clustering algorithm to identify related products and suggest these to clients in a bid to improve sales and reduce dead inventory within these platforms. This was based on patterns and trends derived from large datasets of transaction history, making decisions on the right products to recommend to specific customers.

Taking note of recent privacy concerns on data mining, the system only keeps track of items typically bought together and uses this data for recommendation, using neither cookies nor records of a specific user’s behavioral patterns.

## 1.3 Objectives

### 1.3.1 General Aim

To develop a K-Means based model that clusters together related products based on transaction history and implement it in an e-commerce web application.

### 1.3.2 Specific Objectives

i. To review avenues that vendors use to push their products to their customer base.
ii. To investigate clustering machine learning algorithms that can be used to identify related items.
iii. To develop a web-based mockup of an e-commerce platform to illustrate product searches, transactions and wish lists.

iv. To develop a clustering machine learning model using K-Means clustering that recommends related items based on past transactions.
v. To test the model on the e-commerce platform mockup.

## 1.4 Justification

In Kenya, local retailers are not well supported by the citizens. This leads to businesses shutting down early before they can establish a name for themselves. As a result, people end up spending much of their income purchasing goods from foreign countries, not to mention the additional shipping costs from the country your item is coming from.

With the help of this project, challenges faced by local markets such as competition from foreign countries, overpriced costs, and inability to connect with appropriate buyers were tackled. Using technology, communication between the buyer and the seller was made easier. In addition, the use of machine learning made it easier for buyers to find products they may be interested in.

This in turn leads to an improvement in the quality of service driving higher sales because of higher levels of customer satisfaction. The transaction data is also fully utilized leading to a meaningful use of big data to generate business value.

## 1.5 Scope

The mockup e-commerce web application provides functionality to search for and view inventory items, as well as related items based on the items added to the cart. Once a purchase is made, the items bought are added into the transaction history database as an additional dataset for the next ML recommendation.

The web application has an administrator's module, where transaction history, product rankings based on sales, and registered users can be viewed. In addition, there is a customer module that allows users to sign up and log in, search for items, add items to their cart, and make a purchase.

The machine learning algorithm makes decisions purely based upon the transaction history of multiple users and does not infer similarities between products in the inventory. Views and browsing history are also not considered by the ML model.

## 1.6 Limitations

The popularity of mobile applications over websites could prove to hinder success of the project in terms of the audience it will attract and the traction it will gain upon deployment. Internet access is a necessary requirement to access the application.

Since this project has not been launched to a massive customer base, transaction data, which forms the training data, was fed by a small group of people which may result in biased recommendations based on the demographic of the users.

# Chapter 2: Literature Review

## 2.1 Introduction

This chapter reviews ways vendors push products to their customers as well as current breakthroughs and research in the field of Machine Learning and the various machine learning algorithms, libraries, or APIs in use over the internet with their strengths and drawbacks.

## 2.2 Recommendation Avenues Used by E-Commerce Platforms

Vendors typically make out relevant products to recommend to customers either through content-based filtering, collaborative filtering, or a hybrid of the two.

### 2.2.1 Content-Based Filtering

Using cookies, customer data is kept track of over multiple site visits. Recommendations are then made based on the user's browsing history using the logic that if a user likes viewing a specific item, he will also like the other similar item (MacDonald, 2021).

### 2.2.2 Collaborative Filtering

Data from users who have purchased related products is considered and the information combined to make decisions about recommendations. This method assumes customers who make similar purchases have similar interests (MacDonald, 2021).

## 2.3 Classification of ML Algorithms

Discussed below are the various classes in which machine learning algorithms may lie.

### 2.3.1 Supervised Machine Learning

Training is done based on known patterns of input and output data enabling the model to predict future outputs (Brownlee, 2016). Each example of the input data used in training typically comes with its corresponding, correctly labelled, output.

### 2.3.2 Unsupervised Machine Learning

The model is made to find hidden patterns in input data. Inferences are drawn from data that does not have labelled responses associated with the input data. One of the most common techniques for unsupervised machine learning is clustering which puts data into distinct groups based on shared characteristics in the data (Russell & Norvig, 2015).

### 2.3.3 Parametric Machine Learning Algorithms

The function is simplified to a known form to simplify the learning process. Data is summarized with a set of parameters of fixed size in this scenario. A form of the function is first selected by the parametric machine learning algorithm, for example the form of the mathematical quadratic equation, given in Equation 2.1, then the training data is used to try to learn the coefficients **a**, **b**, and **c** (Russell & Norvig, 2015).

$$ax^2 + bx + c = 0 \qquad \text{Equation 2.1}$$

The parametric ML algorithm has the disadvantage of being constrained to a specific form, limiting the problem complexity that can be effectively managed.

### 2.3.4 Non-Parametric Machine Learning Algorithms

No strong prior assumptions are made about the form of the mapping function. This provides freedom to learn any new and unique functional form given a specific set of training data.

The flexibility of the algorithm means diverse functional forms can be learnt by the machine, resulting in higher power prediction models (Russell & Norvig, 2015).

Such algorithms, however, suffer from being slow to implement and typically require a large set of training data to produce an optimal prediction model.

### 2.3.5 Lazy Learning ML Algorithms

The training data is simply stored, and the algorithm waits until a test data is input into the system. The speed of such algorithms is typically slow as predictions are made based on current data, not an algorithm developed based on historical data (Sammut & Webb, 2011).

### 2.3.6 Eager Learning ML Algorithms

The training data is manipulated and used in the development of a predicting algorithm as soon as it is received. This gives the eager learning algorithm the advantage of faster speed compared to their lazy learning counterparts since the predicting algorithm is already calculated prior to the input of testing data (Sammut & Webb, 2011).

## 2.4 Clustering Machine Learning Algorithms

Clustering is an unsupervised ML method that involves the clumping together of data points. Examples of such are described below.

### 2.4.1 K-Means Clustering

This is an unsupervised type of clustering algorithm. The number of groups to cluster data into are first selected then, using random center points, the data is classified based on the center point nearest to it (Seif, 2018). Based on the classified points, the group center is recomputed by taking the mean of all vectors in the group. This is repeated until the group centers do not change much on successive iterations (Pedregosa et al., 2011).

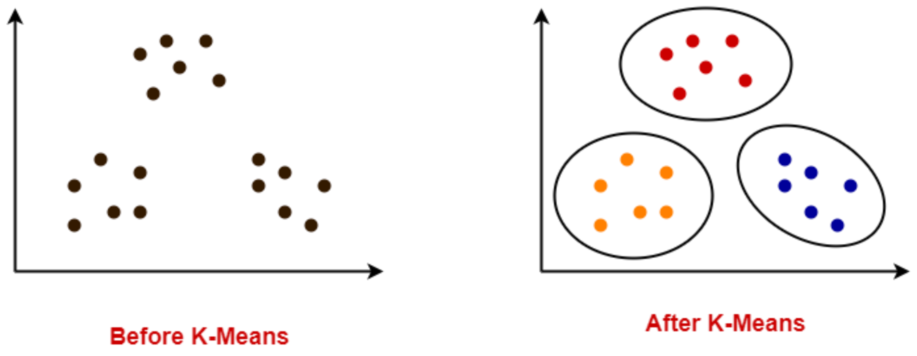



Figure 2.1: K-Means clustering (Singhal, 2020).

Mathematically, the k-means clustering algorithm aims to partition **n** observations into **k** sets, where $\mathbf{k} \leq \mathbf{n}$, to minimize the variance or within-cluster sum of squares (WCSS) as demonstrated in Equation 2.2 where $\boldsymbol{\mu}_i$ is the mean of points in $\mathbf{S}_i$ (Kriegel et al., 2017).

$$\underset{S}{\arg\min} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 = \underset{S}{\arg\min} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i \qquad \text{Equation 2.2}$$

### 2.4.2 Mean-Shift Clustering

Unlike K-Means clustering, the number of groups are not pre-selected (Seif, 2018). Numerous sliding windows with a specific radius move on the data region until a maximum density of data points are within each, that is, shifting to a different position would mean less data points being contained in the sliding window. The data points are then clustered based upon the sliding window in which they reside.

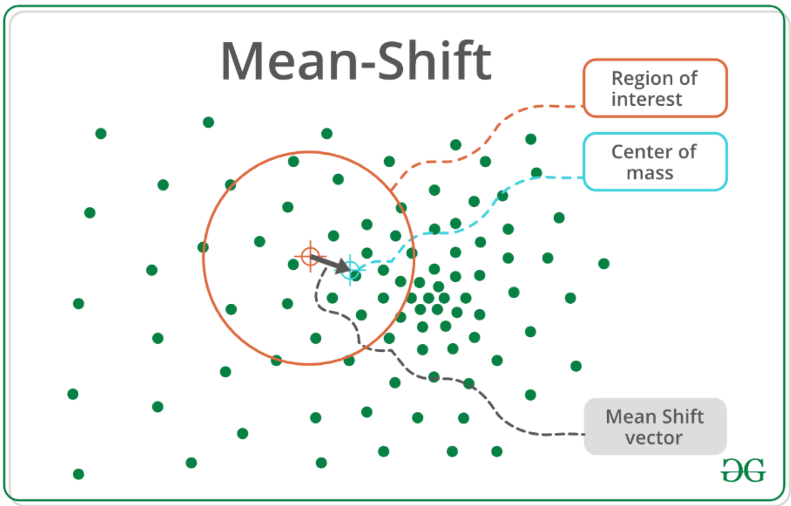



Figure 2.2: Mean-Shift clustering with the sliding window (Tripathi, 2019).

Starting with an initial estimate $\mathbf{x}$, let a kernel function, $\mathbf{K(x_i - x)}$ be given which determines the weight of nearby points for mean re-estimation (Cheng, 1995). Therefore, the weighted mean of the density in the window determined by $\mathbf{K}$ is given by $\mathbf{m(x)}$ in Equation 2.3.

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x)x_i}{\sum_{x_i \in N(x)} K(x_i - x)}$$ Equation 2.3

$\mathbf{N(x)}$ is the neighborhood of $\mathbf{x}$, a set of points for which $\mathbf{K(x_i - x) \neq 0}$.

The difference $\mathbf{m(x) - x}$ is the mean shift in Fukunaga and Hostetler. The algorithm then sets $\mathbf{x \leftarrow m(x)}$ and the estimation repeated until $\mathbf{m(x)}$ converges (Cheng, 1995).

### 2.4.3 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

A clustering algorithm like mean-shift but with the notable advantage of the sliding windows being free to be of any size and any outlier being treated as noise, unlike mean-shift which will throw outliers into a cluster even if the outlier is very different from the cluster it is placed into (Seif, 2018).

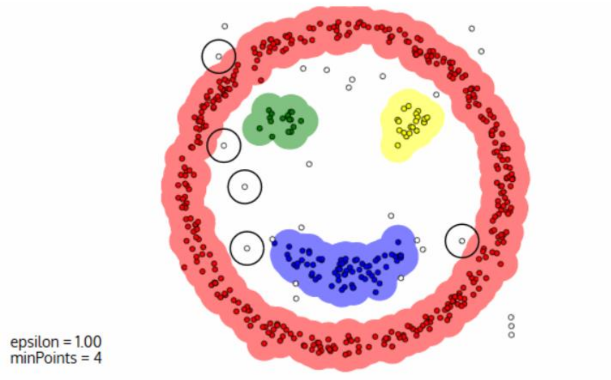



Figure 2.3: DBSCAN with a distance threshold (epsilon) of 1.00 and minPoints of 4 (Seif, 2018).

### 2.4.4 Expectation-Maximization (EM) Clustering Using Gaussian Mixture Models (GMM)

Gaussian Mixture Models (GMMs) assume data points are Gaussian distributed offering more flexibility compared to K-Means clustering (Seif, 2018). Gaussian parameters, such as mean and standard deviation, for each cluster are found using an optimization algorithm called Expectation-Maximization. Therefore, the clusters can take elliptical shapes instead of being limited to circles as in K-Means. In addition, multiple clusters per data point are possible due to the use of probabilities, supporting mixed membership.

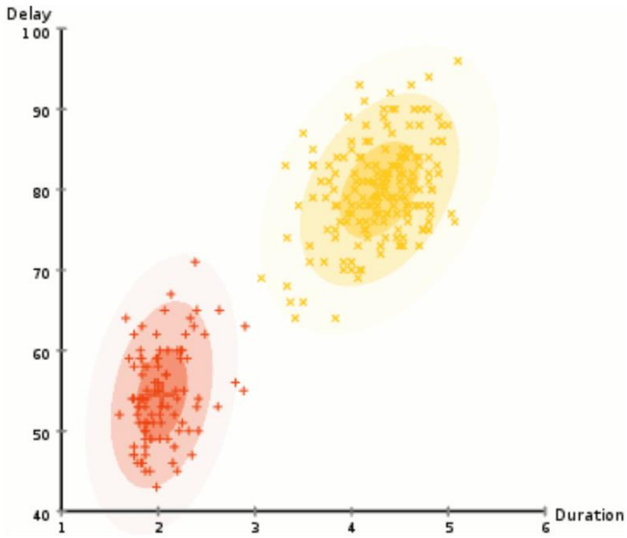



Figure 2.4: EM clustering using GMMs (Seif, 2018).

### 2.4.5 Agglomerative Hierarchical Clustering

Each data point is first treated as a single cluster then on each iteration two clusters are combined into one based on the smallest average linkage distance (Seif, 2018). This is done until only one cluster remains containing all data points. The number of clusters is therefore done by choosing when to stop combining clusters.

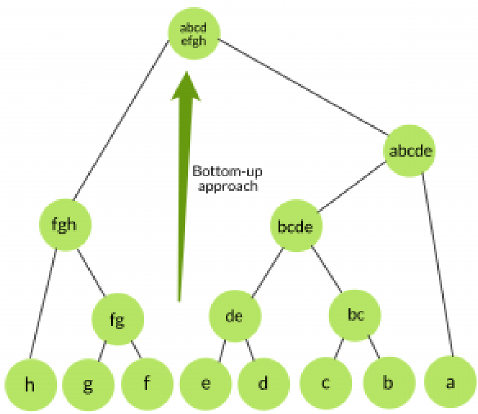



Figure 2.5: Hierarchical agglomerative clustering (Dey, 2021).

### 2.4.6 Justification on the Choice of k-Means Clustering Algorithm

K-Means clustering is simple to understand and implement in code (Priy, 2021). This contrasts with mean shift where the selection of the window size or radius ‘r’ can be non-trivial and DBSCAN which does not perform well on clusters with varying density since the distance threshold varies from cluster when the density varies.

Since EM clustering using GMMs works with probabilities, it is not as ideal as K-Means for our application that needs definite clusters that are easy to extract. Another advantage of K-Means is its fast-running time due to its linear time complexity of O(n) unlike Agglomerative Hierarchical clustering's time complexity of $O(n^3)$ (Seif, 2018).

In conclusion, K-Means is the best option for use as a clustering algorithm in this project.

## 2.5 Related works

### 2.5.1 Tree Identification Using a Distributed K-Mean Clustering Algorithm

The conference proceedings by Kuo-Tai et al (Fan et al., 2010) sought to research models and techniques for efficiently processing large volumes of remote sensor images. To speed up computation, a cluster computing environment was adopted.

The high spatial resolution test image was first partitioned into hundreds of manageable sub-images. The sub-images were then distributed to compute nodes for processing as scheduled by the head node. K-Means clustering algorithm was applied to each node.

In addition to properly identifying tree types of the test site from high spatial resolution images, vast improvement in computation time was obtained. Moreover, the implemented K-Means clustering algorithm performed much faster than stand-alone alternatives.

### 2.5.2 Unsupervised K-Means (U-k-means) Clustering Algorithm

The journal article by Kristina P. Sinaga and Miin-Shen Yang (Sinaga & Yang, 2020) took note of the fact that even though the k-means algorithm is an unsupervised form of machine learning in clustering, the algorithm and its extensions are influenced by initializations of the number of clusters to group data into.

In the paper, an unsupervised learning schema was constructed that could find the optimal number of clusters, free of any initializations by the developers. The merit of entropy-type penalty terms was adopted to construct a competition schema.

The paper eventually concluded with findings of the advantages of U-k-means being freedom from initializations and the use of parameters that are robust to different cluster volumes and shapes making it possible to automatically find the number of clusters, thus demonstrating the superiority of U-k-means clustering algorithm.

### 2.5.3 K-Means Clustering: Determining the Initial Focal Point

The paper by Youguo Li and Haiyan Wu (Li & Wu, 2012) proposed an improved K-Means clustering algorithm by combining the largest minimum distance algorithm and the traditional K-Means algorithm.

The improved K-Means algorithm not only kept the high efficiency of the standard K-Means, but it also raised the speed of convergence effectively by improving the way of selecting the initial cluster focal point, thus being better in both cluster precision and stability.

The paper solved two disadvantages of the traditional algorithm: great dependence to the choice of the initial focal point, and the ease to be trapped in a local minimum.

### 2.5.4 Summary of Related Works

Kuo-Tai et al demonstrated the advantage of speed and reduced computation time of the K-Means clustering algorithm in their case study of identifying tree types from high resolution images (Fan et al., 2010).

The simple K-Means clustering algorithm, however, has the limitations of the user having to specify the number of clusters, assumptions being made that the clusters are spherical, and only being able to manage numerical data.

From the related works, the gaps of the K-Means clusters being spherical and being limited to only numerical data are yet to be tackled.

Table 2.1: Gaps addressed in related works.

| Reference | Gap Addressed |
|---|---|
| (Sinaga & Yang, 2020) | Automatic initialization of the number of clusters. |
| (Li & Wu, 2012) | Selection of the initial cluster focal point. |

## 2.6 Conceptual Framework

To minimize complexity, the system primarily has two classes of users, either an administrator, or a customer.

The system enables the customer to create an account by signing up, and stores their credentials, as well as purchase history, in a local database. The customer can browse the inventory and make purchases. The payment transaction is managed either manually by the administrator, if in form of cash, or automatically by an external PayPal sandbox API, that either confirms or cancels an order. In addition, the system provides a means for the customer to send feedback messages to the administrator should they wish to.

The system gives the administrator access to the list of registered customers as well as orders made by them. The administrator has the obligation to confirm or deny orders, view inventory, and stock up any items that may be running out of stock, and view feedback messages from the customer.

Machine learning, to find patterns based on the purchase history of the customers, is done in the background, with the predictions based on a user's session browsing patterns or purchase history displayed as recommended items the customer may want to have a look at displayed at the bottom of the user interface.

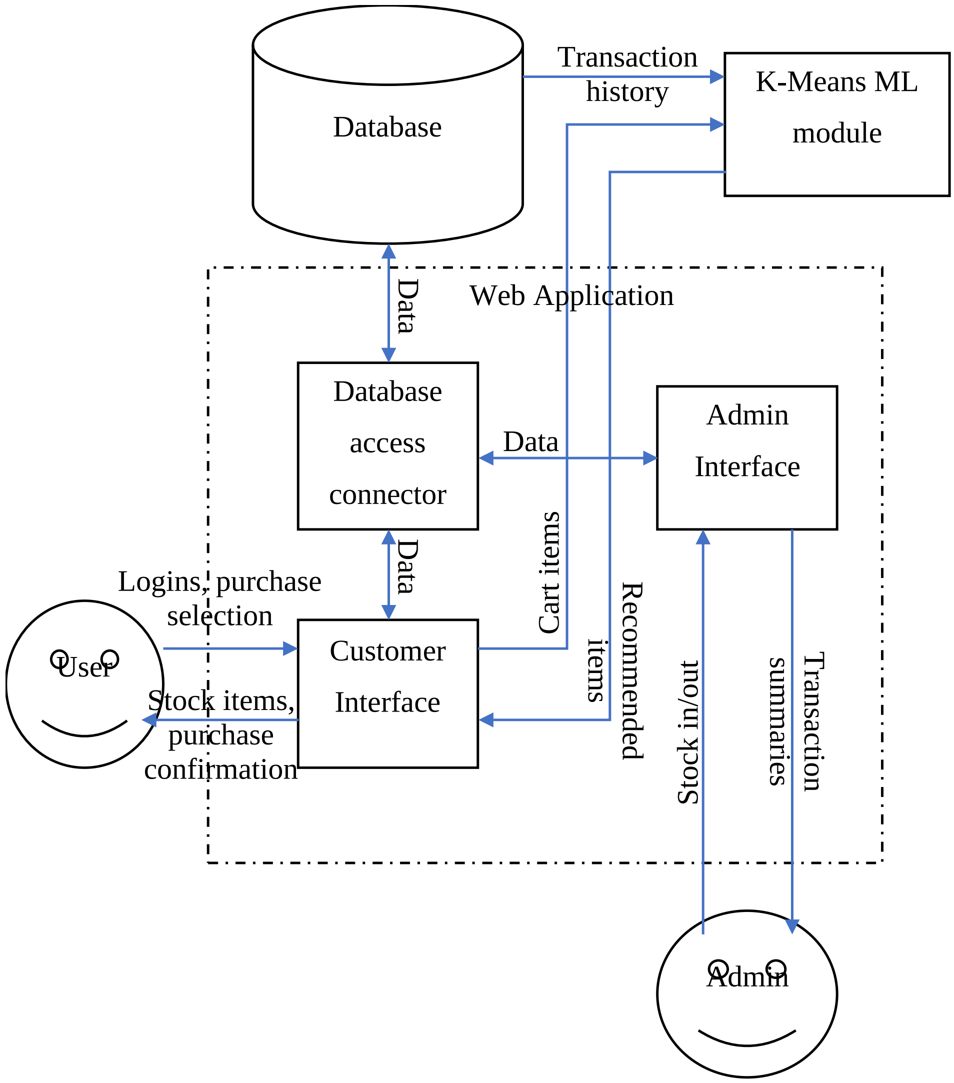



Figure 2.6: Conceptual Framework.

# Chapter 3: Research Methodology

## 3.1 Introduction

This section covers the software development methods that will be used to develop the proposed web application.

## 3.2 System Development Methodology

The nature of this project made it optimally ideal to use an evolutionary approach in the software development process. As the web application was developed by a team of two, and had to accommodate changing requirements, an agile approach was used, more specifically, Rapid Application Development (RAD). This ensured enhanced flexibility and quick adjustments during the development process as the use of machine learning on transactional data was a new concept to the developers who intended to learn implementation details during the development process (Kissflow Inc., 2021).

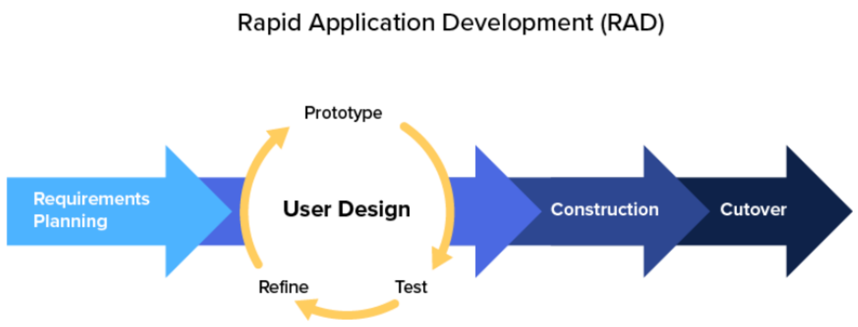



Figure 3.1: RAD process visualization  (Kissflow Inc., 2021).

Object Oriented Analysis and Design (OOAD) was employed to breakdown the entire application into objects and their definitions that were easier to understand (Booch et al., 2007).

With RAD, more emphasis was put on the adaptive process rather than the planning process. The reiterative testing of designs discovered and eliminated potential sources of errors and eventually led to the development of higher quality software with a better machine learning algorithm. User Interface requirements were of utmost importance to make the web application intuitive to navigate through.

### 3.2.1 Requirements Planning

The project studied widely available documentation on the implementation of K-Means clustering algorithm as a machine learning method.

The machine learning module was to satisfy the core functional requirement of being able to relate purchase items based on transaction history and items typically ordered together.

### 3.2.2 User Design

From the collected requirements, use case diagrams, which represent the various classes of users with stick diagrams and the various functionalities of the systems with ovals, linked with arrows representing association were drawn. These visually represent the interactions between the system and its actors.

ERDs were also drawn which visualize the relationship of various database entities (Booch et al., 2007). Flowcharts were drawn to visualize the behavior of specific modules. DFDs which represent the flow of data through processes and sequence diagrams which describe how, and in what order, the system objects work together were drawn as well. OOAD paradigm was utilized in the design.

The design of the interface and the general web application functionality was improved upon in iterative processes by releasing prototypes that were assessed and refined upon with each successive feedback provided by the customers as documented in Table B.1.

### 3.2.3 Construction

This was the last step of the web application development cycle where the final shopping web app was constructed in totality considering all the feedback provided during the user design phase. OOP method was employed. A localhost MySQL database provided by the XAMPP application holds the database information with it being accessible on the constructed web application using PHP language.

This stage mostly perfected on the final prototype that optimally satisfied the user requirement needs.

## 3.3 Tools and Techniques Applied

The core machine learning algorithm is managed by a python file which functions as the ML module for the web application. The python file is connected to the PHP web application files through command prompt commands. The K-Means clustering function, imported from the Scikit-learn python ML library (Pedregosa et al., 2011), takes the n-dimensional location of each inventory item as an array and clusters related items together, after running multiple iterations to find centroids or cluster centers, returning items in a similar cluster to the item selected by a customer.

XAMPP application hosts the entirety of the project, making it executable and testable on a local Apache server. XAMPP has built in tools for creating and updating a local MySQL database and using Hypertext Pre-processor (PHP) programming language to create an application that can connect to the said database.

MySQL database queries were implemented in the PHP source code for intuitive insertion and updating of values in the connected database.

To ensure delivery of the project on schedule, bootstrap CSS and JavaScript files were used and slightly modified to suite our design aesthetic for faster and easier web development (Bootstrap Core Team, 2021).

PayPal sandbox API was utilized as the payment module to simulate payment transactions using virtual currency (PayPal Developer, 2019). A dummy business merchant account was created to receive all the payments along with dummy personal accounts used to simulate customers making payments using virtual currency.

## 3.4 Milestones and Deliverables

### 3.4.1 Proposal

This is a written statement of the system design including explanations of the purpose of the system. This document fully outlines the background behind the project as well as the problem statement that justifies the development of the project.

### 3.4.2 Analysis and Design diagrams document

Use case diagrams, Entity Relationship Diagrams (ERDs), and class activity diagrams will be documented to visualize the functionality of the system.

The use case diagram represents the users of the system, in this case customers and site administrators, using stick figures and the various functionalities of the system represented by ovals. An association arrow is used to represent the relationship between a user and a use case.

ERDs illustrate how various entities are related in the database. Entity attributes, relationship sets, and associations that link entities to relationship sets are captured here.

The class activity diagrams visualize the behavior of modules in the system in form of flowcharts.

### 3.4.3 Working Prototype

This will include an authentication module that allows new users to sign up and current users to login using their registered passwords. The system administrators will have a separate secure login page that cannot be accessed by regular users.

After a successful login, the web application should present the user with the home page where the search and purchase of items can be done. Links to the 'About Us' page and a page to message the system administrators with complaints or concerns are also provided in the home page menu bar. The home page has a link to the in-session user's records in case they need to update their personal details.

The cart module, unique to each specific user, has an interface to show items the user has selected to order as well as their respective icons. Functionality is provided to either remove items from the cart or increase the order quantity as well as a link that redirects the user to a payments module after they confirm the purchase.

The payments module is a PayPal sandbox API that accepts payments in virtual USD using the exchange rate of KES113.15 to USD1.00 (Central Bank of Kenya, 2021). Dummy personal accounts will be used to simulate payments to a dummy business account, each with their own unique credentials. This module either confirms or cancels an order automatically (PayPal Developer, 2019).

### 3.4.4 Test Cases

A test case is a set of actions applied to a system to gauge whether the working prototype satisfies the software requirements and functions as intended. A collection of such test cases is referred to as a test suite. The steps taken during testing, applied test data, preconditions, and post conditions that verify requirements will be documented in the test case document.

### 3.4.5 User Manual

The user guide is intended to assist any of the users of the application. It is highly informative and will explain the use of the application from an abstracted point of view.

An excellent user manual increases the usage of the application. Users may get frustrated and choose not to use the application if they do not understand how to use it.

### 3.4.6 Development Manual

The purpose of the developer documentation is to describe various aspects of how developers should interact with a particular software library or service.

The developer manual describes the content material of the application. Part of this is contained as comments in the source code of the system. This should make it easier for developers who were not involved in the building process of the system understand the source code and easily make modifications if necessary.

# Chapter 4: System Analysis and Design

## 4.1 Introduction

Based on the OOAD paradigm chosen for this project, this chapter breaks down the system requirements in addition to outlining the functionality of the system and the interaction with its users using the analysis and design diagrams discussed and drawn below.

## 4.2 System Requirements

### 4.2.1 Functional Requirements

#### i. Authenticate Customers

The system should enable customers to sign up to the platform by submitting their details and choosing a password. This password is then used for subsequent logins by the customer and the system gives a personalized user experience based on the in-session customer.

#### ii. Modify Customer Details

A logged in customer should have the choice to change their details such as address or phone number if they so wish.

#### iii. Provide Customer Feedback

The system should have functionality for customers to give feedback, either positive or negative, to the developers and administrators of the web application.

#### iv. Make Purchases

The system should have functionality for customers to add items they wish to buy into their cart and checkout by confirming the purchase and making a payment.

#### v. Recommend Relevant Related Items

Customers should be able to view other items they may be interested in buying based on what they have added into their cart by means of a recommender algorithm.

#### vi. Summarize Transaction History

The system should enable the system administrators view a summary of transactions and purchases done by customers as well as product ranking based on purchase frequency.

### 4.2.2 Non-Functional Requirements

#### i. Usability

Through relevant navigation buttons and search bars, the system should enable any customer, whether new or a recurrent customer, interact with the system seamlessly. The use of relevant images on the system in addition to color combinations that make the site readable should make the web application have a more user-friendly appearance.

### ii. Fast Performance

Through query optimization techniques, for example terminating the query connection after it is done running, the responsiveness of the system should be better than if no optimization was done. The recommender algorithm was chosen partly based on its linear time complexity which should minimize the running time to return a recommended product to a customer.

### iii. Security

Hashing of customer and administrator passwords should be done before storing the hashed values in the system database. Using the principle of least privilege, the various actors that interact with the system should be given the minimum levels of data access necessary for them to perform their tasks.

## 4.3 Analysis Diagrams

### 4.3.1 Use case Diagram

The system primarily has two actors, a customer, and an administrator, along with an external recommender module, and an external payment module. Their interactions with the system are as shown in Figure 4.1.

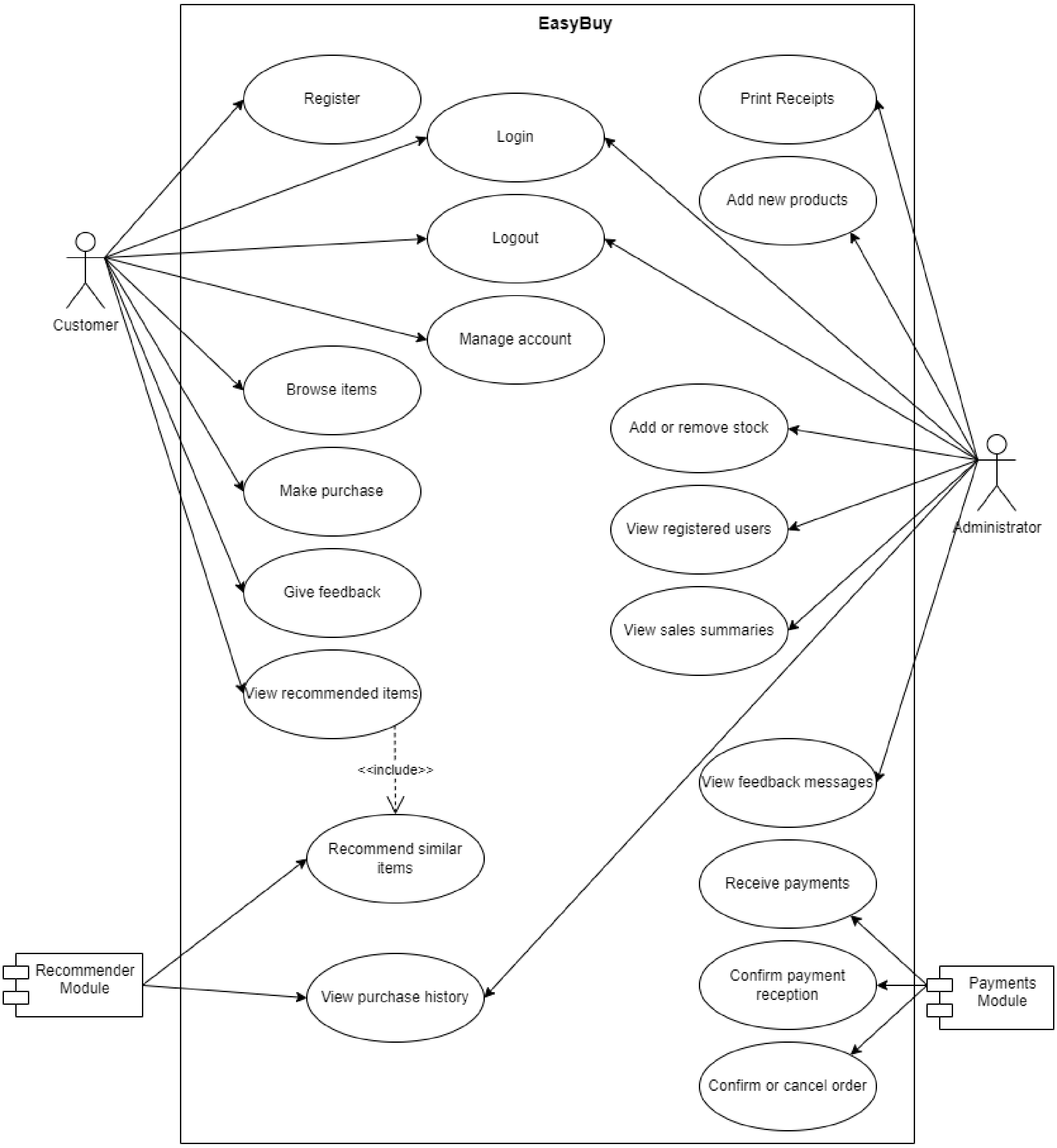



Figure 4.1: Use case diagram.

## 4.3.2 Sequence Diagram

The course of events of the various use cases is as shown in Figure 4.2.

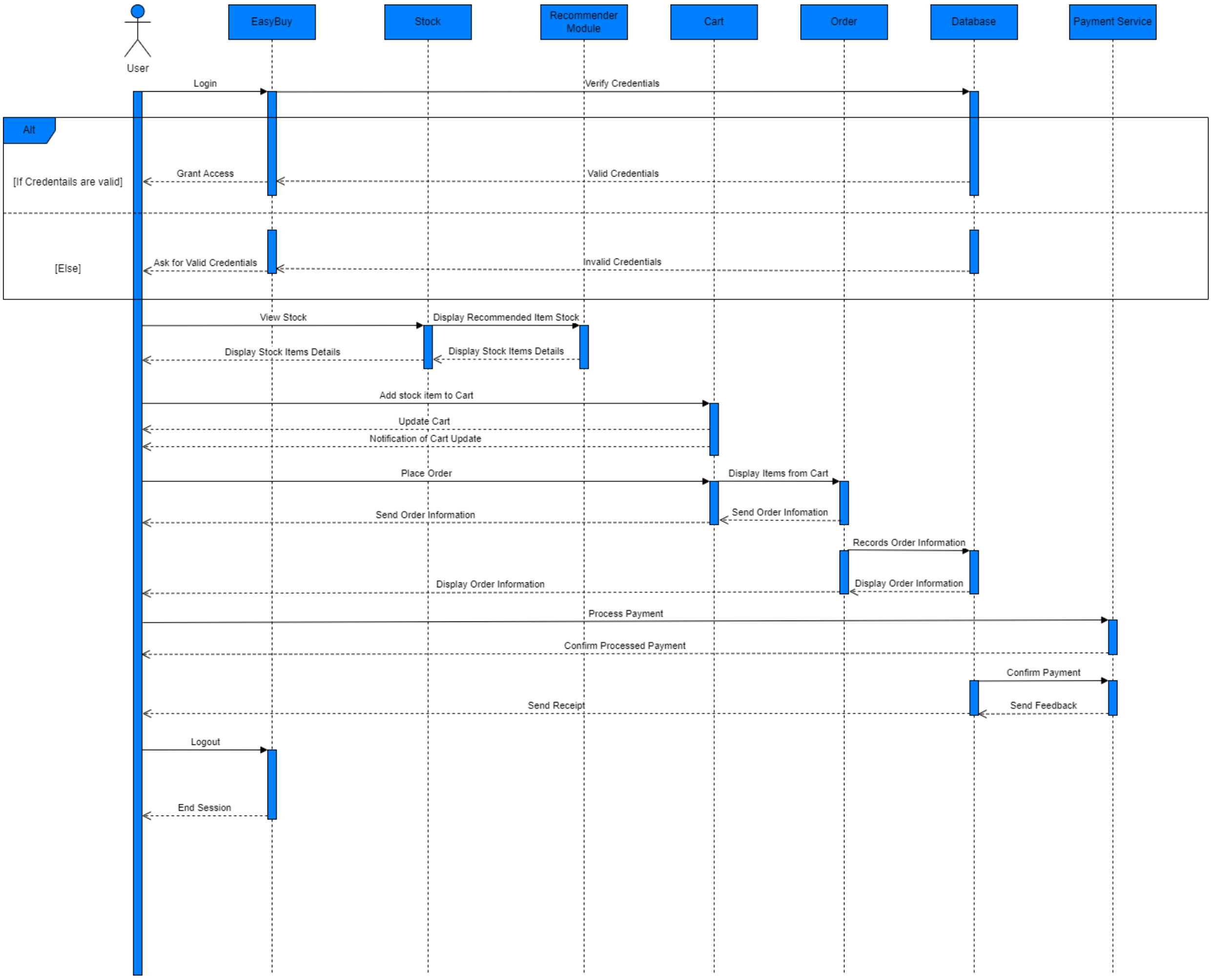



Figure 4.2: Sequence diagram.

### 4.3.3 System Sequence Diagram

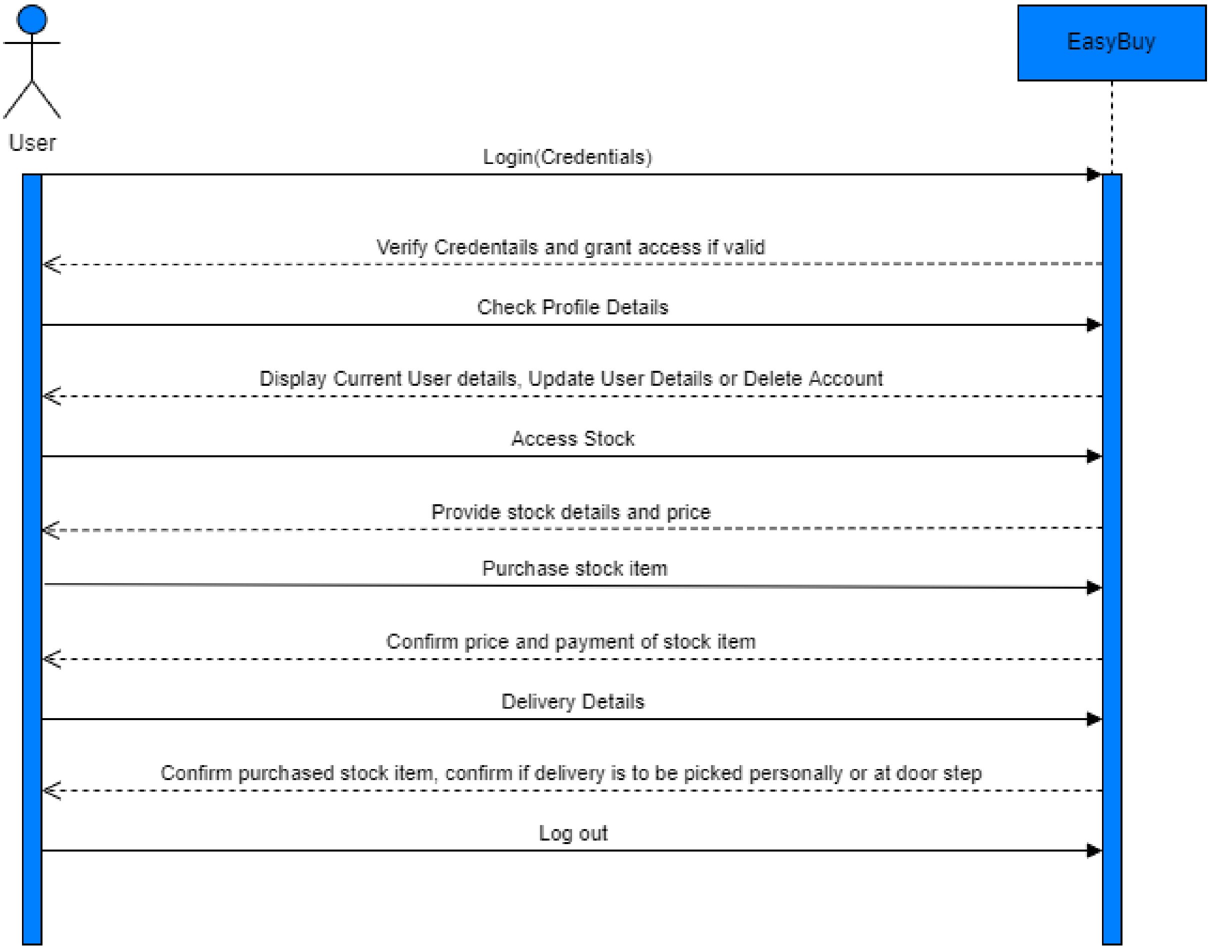



Figure 4.3: System sequence diagram.

### 4.3.4 Entity Relationship Diagram

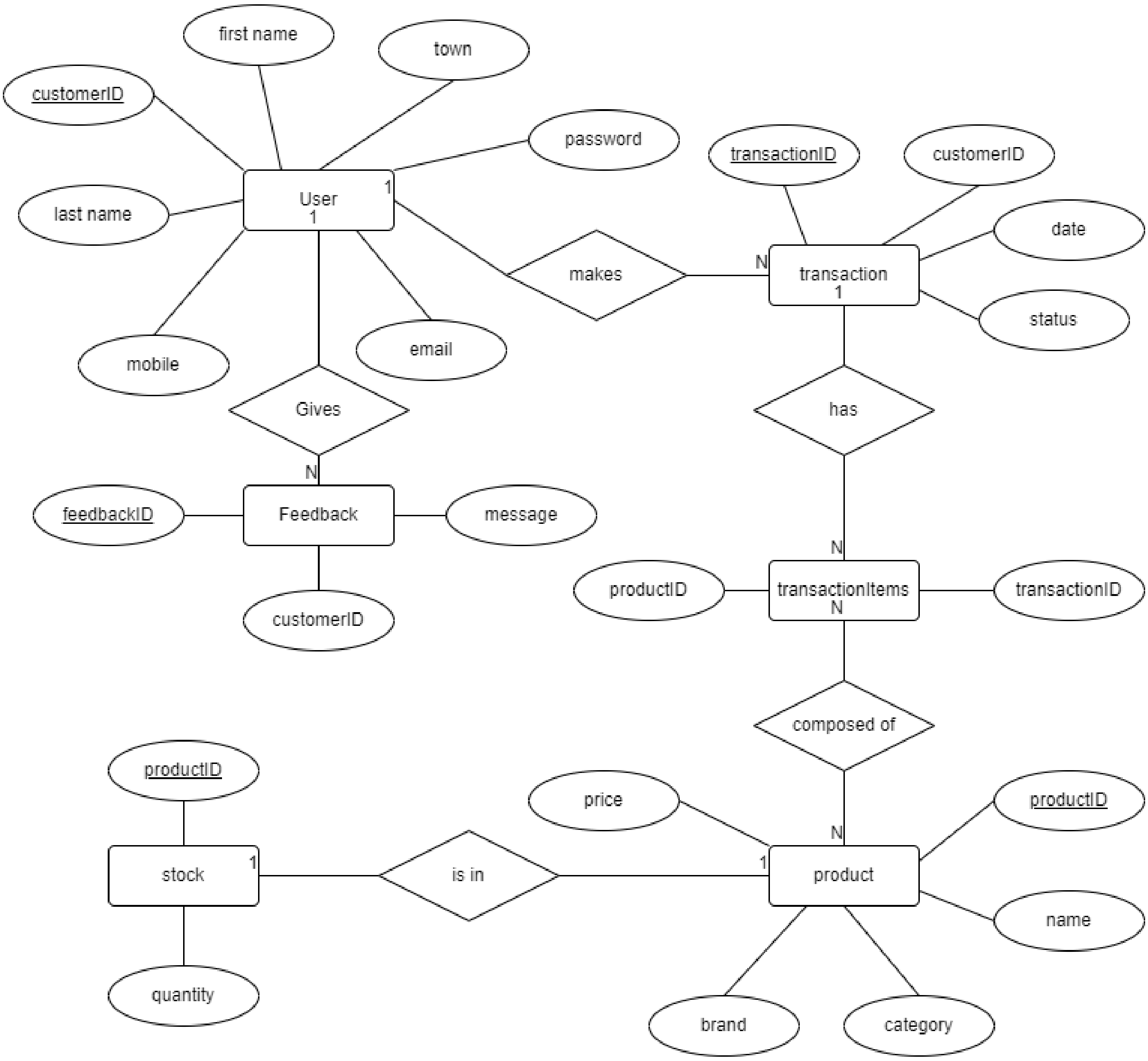



Figure 4.4: Entity relationship diagram.

### 4.3.5 Class Diagram

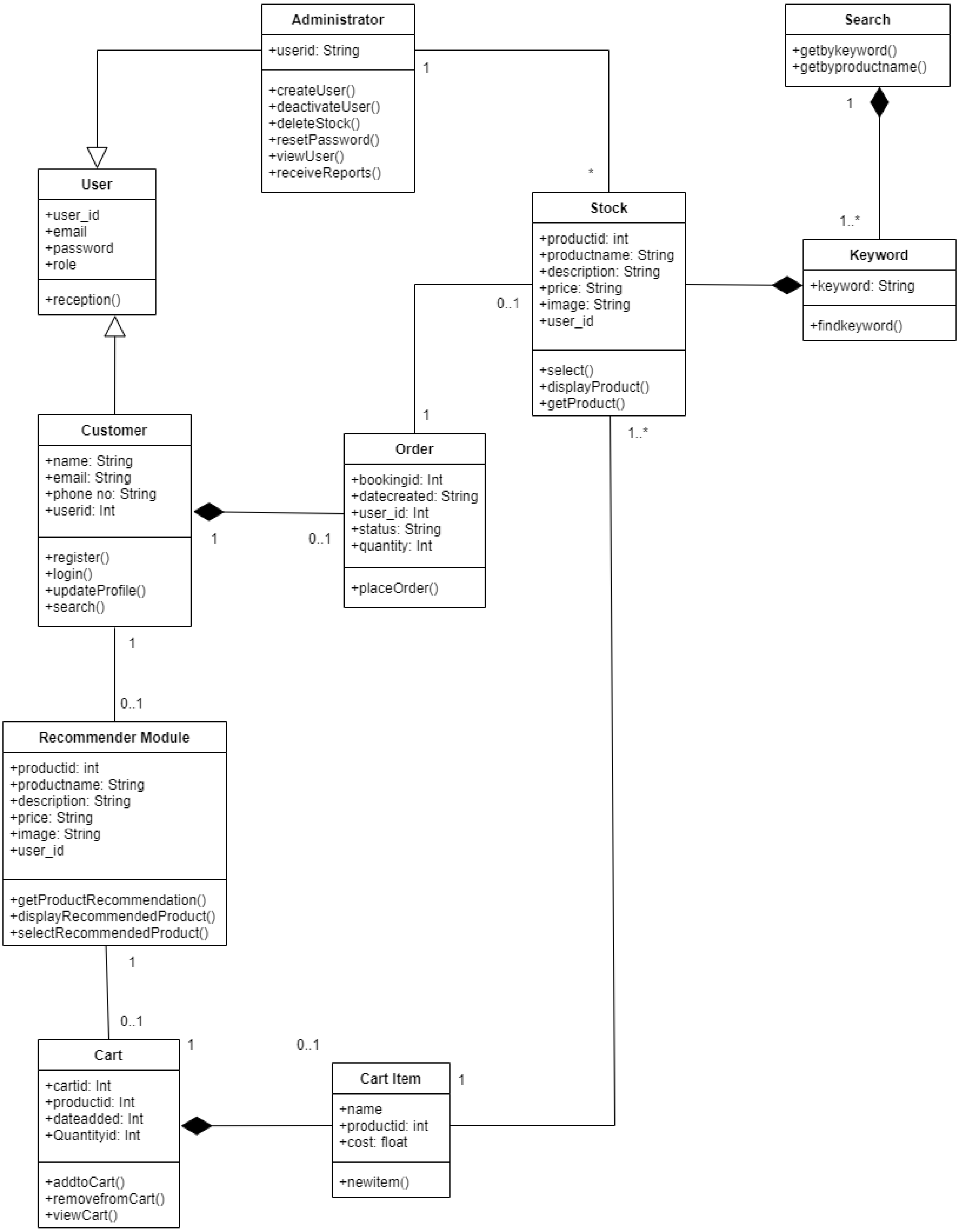



Figure 4.5: Class diagram.

### 4.3.6 Activity Diagram

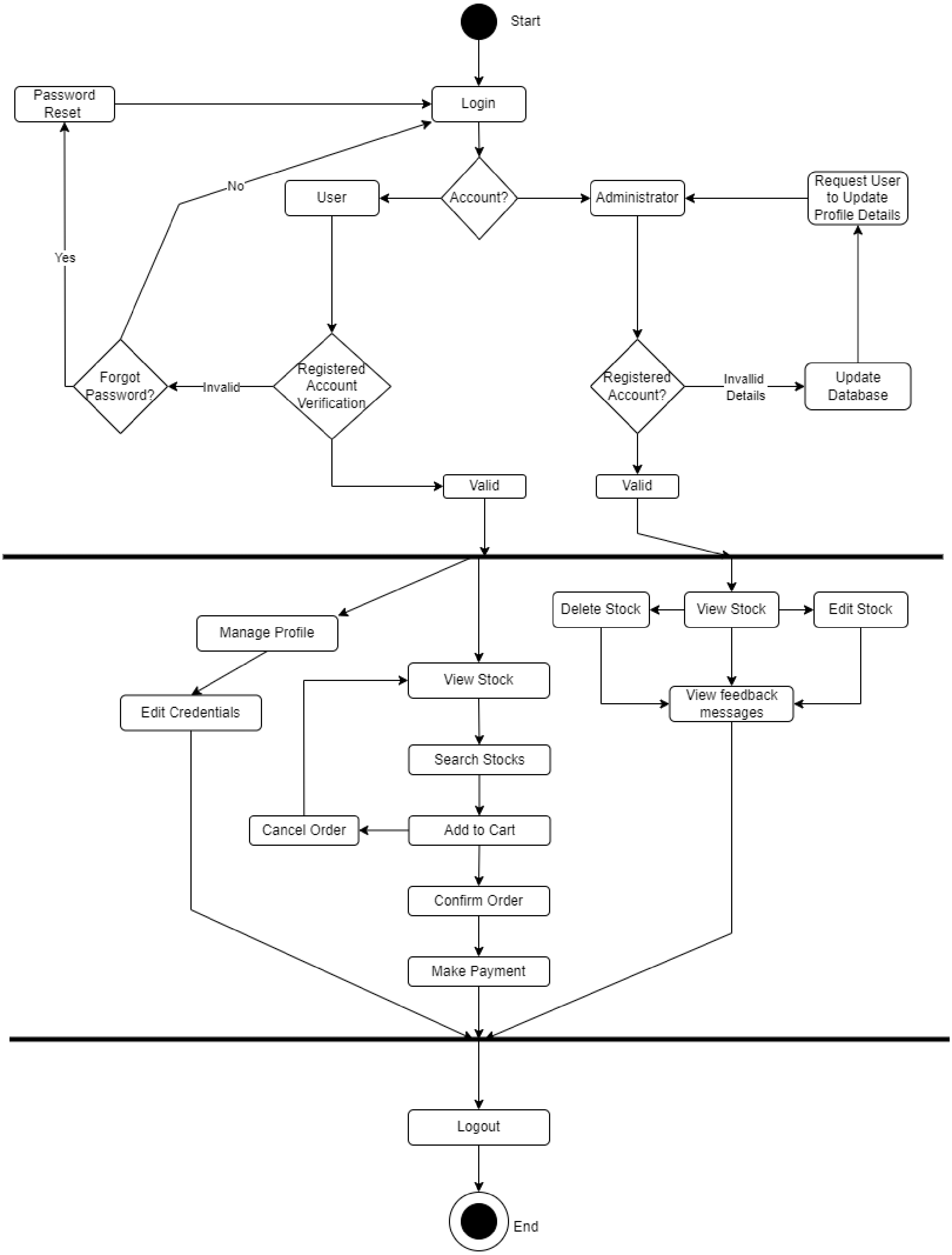



Figure 4.6: Activity diagram.

## 4.4 Design Diagrams

### 4.4.1 Database Schema

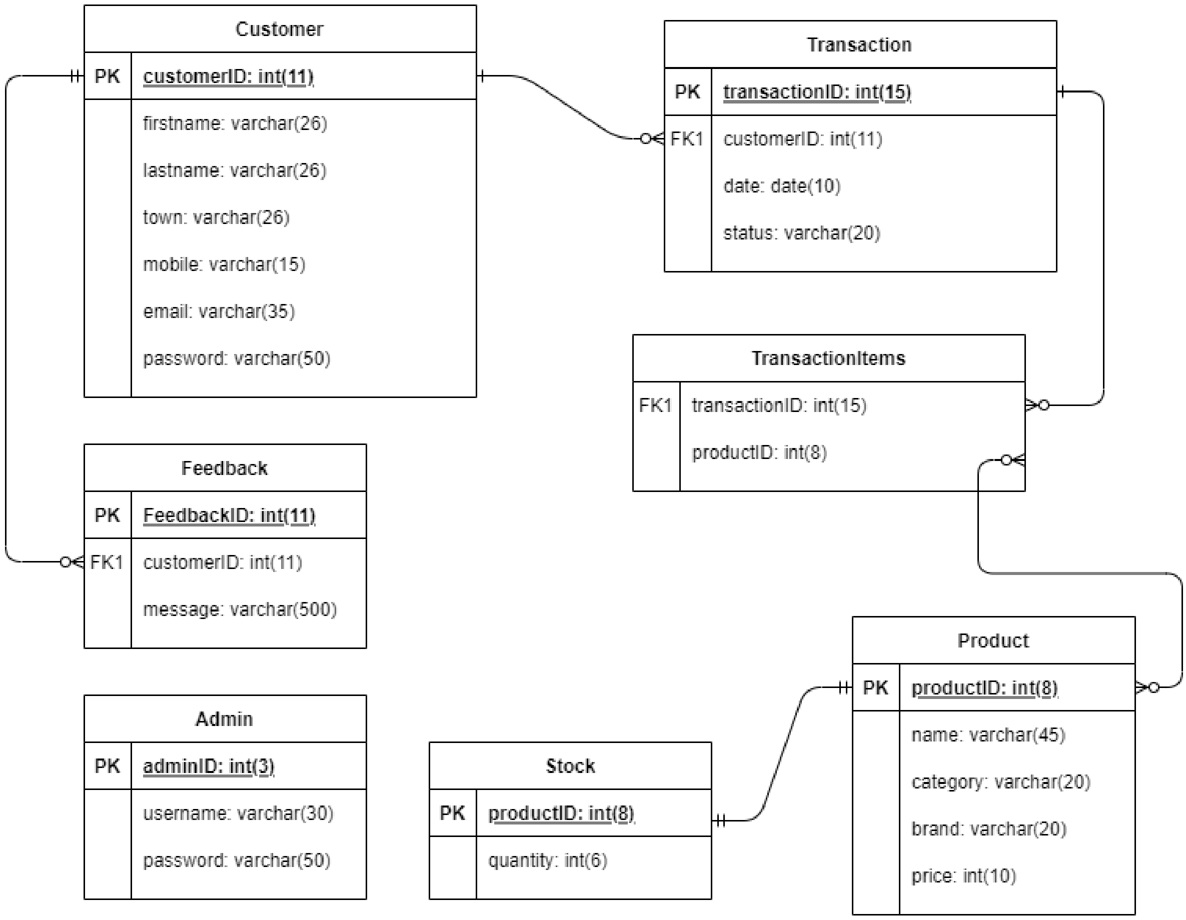



Figure 4.7: Database schema.

### 4.4.2 Wireframes

#### i. Authentication Module

Figure 4.8 shows the layout of the customer sign up page where details are entered in a form with the customer selecting their address from a list.

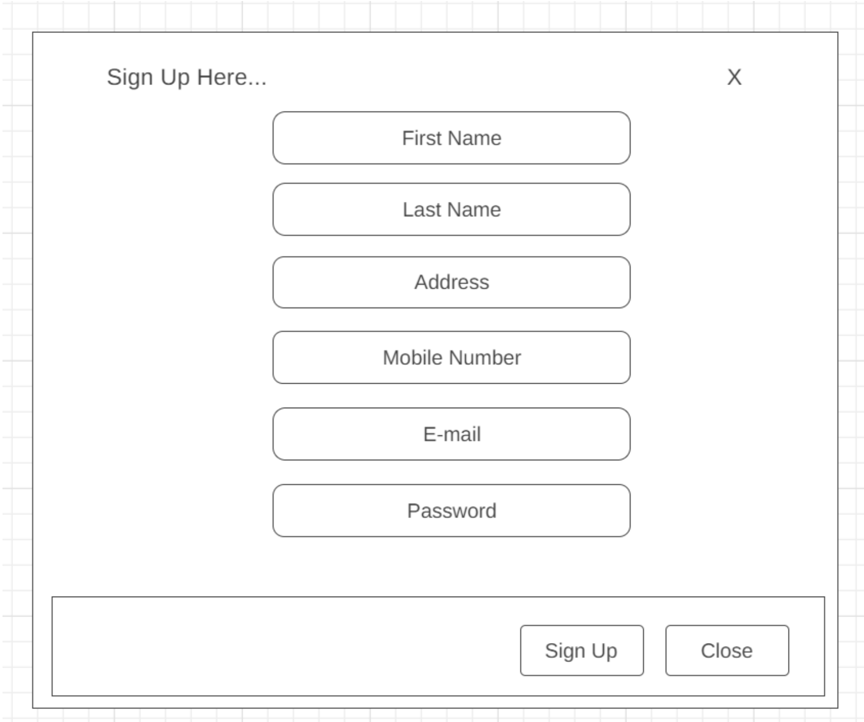



Figure 4.8: Customer sign up page.

Figure 4.9 shows the layout of the customer login page where the customer will be required to enter their credentials in a form that will be checked against stored data in the database.

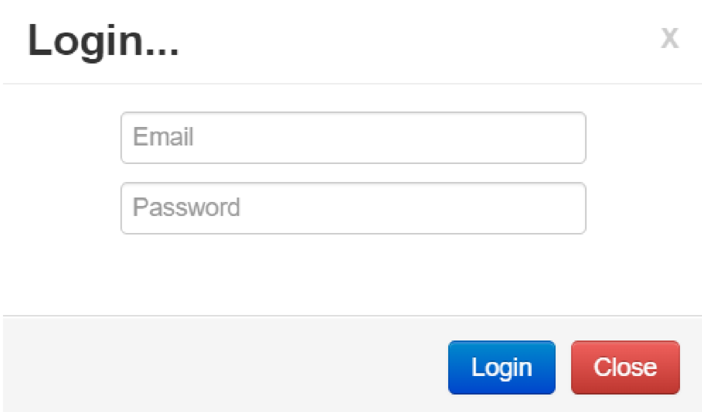



Figure 4.9: Customer login page.

Figure 4.10 shows the layout of the administrator login page. Credentials entered in the form are checked against stored data in the database to allow authentication.

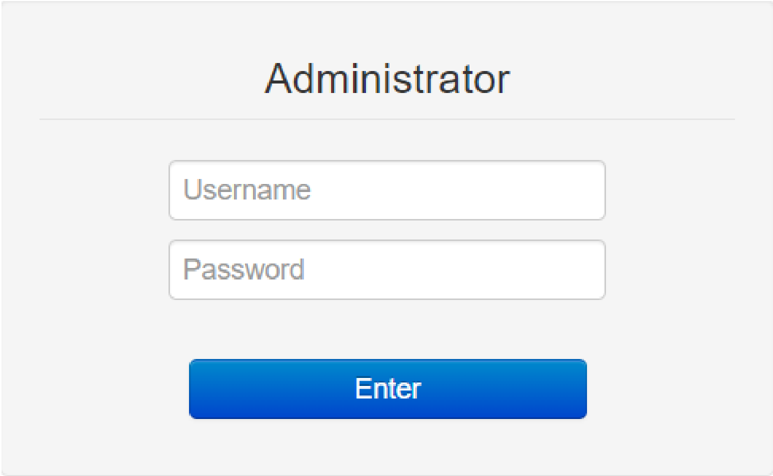



Figure 4.10: Administrator login page.

### ii. User Home Page

Figure 4.11 shows the layout of the customer's home page after a successful login.

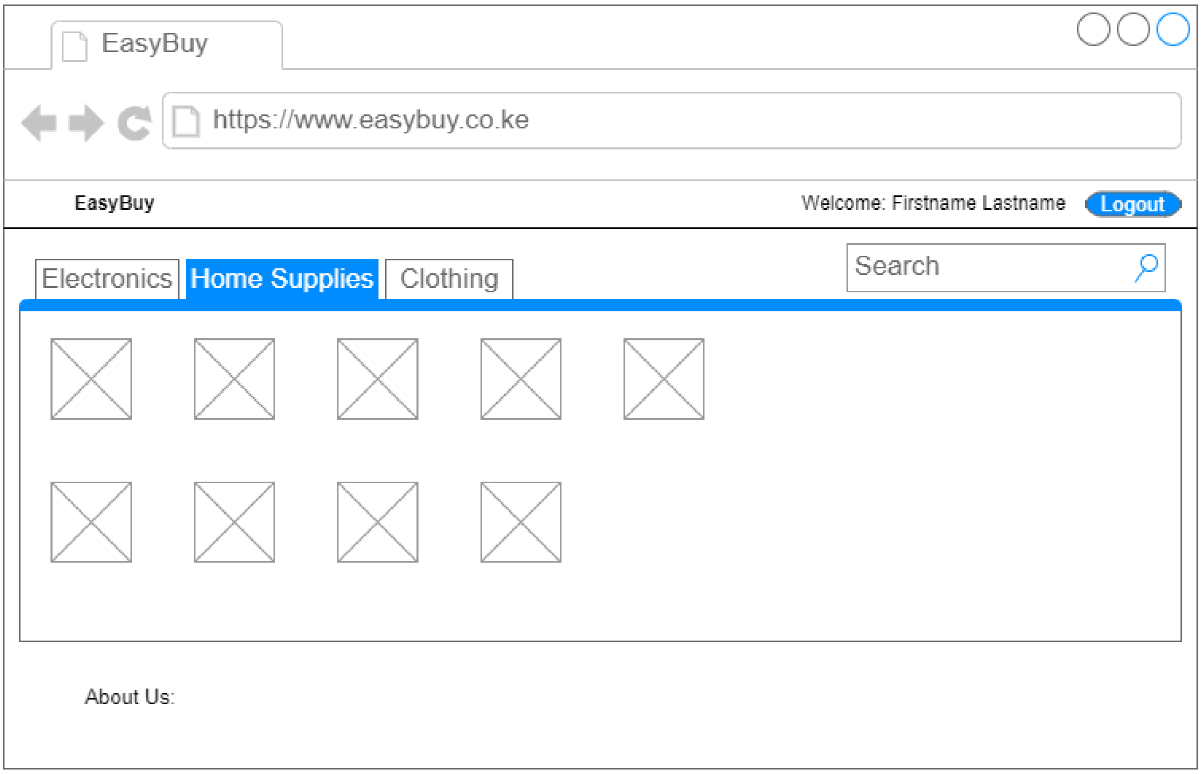



Figure 4.11: Customer home page wireframe.

### iii. Administrator Home Page

Figure 4.12 shows the administrator home page layout after a successful login.

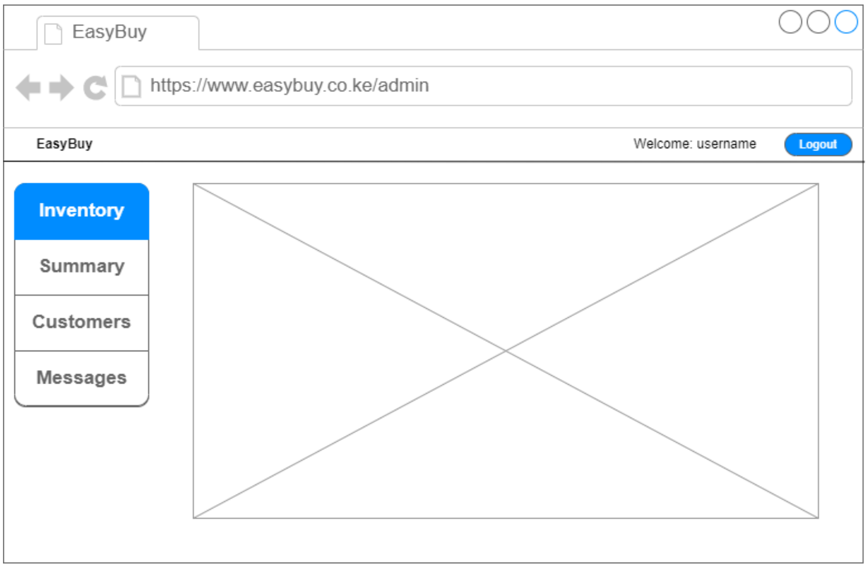



Figure 4.12: Administrator home page wireframe.

### 4.4.3 System Architecture

The customer and administrators have different interfaces for accessing data in the database, with each interface employing the principle of least privilege. The recommender module draws data from both the database and the customer's web browsing session to pass recommendations to the customer through the web application. The payment module fetches order details from the web application which redirects the user to an external payment API upon confirmation of the order. All data is stored in a central database.

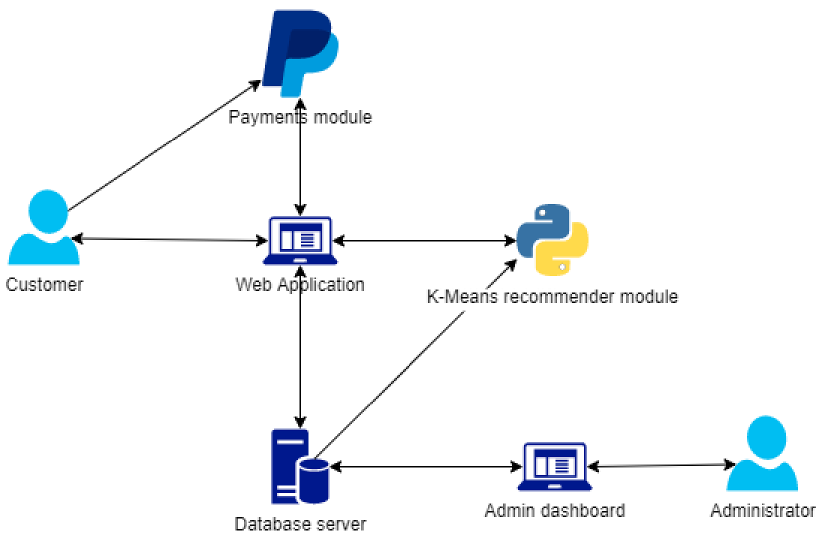



Figure 4.13: System Architecture

## Chapter 5: Implementation and Testing

### 5.1 Introduction

This chapter covers how the actual system was implemented as well as the tests conducted to verify and validate it. Unit testing and integration testing were conducted in this regard as discussed below.

### 5.2 Description of the Implementation Environment

#### 5.2.1 Hardware Specifications

The system was hosted and run on a personal computer with the specifications listed in Table 5.1.

Table 5.1: Hardware specifications of the development and implementation PC.

| | |
|---|---|
| **CPU** | Intel® Core™ i7-8550U @1.99GHz |
| **Memory** | 16.0GB 2133MHz Dual Channel RAM |
| **GPU** | Intel® UHD Graphics 620 |
| **Connectivity** | Intel® Dual Band Wireless-AC 8265 |
| **Storage** | 360GB PCIe 3.0 x4, NVMe SSD |

Due to the Memory consuming nature of the implemented Machine Learning algorithm as shown in Figure 5.1, where some of the contents of RAM were being swapped to non-volatile storage by the OS, the developers recommend the specifications in Table 5.2 for the server hosting the web application.

Table 5.2: Recommended Hardware Specifications.

| | |
|---|---|
| **Memory** | 64GB or more |
| **CPU** | Quad-Core or higher |
| **Storage** | At least 128GB |

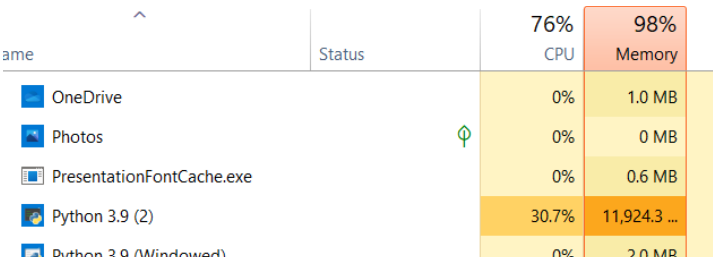



Figure 5.1: A case of high RAM usage by the python ML algorithm.

#### 5.2.2 Software Specifications

The system was implemented and assessed on a PC running Windows 11 Pro, version 21H2. It is recommended to run the system on a secure install of at least Windows 10.

The client-side web browser used to interact with the system hosted on the localhost server was Google Chrome, version 96.0.4664.110. Listed in Table 5.3 are the minimum recommended browser versions necessary to run all the CSS, jQuery, Bootstrap, and HighChart files.

Table 5.3: Minimum recommended web-browser versions.

| Browser | Recommended Version |
|---|---|
| Google Chrome | 93 |
| Mozilla Firefox | 91 |
| Opera | 78 |
| Safari | 14 |
| Microsoft Edge | 92 |

### 5.2.3 Applications Installed

#### i. XAMPP

XAMPP is an open-source cross-platform web server solution consisting of the Apache HTTP server, MariaDB database and interpreters for the scripts written in the PHP language (Apache Friends, 2021). XAMPP for Windows with PHP version 8.0.13 was installed on the implementation PC to enable local web server functionality.

Once the installer was downloaded from the official site; https://www.apachefriends.org/download.html; it was run and the on-screen instructions followed to install the application, as well as the choice of the installation folder, where the scripts of the system reside.

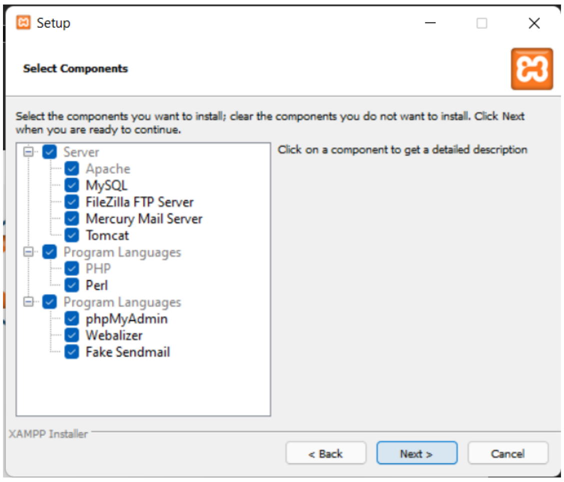



Figure 5.2: Selection of components to install with XAMPP.

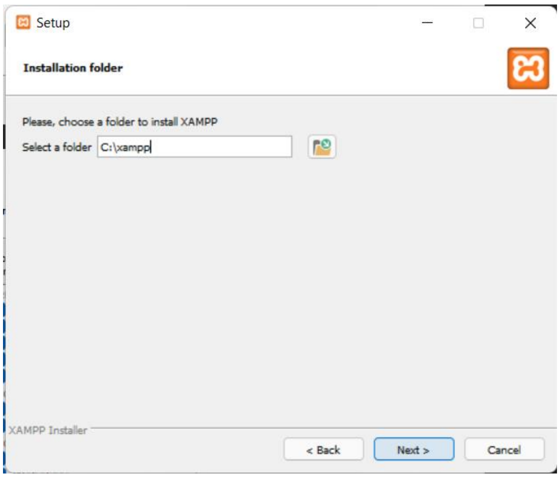
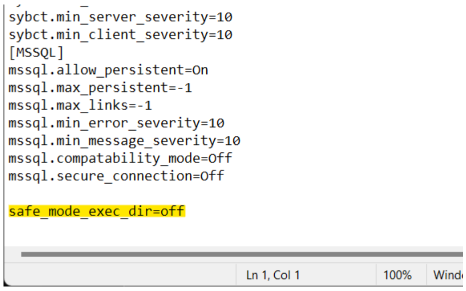



Figure 5.3: Choice of installation folder to install XAMPP.

After installation, the XAMPP control panel was opened and the ‘php.ini’ file contained in the config menu of Apache opened. The text line “safe_mode_exec_dir = off” was added to enable PHP scripts run python scripts automatically via command prompt.


Figure 5.4: Locating the php.ini file.



```
sybct.min_server_severity=10
sybct.min_client_severity=10
[MSSQL]
mssql.allow_persistent=On
mssql.max_persistent=-1
mssql.max_links=-1
mssql.min_error_severity=10
mssql.min_message_severity=10
mssql.compatability_mode=Off
mssql.secure_connection=Off

safe_mode_exec_dir=off
```



Figure 5.5: Modification of the php.ini file by adding the highlighted text.

### ii. Python

Python version 3.9 interpreter and runtime was downloaded and installed, free of charge from the Microsoft Store. This application is what enables the running of the python ML script housing the K-Means algorithm used to cluster related items and make relevant recommendations to the customer based on items in their cart.

To connect the python ML script to the web application scripts, values had to be passed from the PHP scripts to the python script via command prompt. To effectively run a python file via command prompt, “app execution aliases” was searched on the start menu and ‘app installers’ for python turned off to not have them invoked by the ‘python3’ keyword when running python files from command prompt.

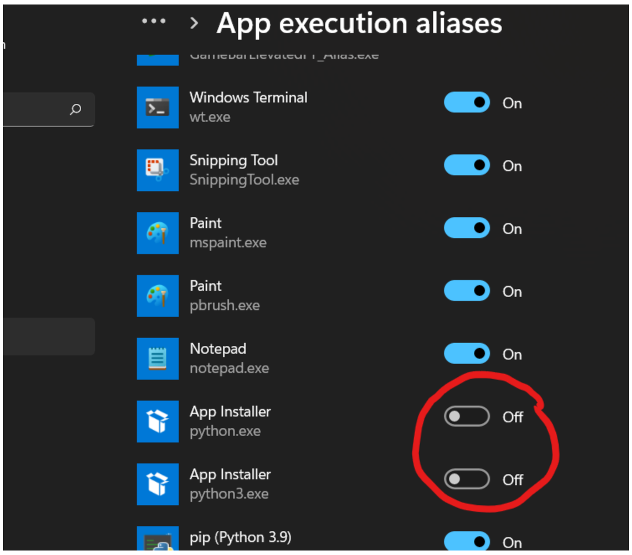



Figure 5.6: Turning off app execution aliases for python installers.

To enable the python script read data from the local database, mysql-connector library was installed in python by typing the command ‘pip3 install mysql-connector’ in Windows Powershell as shown in Figure 5.7.

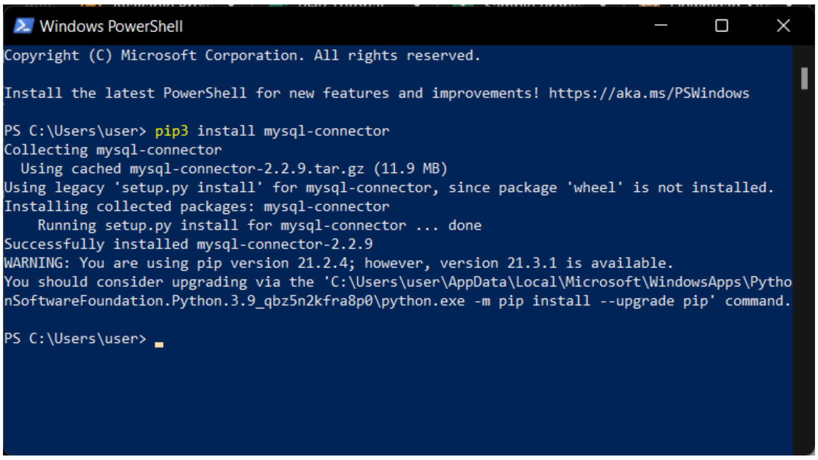



Figure 5.7: Installation of the python mysql-connector library.

The K-Means algorithm was imported from Scikit-learn, a free ML library for the python programming language (Pedregosa et al., 2011). The library was installed in python by running the command line ‘pip3 install -U scikit-learn spicy matplotlib’ in Windows Powershell as demonstrated in Figure 5.8.

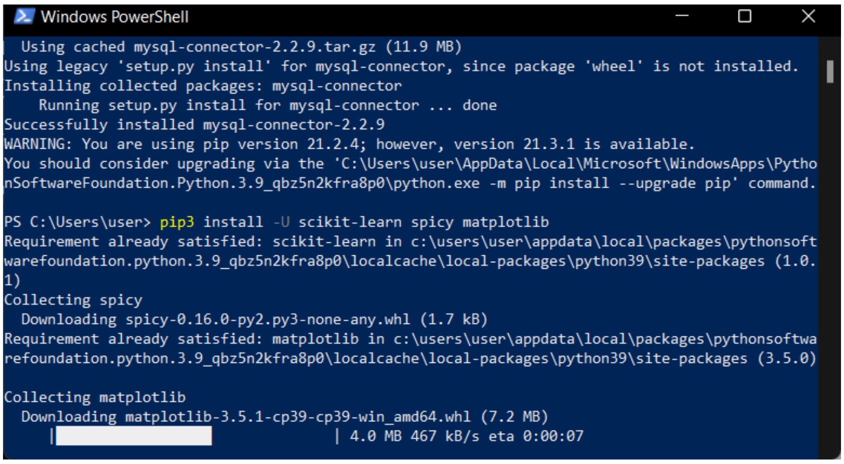



Figure 5.8: Installation of the Scikit-learn python ML library.

## 5.3 Description of the Test Environment

### 5.3.1 Testing Paradigms

All implemented modules were subjected to thorough unit testing before being subjected to integration testing to determine whether they communicate effectively. The basic logic of the individual modules as well as the system were subjected to white box testing. The payment module was subjected to grey box testing. However, due to the complex nature of the machine learning algorithm, it was subjected to black box testing as discussed below.

### i. Authentication Modules

For the administrator module, testing was done with both incorrect and correct credentials to determine when access was authorized.

The customer/default module was evaluated with both correct and incorrect credentials as well. In addition, the response of the system to blank fields during a new customer sign up, a sign up of a customer with an e-mail address already registered in the system, and an e-mail address in the wrong text format were observed.

The developers also assessed if the system correctly identifies the logged in user account based on the given credentials.

### ii. Account Edit Module

This was assessed to see whether it auto fills fields with the correct previous details set up by the customer and if the new values input by the customer reflect in their account details accurately.

### iii. Feedback Module

Once a customer submits a feedback message to the administrators, it was observed if the message was accurately reflected in the administrator's dashboard with the correct matching e-mail address of the customer who sent it.

### iv. Products Module

This was tested to determine whether each product was displayed in its correct category as well as being seamlessly addable to the cart. Evaluated to observe whether an item out of stock appears on the site for purchase.

### v. Cart Module

A very crucial module that was tested to determine whether it can hold in memory items selected and reflect their correct attributes in a summary form, easy to read by the customer. In addition, this was assessed to see if it would allow the ordering of more of a specific item than what is currently in stock or allow a purchase to be made in the event the cart was empty. The mathematical logic of the reflected costs was also confirmed.

### vi. Purchase Confirmation Module

This module was evaluated to observe the reflecting of the order on the administrator dashboard with the correct items, number of units bought, and their total cost showing accurately on a printable receipt.

### vii. Payment Module

Subjected to grey box testing. The module was assessed to see its manipulation on the database when an order placed was fully paid for, and when a payment was cancelled.

### viii. Recommender Module

After insertion of diverse dummy transactions by the developers to feed the K-Means ML algorithm with sufficient training data, this was subjected to black box testing to see if the recommended items are close to what the developers expect based on the dummy transactions.

Moreover, it was evaluated to see if it would recommend an item already selected and placed in the cart. The runtime was also observed to determine if the speed requirements are satisfied.

### ix. Administrator Dashboard Module

Once a huge purchase quantity was made by a few customers of one specific item, this was assessed to see if the popularity of the said item would reflect by its rank and brand share being of higher significance in the ranking list.

### x. Administrator Stock Management Module

Evaluated to observe the behavior of ‘Stock In’ and ‘Stock Out’ actions on the stock quantity. This was also tested to observe the response of the system on addition of a new item to inventory.

### 5.3.2 Test of Functional Requirements

#### i. Authentication Module

Table 5.4: Authentication module testing results.

| Test Case | Description | Test Data | Expected Outcome | Actual Results | Test Verdict | Reference Screenshot |
|---|---|---|---|---|---|---|
| 1 | Verify that users cannot register with an existing email address. | Email address ‘maina@gmail.com’ that is in use by another customer used to register a new user. | An alert shown that the email already exists. | Alert message, ‘Email already exists,’ displayed. | Pass | Figure E.6 |
| 2 | Verify that a user cannot register with empty fields. | Full name is ‘Nairobi Delmonte’, email address is left blank, phone number is ‘0723667342’, password is ‘left empty’, and location is ‘Kasarani’. | An alert shown that the user needs to fill in the empty field. | Alert message ‘Please fill in this field’ pointing to the e-mail text box. | Pass | Figure E.5 |
| 3 | Verify that a user cannot register with an incorrect e-mail format. | Full name is ‘Nairobi Delmonte’, email address is ‘delmonte’, phone number is ‘0115456743’, password is ‘left empty’, and location is ‘Mathare’. | An alert will be shown that the user needs to correct their e-mail address. | Alert message “Please include an ‘@’ in the email address” pointing to the e-mail text box. | Pass | Figure E.7 |
| 4 | Verify that a user can login with correct credentials. | Email address is ‘maina@gmail.com’ and the password is ‘joshua’. | User will be redirected to the respective pages depending on the user type. | Successful redirection to the home page. | Pass | Figure E.2 |
| 5 | Verify that a user cannot login with invalid credentials. | Email address entered is “maina@gmail.com” and the password is ‘yosh’. | An alert will be shown that the user or the password is invalid. | Alert message ‘Invalid Email or Password’ displayed. | Pass | Figure E.3 |

### ii. Account Edit Module

Table 5.5: Account page testing results.

| Test Case | Description | Test Data | Expected Outcome | Actual Results | Test Verdict | Reference Screenshot |
|---|---|---|---|---|---|---|
| 6 | Verify that details are captured by the system upon registration. | Full name is “Yvonne Miriam”, email address is “yvonne@gmail.com”, phone number is “0755775563”, password is “yvonne”, and location is “Ngong”. | Customer details accurately reflected in the ‘My Account’ tab when the user clicks on their name in the top right corner of the home page. | Name: Yvonne Miriam Address: Ngong Mobile No: 0755775563 Email: yvonne@gmail.com | Pass | Figure E.4 |
| 7 | Verify that a user can update their details. | Full name was “Yvonne Miriam” updated to “Yvonne Smith”. Address changed from “Ngong” to “Kasarani”. | Name and location change will display on the profile home page. | Name: Yvonne Smith Address: Kasarani Mobile No: 0755775563 Email: yvonne@gmail.com | Pass | Figure E.8 and Figure E.9 |
| 8 | Verify that a user can reset their password if necessary. | Old password is “joshua” and new password is “maina”. | Profile password will be changed. | Successful login with new password. | Pass | Figure E.1 and Figure E.2 |

### iii. Feedback Module

Table 5.6: Feedback module testing results.

| Test case | Description | Test Data | Expected Outcome | Actual Results | Test Verdict | Reference Screenshot |
|---|---|---|---|---|---|---|
| 9 | Verify that a specific customers e-mail address sent feedback | Email address is “yvonne@gmail.com” | Administrator will be able to see which account the feedback came from. | Display of ‘yvonne@gmail.com’ in the Email column. | Pass | Figure E.10 and Figure E.11 |
| 10 | Verify that customers email address sent a feedback message. | Message from ‘yvonne@gmail.com’ is “I ordered shoes, but the size delivered does not fit. Is it possible to get a refund?” | Administrator will be able to see the message that the account sent. | Successful display of the feedback message in the Message column. | Pass | Figure E.10 and Figure E.11 |

### iv. Products Module

Table 5.7: Product module testing results.

| Test case | Description | Test Data | Expected Outcome | Actual Results | Test Verdict | Reference Screenshot |
|---|---|---|---|---|---|---|
| 11 | Verify that product is in its correct category | “Apple Airpods” are in the category for “Electronics” | A pair of Apple Airpods will be in the Electronics slot in the Product page | Apple Airpods present only on the Electronics tab of the product catalog. | Pass | Figure E.15 |
| 12 | Verify that selected products can be added to the cart | Adding a ‘PlayStation 5’, three A4 single ruled books, and an ‘iPhone 13 Pro Max’ to cart. | All selected items to be held and saved in the cart. | Cart contains one PlayStation 5, three A4 single ruled books, and one iPhone 13 Pro Max. | Pass | Figure E.16 |
| 13 | Verify that out-of-stock products do not appear on the product page | Stock of ‘iPhone 13’ set to zero. | ‘iPhone 13’ will not show in the Electronics Category | ‘iPhone 13’ missing from the electronics catalog. | Pass | Figure E.15 |

### v. Cart Module

Table 5.8: Cart module testing results.

| Test case | Description | Test Data | Expected Outcome | Actual Results | Test Verdict | Reference Screenshot |
|---|---|---|---|---|---|---|
| 14 | Verify items' correct attributes. | One PlayStation 5, three A4 single ruled books, and one iPhone 13 Pro max added to cart. | Correct images, names, quantity, and prices displayed on the cart module. | Correct details displayed with a total cost of Ksh.195,630. | Pass | Figure E.16 |
| 15 | Verify the addition or reduction of an item on the cart | Clicking the plus icon with one 'iPhone 13' already in the cart. | 2 'iPhone 13's in cart. | Quantity of 'iPhone 13' in cart changed from 1 to 2. | Pass | Figure E.13 |
| 16 | Verify a user cannot add to cart more than what is in stock. | Setting the stock of iPhone 13 to 2 and clicking the plus icon when two iPhone 13s are already in the cart. | Inability to add more of the product. | Alert message "Sorry, only 2 iPhone 13's in stock." Displayed with no change to the cart. | Pass | Figure E.13 and Figure E.14 |

### vi. Purchase Confirmation Module

Table 5.9: Purchase confirmation module testing results.

| Test case | Description | Test Data | Expected Outcome | Actual Results | Test Verdict | Reference Screenshot |
|---|---|---|---|---|---|---|
| 17 | Confirm an order and show the items on a receipt. | A customer orders Bose headphones, an XBOX, an XBOX controller, 4 Krangles potato crisps, and three bottles of Coca-Cola diet coke, and pays via PayPal. | Receipt will display the number of items purchased and the respective costs. | Successful display of receipt items with a total cost of Ksh.100,356 on the administrator transaction dashboard. | Pass | Figure E.27 and Figure E.31 |

### vii. Payment Module

Table 5.10: Payment module testing results.

| Test case | Description | Test Data | Expected Outcome | Actual Results | Test Verdict | Reference Screenshot |
|---|---|---|---|---|---|---|
| 18 | Verify when an order has paid for. | Customer makes the order in Figure E.27 and pays via PayPal. | PayPal business merchant account reflects a received payment, and the transaction is marked confirmed on the administrator dashboard. | PayPal business merchant account shows a received payment of $886.93 USD on 26 Dec 2021 at 8:03AM. | Pass | Figure E.28, Figure E.29, and Figure E.30 |
| 19 | Verify cancellation of an order. | Customer cancels an order before completion of the payment process. | Transaction cancelled with no stock updates. | Transaction marked cancelled on the administrator dashboard. | Pass | Figure E.32 and Figure E.33 |
| 20 | Detect when the PayPal account balance is insufficient for an order. | An account with a balance of $26.50 USD making an order of $669.60 USD. | Suggest a different form of payment. | PayPal uses the remaining account balance and charges what is left to the customer's debit card. | Pass | Figure E.34 |

### viii. Recommender Module

Table 5.11: Recommender module testing results.

| Test case | Description | Test Data | Expected Outcome | Actual Results | Test Verdict | Reference Screenshot |
|---|---|---|---|---|---|---|
| 21 | Suggest related products based on items in the cart. | 'PlayStation 5, A4 single ruled book, and 'iPhone 13 Pro Max' in cart. | Recommended items that make sense. | AirPods pro and iPhone 13 Pro Max case, black and blue BIC pens, a PlayStation 5 controller and wireless headphones recommended in the 'You may also like…' section. | Pass | Figure E.16 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 22 | Not suggest items already in the cart. | Adding the recommended Dualsense controller to cart | Item added to cart removed from recommendations. | Dualsense controller no longer in the recommended list. | Pass | Figure E.17 |

### ix. Administrator Dashboard Module

Table 5.12: Admin dashboard module testing results.

| Test case | Description | Test Data | Expected Outcome | Actual Results | Test Verdict | Reference Screenshot |
|---|---|---|---|---|---|---|
| 23 | Verify product and brand ranking. | Have a customer make a large order of forty black BIC pens. | BIC brand share to increase and black BIC pens to be the top selling product. | Administrator dashboard shows BIC brand share leading at 17.2% and BIC black ball point pens as the top selling product with 48 total units sold. | Pass | Figure E.18 |
| 24 | Verify deactivation of customer account. | Administrator deactivates the customer with the e-mail adallah@gmail.com. | Deactivated customer does not appear in the customer list and is unable to login. | Adallah missing from the customer list on the administrator dashboard and cannot login as a customer. | Pass | Figure E.25 and Figure E.26 |

### x. Administrator Stock Management Module

Table 5.13: Stock management module testing results.

| Test case | Description | Test Data | Expected Outcome | Actual Results | Test Verdict | Reference Screenshot |
|---|---|---|---|---|---|---|
| 25 | Observe the Behavior of “Stock In” and “Stock Out” operations. | Stock in two leather jackets with seven originally in stock. | Increases in the product stock by the specified amount. | Leather jacket stock changed from 7 to 9. | Pass | Figure E.19 and Figure E.20 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 26 | Addition of a new product to the catalog. | Adding Oreo dipped cookies. Category: Snacks. Unit cost: Ksh.150. Brand: Cadbury. Stock: 33. | Added product to appear in inventory. | Oreo dipped cookies appears in the snacks catalog of the products module. | Pass | Figure E.21 and Figure E.22 |
| 27 | Verify discontinuation of a product previously on sale. | Discontinue 'Samsung Galaxy Watch Active' in the accessories category. | Discontinued product disappears from the product catalog. | Samsung Galaxy watch active no longer appears on the accessories inventory. | Pass | Figure E.23 and Figure E.24 |

### 5.3.3 Test of Non-Functional Requirements

Table 5.14: Non-Functional requirements testing.

| Test case | Description | Test data | Expected outcome | Actual Results | Test Verdict | Reference Screenshot |
|---|---|---|---|---|---|---|
| 28 | Navigation | Finding Apple AirPods. | Easy to find in alphabetically ordered Electronics tab. | Insignificant time spent locating the item by the customer. | Pass | Figure E.15 |
| 29 | Readability | Reading text on the products module | Ease to read based on the use of contrasting colors. | No user complaints on readability. | Pass | Figure E.15 |
| 30 | Speed: Cart module load time from sleep. | Loading the cart module on the system just turned on shortly before. | A page load time of at most 0.5 seconds. | Content loaded in 2.36s. Issue discussed in Recommendations. | Fail | Figure E.35 |
| 31 | Principle of least privilege. | Trying to edit a customer's details from the administrator module. Trying to edit a product stock from the customer module. | Administrator and customer cannot do more than necessary. | Administrator does not have a means to edit customer details. Customers do not have a means to edit the inventory. | Pass | Figure E.12 and Figure E.26 |

## 5.4 Discussion of Results

From the K-Means source code in Appendix C: the number of clusters (n_clusters) initialized directly affect the quality of the K-Means clustering algorithm (scikit-learn developers, 2008). The optimum number of clusters for this system, with its 53 products, was found to be around 9, a high enough value for the recommendations to be sensible enough to developers but not too high a value for there to be several clusters with only a single element.

The maximum number of iterations (max_iter) that the K-Means algorithm runs to minimize the WCSS was left at its default value of 300 as it did not affect the runtime of the algorithm by a significant margin.

# Chapter 6: Conclusions and Recommendations

## 6.1 Challenges

The concept of clustering and ML in general being a novel idea to the developers caused a delay in the delivery of the first system prototype due to time consumption of research conducted.

Initial plans to use the PHP built-in K-Means clustering algorithm were scrapped as it did not include the features necessary for this project. Additional research had to be done on how to connect PHP scripts with different programming languages, further delaying project deliverables.

Using Apache NetBeans IDE to code the ML algorithm in Java proved futile as it was challenging to maintain a reliable database connection.

Python, however, proved to be a better option with the language being better at handling arrays and lists. Despite that, a unique set of challenges was encountered where research had to be done on stack overflow, W3Schools, and GeeksforGeeks to enable the python script to communicate automatically with the PHP scripts, as outlined in 5.2.3 on page 32.

The single PHP session handler global variable, $_SESSION, meant only either an admin or a customer, not both, could be logged in at any given time.

## 6.2 Conclusions

As the main objective of this study was to implement a K-Means clustering algorithm on a mock web-based e-commerce platform, not only does the system sufficiently satisfy its goal, but it also demonstrates the quick linear runtime of the K-Means algorithm since negligible time is taken in clustering items and the source code making decisions based on the clusters generated.

The mock e-commerce platform, though missing few insignificant features present in a fully-fledged e-commerce platform, is sufficient to provide a realistic interface with which to interact with the K-Means clustering algorithm and demonstrate its functionality.

## 6.3 Recommendations

Although the K-Means clustering ML python script was invoked every time a recommendation of a related item was to be made, it was purely for academic demonstration purposes. Taking note of the runtime of the ML algorithm as demonstrated in Table 5.14, it is recommended to have it run during low internet traffic hours, such as midnight, and the output array list containing the product IDs as keys and the cluster they have been placed into as values be saved and used for every transaction within the day, before a new cluster list is generated using the new transaction data generated during the day as well as historical data.

### 6.4 Future Works

This documentation suggests research be done to develop an improved K-Means algorithm that utilizes less computer memory in its operation.

A suggestion is also made to develop a version of the K-Means clustering algorithm that can not only work with numerical data, but also text and non-discrete analog data.

## References


Advani, V. (2021, October 19). *What is Artificial Intelligence? How does AI work, Types and Future of it?* Great Learning. https://www.mygreatlearning.com/blog/what-is-artificial-intelligence/

Apache Friends. (2021). *XAMPP Apache + MariaDB + PHP + Perl*. Apache Friends. https://www.apachefriends.org/index.html

Booch, G., Maksimchuk, R., Engle, M., Young, B., Conallen, J., & Houston, K. (2007). *Object-Oriented Analysis and Design with Applications* (Third). Addison-Wesley Professional.

Bootstrap Core Team. (2021). *Build fast, responsive sites with Bootstrap*. Getbootstrap. https://getbootstrap.com/

Brownlee, J. (2016, March 16). *Supervised and Unsupervised Machine Learning Algorithms*. Machine Learning Mastery. https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/

Central Bank of Kenya. (2021, December 29). *Foreign Exchange Rates*. CBK | Central Bank of Kenya. https://www.centralbank.go.ke/rates/forex-exchange-rates/

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*(8), 790–799. https://doi.org/10.1109/34.400568

Dey, D. (2021, October 6). *ML | Hierarchical clustering (Agglomerative and Divisive clustering)*. GeeksforGeeks. https://www.geeksforgeeks.org/ml-hierarchical-clustering-agglomerative-and-divisive-clustering/

Fan, K.-T., Tzeng, Y.-C., Lin, Y., Su, Y.-J., & Chen, K.-S. (2010). *Tree identification using a distributed K-mean clustering algorithm*. 3446–3449. https://doi.org/10.1109/IGARSS.2010.5654381

Kissflow Inc. (2021, September 5). *Rapid Application Development (RAD): Changing How Developers Work*. Kissflow. https://kissflow.com/low-code/rad/rapid-application-development/

Kriegel, H.-P., Schubert, E., & Zimek, A. (2017). The (black) art of runtime evaluation: Are we comparing algorithms or implementations? *Knowledge and Information Systems*, *52*, 341–378.

Li, Y., & Wu, H. (2012). *A Clustering Method Based on K-Means Algorithm*. *25*, 1104–1109. https://doi.org/10.1016/j.phpro.2012.03.206

MacDonald, J. (2021, April 19). *21 Effective Product Recommendation Tips (That Increase Conversions)*. The Good. https://thegood.com/insights/ecommerce-product-recommendation/

Mastercard Worldwide Insights. (2008). *Online shopping in Asia-Pacific – patterns, trends and future growth*. www.mastercard.com/us/company/en/insights/studies/2008/asiaonlineshopping.html

Mitchell, T. (1997). *Machine Learning* (1st ed.). McGraw-Hill Education.

PayPal Developer. (2019, June 12). *PayPal sandbox testing guide*. PayPal Developer. https://developer.paypal.com/docs/api-basics/sandbox/

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Priy, S. (2021, September 22). *Clustering in Machine Learning*. GeeksforGeeks. https://www.geeksforgeeks.org/clustering-in-machine-learning/

Russell, S., & Norvig, P. (2015). *Artificial Intelligence: A Modern Approach*. Pearson Education Limited.

Sammut, C., & Webb, G. (2011). *Encyclopedia of Machine Learning*. Springer Science+Business Media.

scikit-learn developers. (2008). *Sklearn.cluster.KMeans*. Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

Seif, G. (2018, February 5). *The 5 Clustering Algorithms Data Scientists Need to Know*. Towards Data Science. https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68

Sinaga, K., & Yang, M.-S. (2020). Unsupervised K-Means Clustering Algorithm. *IEEE Access*, *8*, 80716–80727. https://doi.org/10.1109/ACCESS.2020.2988796

Singhal, A. (2020). *K-Means Clustering Algorithm | Examples*. GateVidyalay. https://www.gatevidyalay.com/k-means-clustering-algorithm-example/

Tripathi, A. (2019, May 16). *ML | Mean-Shift Clustering*. GeeksforGeeks. https://www.geeksforgeeks.org/ml-mean-shift-clustering/

Wang, N., Liu, D., & Cheng, J. (2008). Study on the Influencing Factors of Online Shopping. *Proceedings of the 11th Joint Conference on Information Sciences (JCIS 2008)*, 497–500. https://doi.org/10.2991/jcis.2008.84

# Appendix A: Gantt Chart

WEB BASED AI SHOPPING SYSTEM

| # | Activity | | | DURATION Start | DURATION End |
|---|---|---|---|---|---|
| 1 | Overview of Project | 27-Sep | 01-Oct | 5 | 5 |
| 2 | Abstarct and Chapter 1 | 04-Oct | 08-Oct | 10 | 10 |
| 3 | Chapter 2 and Referencing | 11-Oct | 15-Oct | 15 | 15 |
| 4 | Chapter 3 and Referencing | 18-Oct | 22-Oct | 20 | 20 |
| 5 | Proposal Submission | 28-Oct | 29-Oct | 25 | 25 |
| 6 | Chapter 4 and Supervisor Feedback | 08-Nov | 12-Nov | 30 | 30 |
| 7 | Works | 15-Nov | 17-Dec | 30 | 65 |
| 8 | Development Sprint 1 | 25-Nov | 26-Nov | 70 | 70 |
| 9 | Development Sprint 2 | 02-Dec | 03-Dec | 75 | 75 |
| 10 | Development Sprint 3 | 09-Dec | 10-Dec | 80 | 80 |
| 11 | Final Presentation | 06-Jan | 06-Jan | 85 | 85 |
| 12 | Final Documentation Submission | 07-Jan | 10-Jan | 90 | 90 |

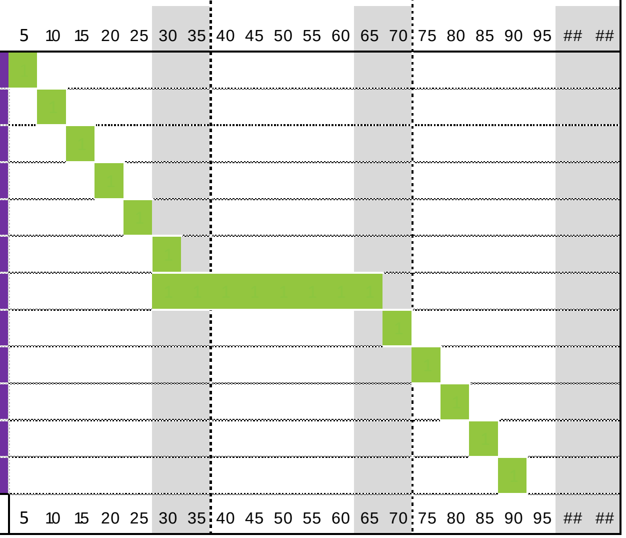



Figure A.1: Project Gantt chart.

## Appendix B: RAD User Design Process Issues and Fixes

Table B.1: Discovered bugs and kinks and corresponding fixes.

| Prototype | Delivery Date | Issue | Fix |
|---|---|---|---|
| 1 | 23rd Nov 2021 | Difficult to read brand text on products module. | Adding a contrasting text outline using the CSS property -webkit-text-stroke-color. |
| | | Lack of visual icons. | Importing an icon pack from fontawesome. |
| | | Unsorted inventory items. | Adding an 'ORDER BY name' statement to the SQL queries. |
| | | Items in cart still appearing under recommendations. | Modifying the SQL query to not show items already in cart. |
| | | Ability to purchase when the cart is empty. | Adding a condition to display the purchase button only when items are in cart. |
| | | Prices displayed in a difficult to read format. | Adding a 'thousands' comma (,) separator to prices' text using number_format(). |
| | | Irrelevant items recommended. | Increasing the number of K-Means clusters from 5 to 9. |
| | | "Buru buru" displayed as the default address in the customer account edit module. | Address selection dropdown list modified to start with the currently registered customer address. |
| 2 | 9th Dec 2021 | Cramped text fields in the customer sign up module. | Text fields styled to be columnar with margins. |
| | | Difficult to interpret pie chart on the admin dashboard. | Pie chart modified to display brands in descending order of the number of units purchased. |
| | | Images not correctly uploaded when the admin adds a new product. | Image field in the products table of the database modified to accept longer strings. |
| | | Transaction dates in a difficult to interpret format. | Date format changed from 12-30-2021 to 30th Dec 2021 using date("format",value) |
| | | Lack of a means to filter products, transactions, or customers on the admin dashboard. | Search bar added to the admin pages. |
| 3 | 17th Dec 2021 | Ability to order more than what is in stock. | Check added to limit the maximum number of items purchasable to the number in stock. |
| | | Errors thrown when users insert text with an apostrophe e.g., "Ng'ang'a", "doesn't", into text fields. | Using addslashes($_POST['name']) to allow passing to the DB 'name' text with special characters. |

| | | | |
|---|---|---|---|
| | | Footer containing copyright text not sticking to the bottom of windows with little content. | Adding a JavaScript condition to ensure the footer is always at the bottom of the window if the content does not fill the window. |
| 4 | 21st Dec 2021 | Lack of functionality to deactivate a customer account or discontinue a product. | Status columns added to customer and products table in the DB and discontinue and deactivate buttons added to the products and customer administrator dashboards, respectively. |
| | | No connection to a payments API. | System integrated with a virtual PayPal sandbox API to simulate payments using virtual currency. |

## Appendix C: K-Means Python Source Code

```python
#importing relevant libraries
#!/usr/bin/env python
import sys
import mysql.connector
from sklearn.cluster import KMeans
import numpy as np

#establish a connection to the database
conn = mysql.connector.connect(host='localhost', database='commerce', user='root', password='')

#cursor to handle SQL statements
mycursor=conn.cursor()

mycursor.execute("SELECT * FROM transactionitems")

#disconnect from server as we are running no more MySQL queries
conn.close()

#fetch each column as an array
results=mycursor.fetchall()
cols=zip(*results)
outlist=[]
for col in cols:
arr=np.asarray(col)
type=arr.dtype
outlist.append(np.asarray(arr, np.int32))

#pop the first array with the transaction IDs
outlist.pop(0)

#n_clusters is the number of clusters items will be grouped into
#fit the arrays into their relevant clusters using kmeans.fit
kmeans=KMeans(n_clusters=9)
kmeans.fit(outlist)

#sys.argv[k] are the inputs passed via command line
numberargs=int(sys.argv[1])#no. of items in cart
k=1
similar=[]#array holding IDs of items in the same cluster

while k <= numberargs:
        idval = int(sys.argv[k+1])#productID value to check
        idval -= 1#decrement since indices of array start at 0, not 1
        j=0
        while j < len(kmeans.labels_):
              if kmeans.labels_[j]==kmeans.labels_[idval] and j!=idval:
                    similar.append(j+1)
              j+=1
        k+=1

unique_ids = list(set(similar))#eliminate duplicate IDs from the array
print(unique_ids)#output the list to command prompt for PHP to read
```

## Appendix D: Link to GitHub Repository

The most recent version of the project scripts is hosted on the GitHub server at the link https://github.com/joeotiza/commerce.

## Appendix E: Test Case Screenshots

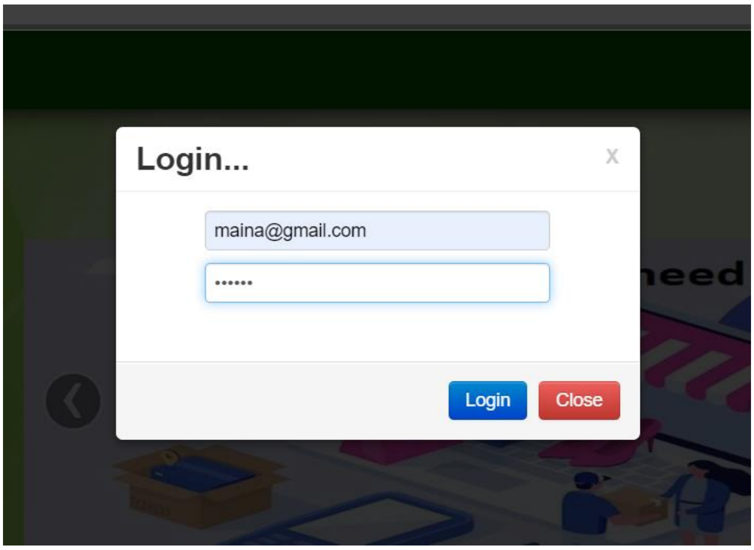



Figure E.1: Login attempt with correct credentials.

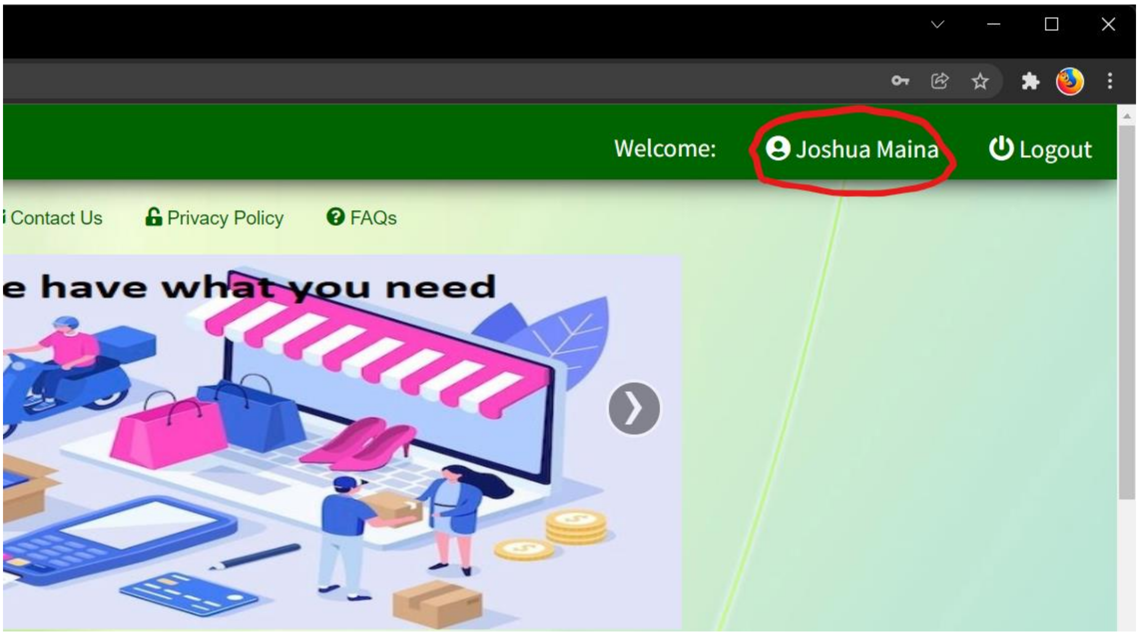



Figure E.2: Correct customer's name displayed at the top-right.

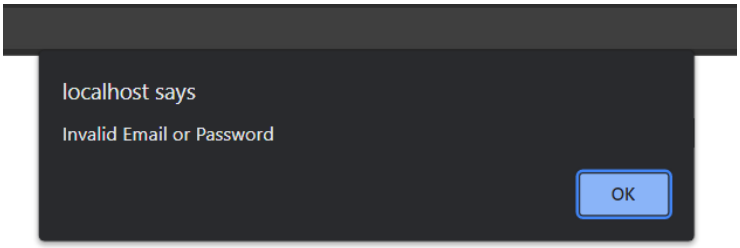



Figure E.3: Message displayed when a login is attempted with wrong credentials.

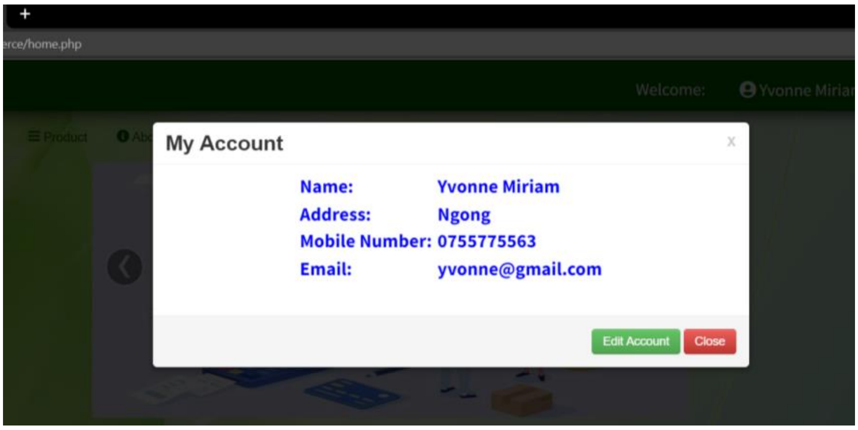



Figure E.4: Customer details tab.

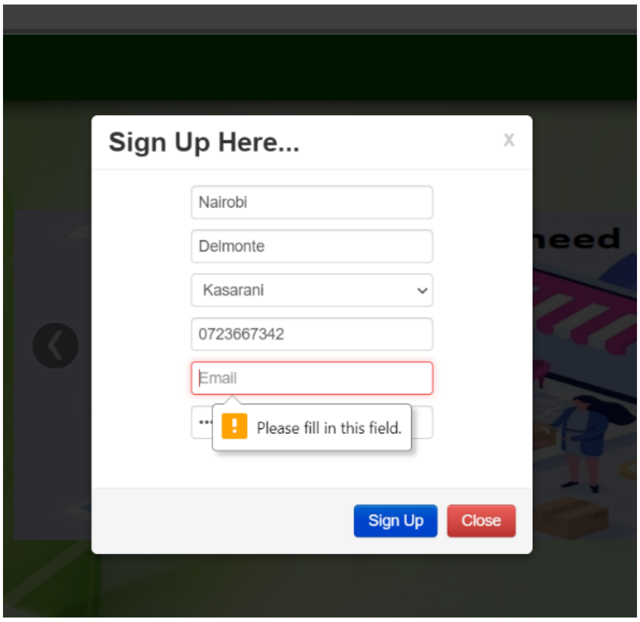



Figure E.5: System response when a field is left blank during customer signup.

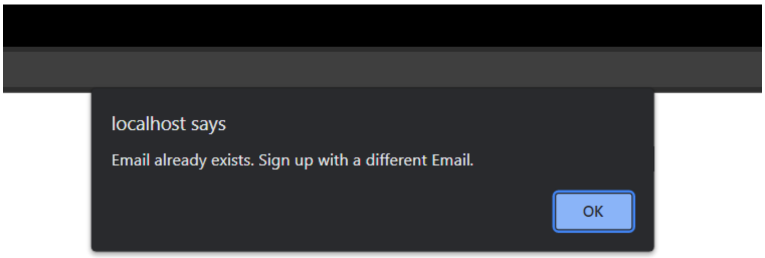



Figure E.6: Sign up attempted with a previously registered e-mail address.

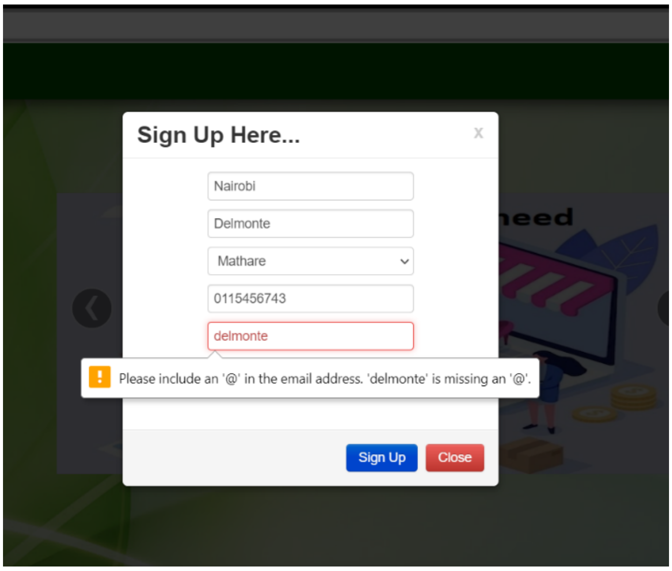



Figure E.7: System response to an incorrect email address format.

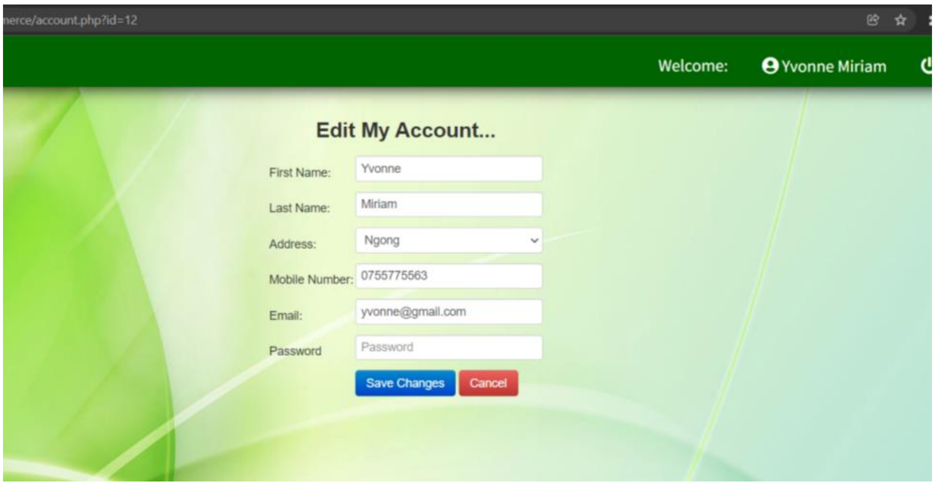



Figure E.8: Current account details in the edit account page.

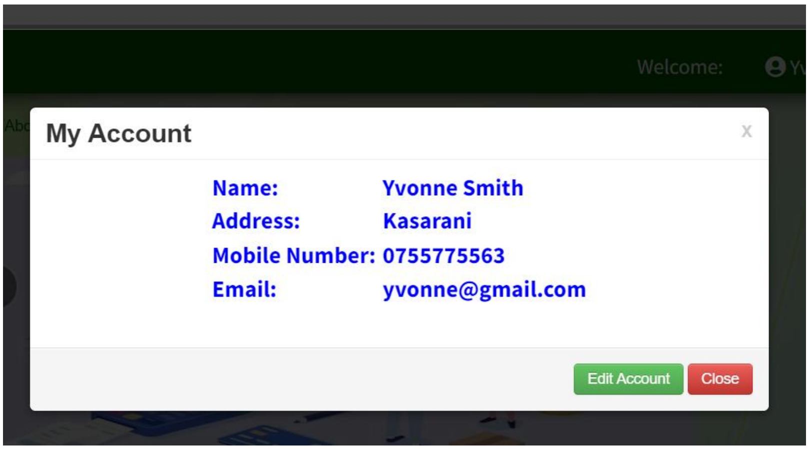



Figure E.9: Customer details after account edit.

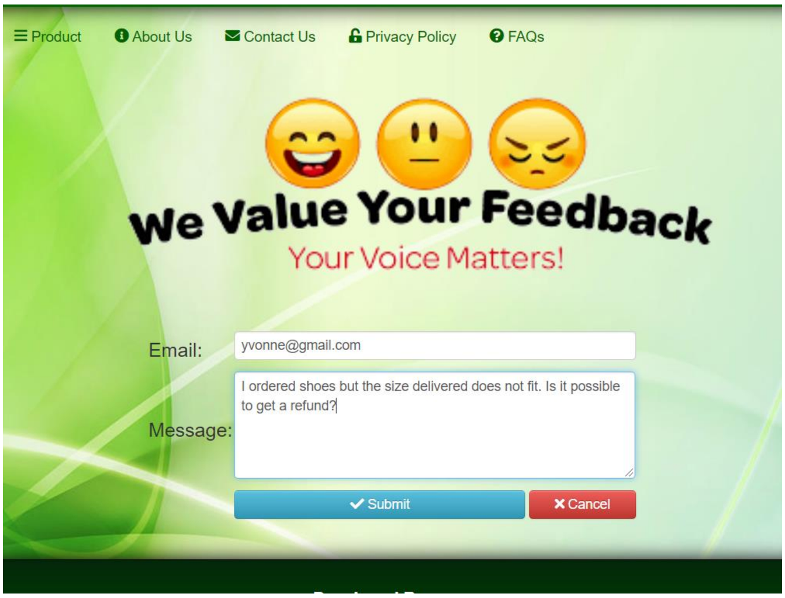



Figure E.10: Feedback submission module.

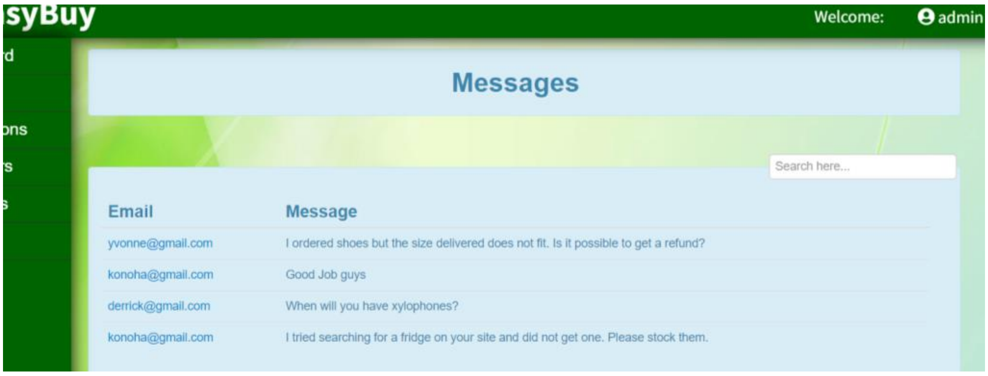



Figure E.11: Feedback messages in the administrator dashboard.

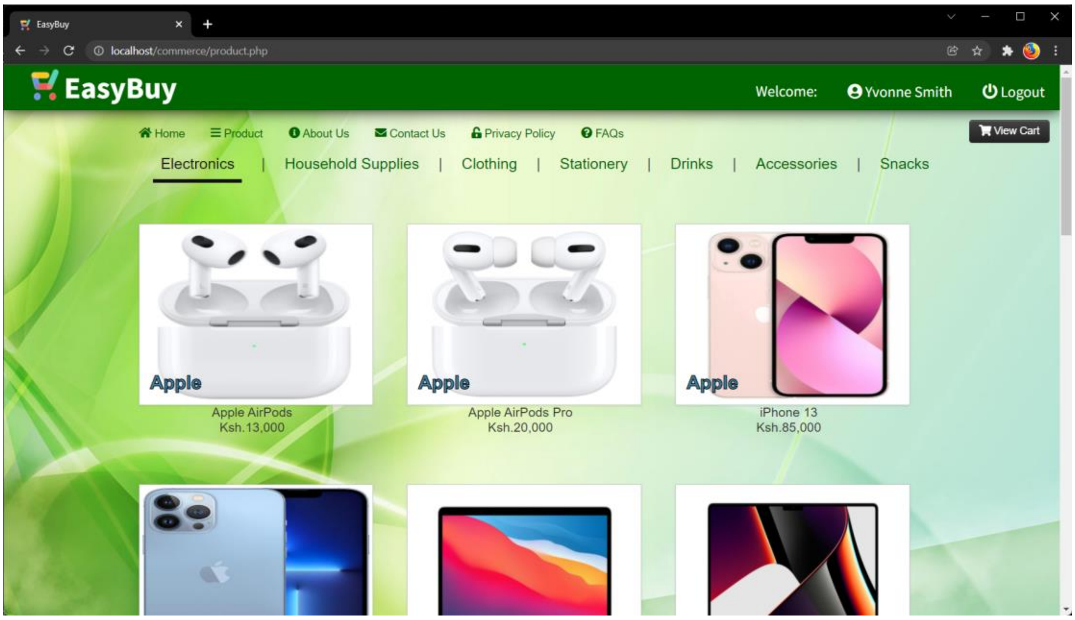



Figure E.12: Product catalog.

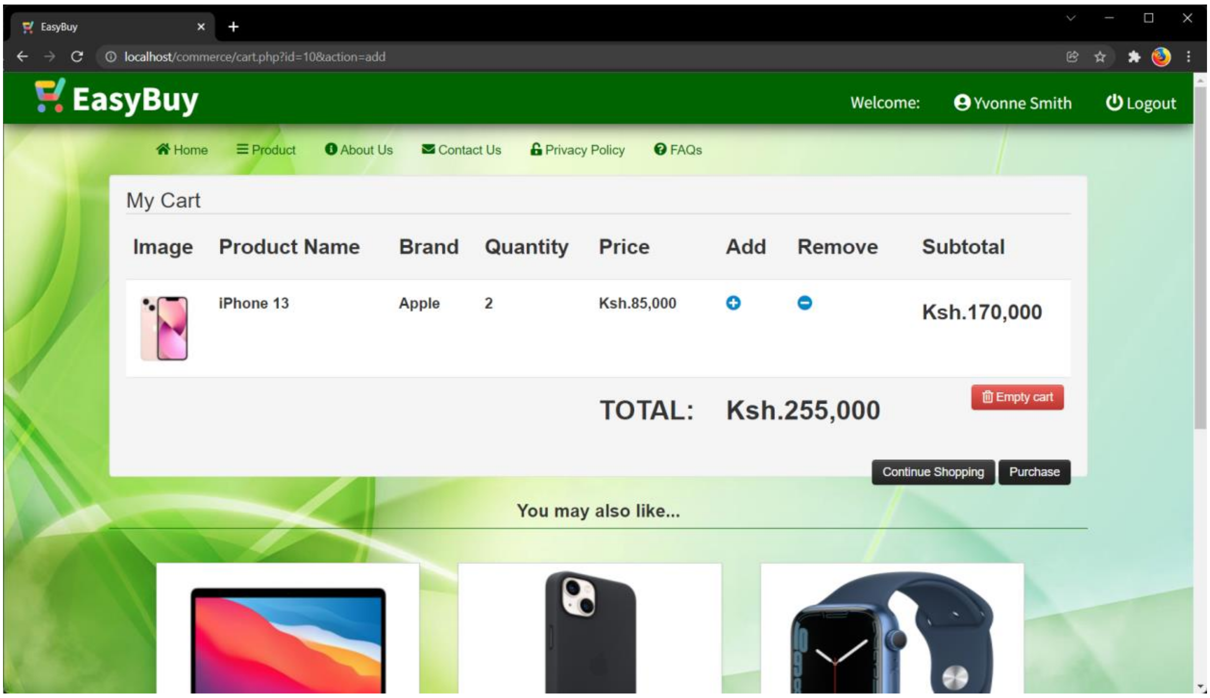



Figure E.13: Ordering all stock of iPhone 13.

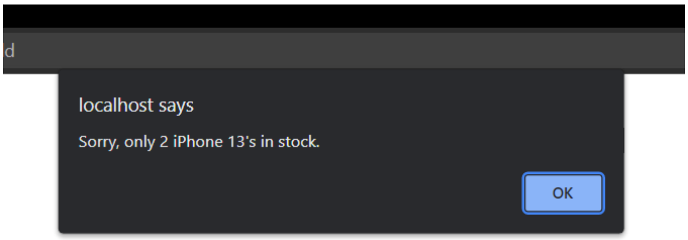



Figure E.14: Attempt to order more than in stock.

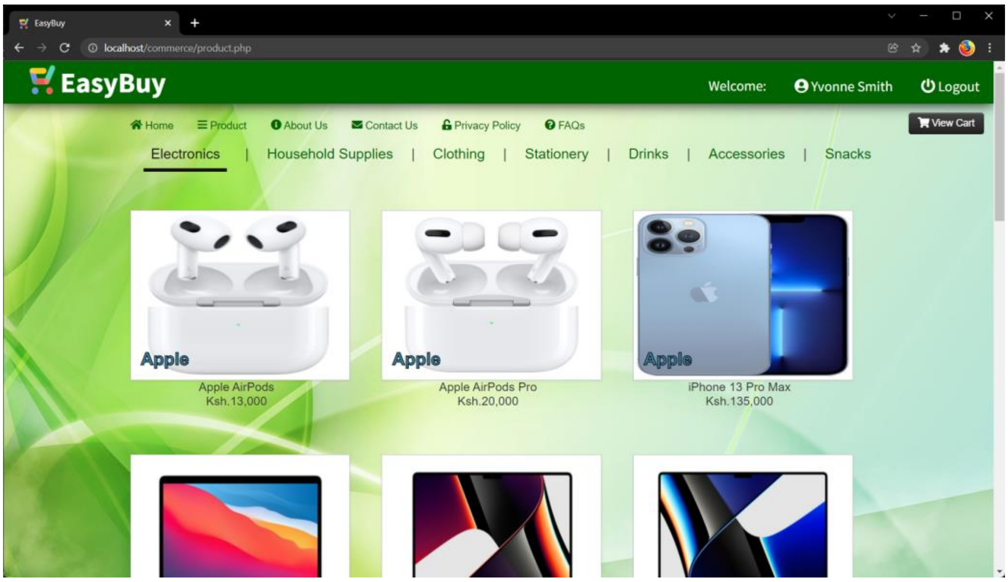



Figure E.15: Inventory missing the regular iPhone 13 that is out of stock.

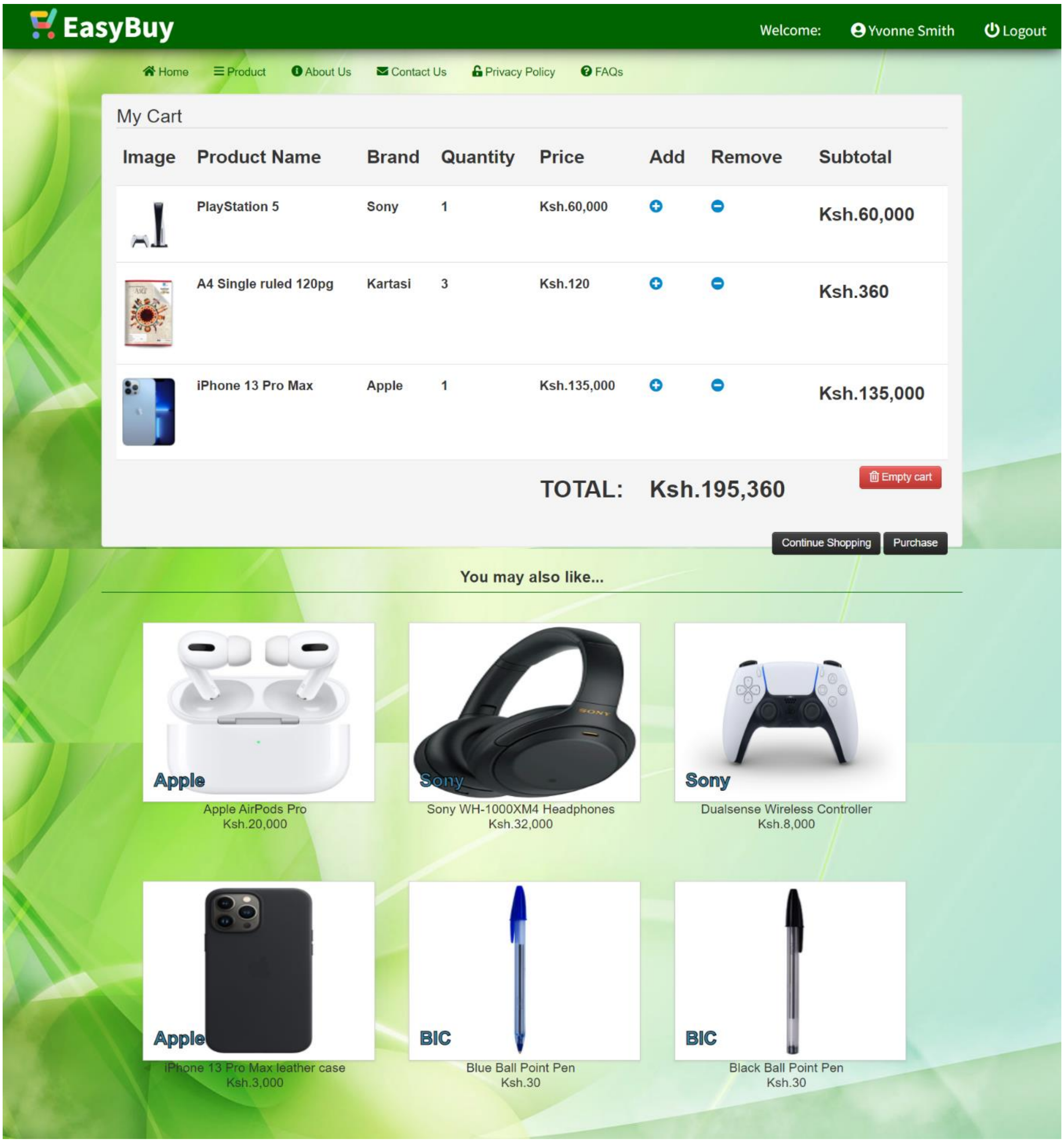



Figure E.16: Items in cart and recommended items.

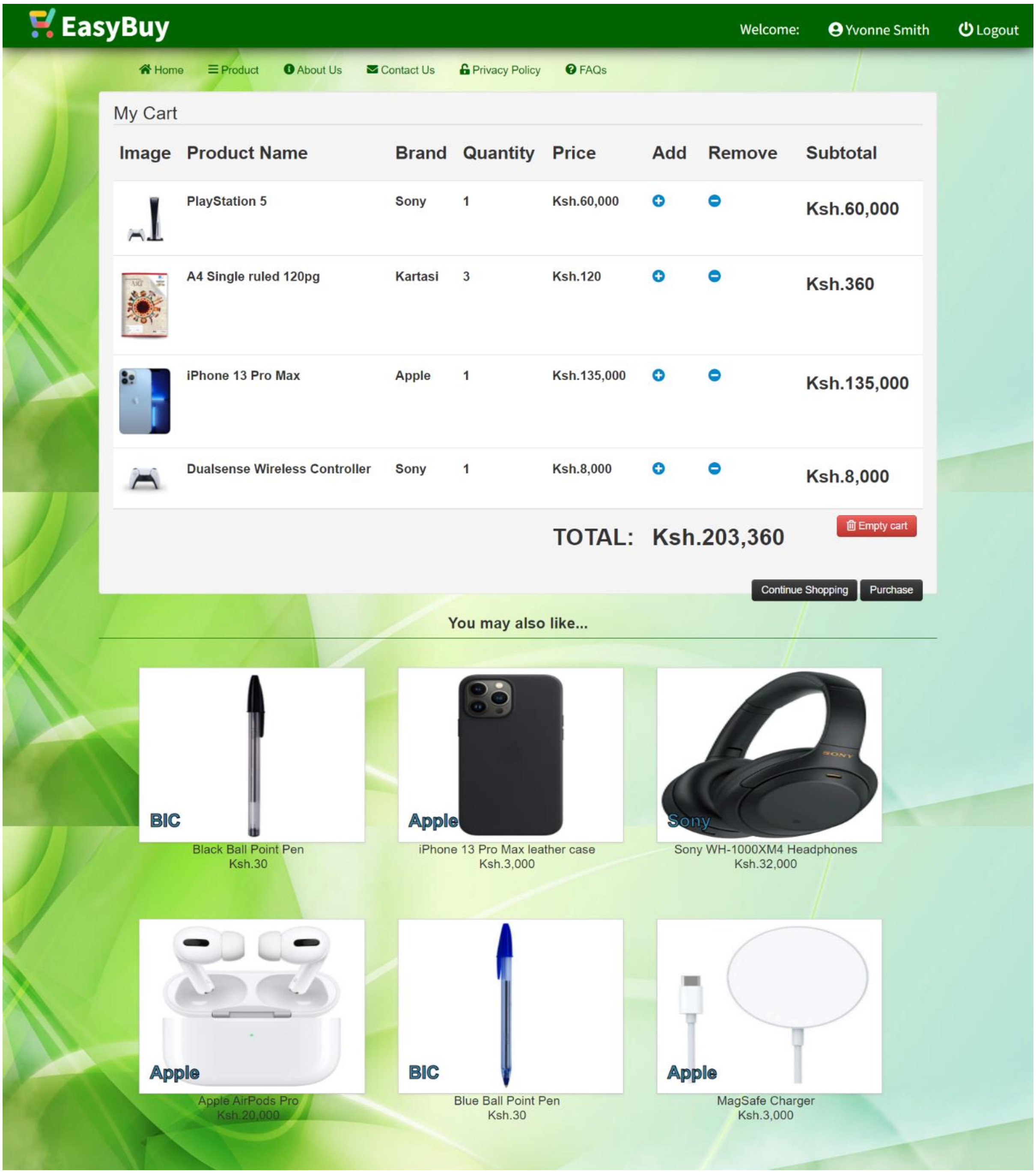



Figure E.17: Dualsense controller no longer in recommended list.


Figure E.18: Brand and product ranking after an order of 40 BIC black pens.

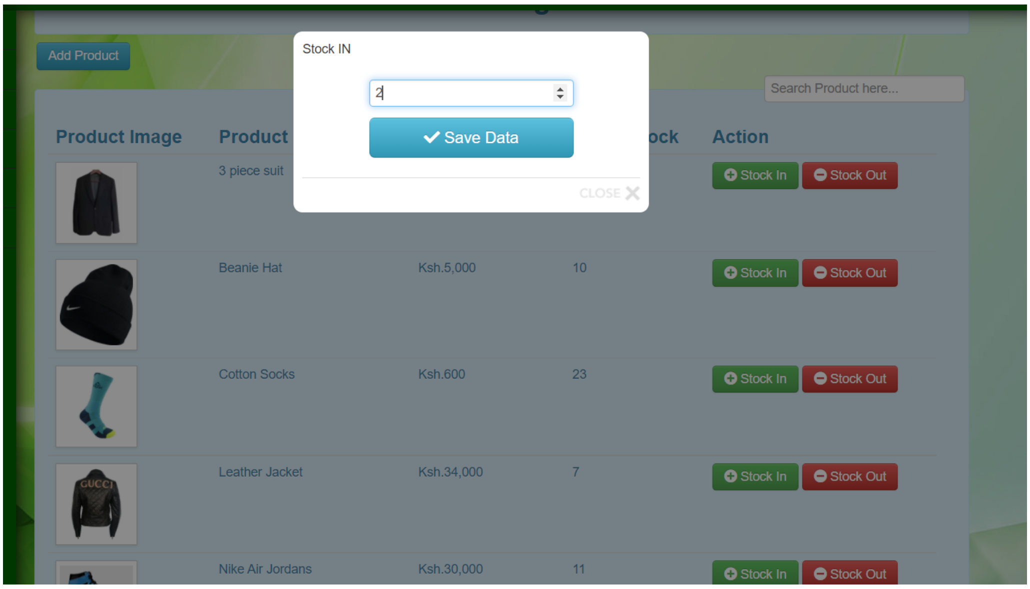



Figure E.19: Stocking in 2 leather jackets.

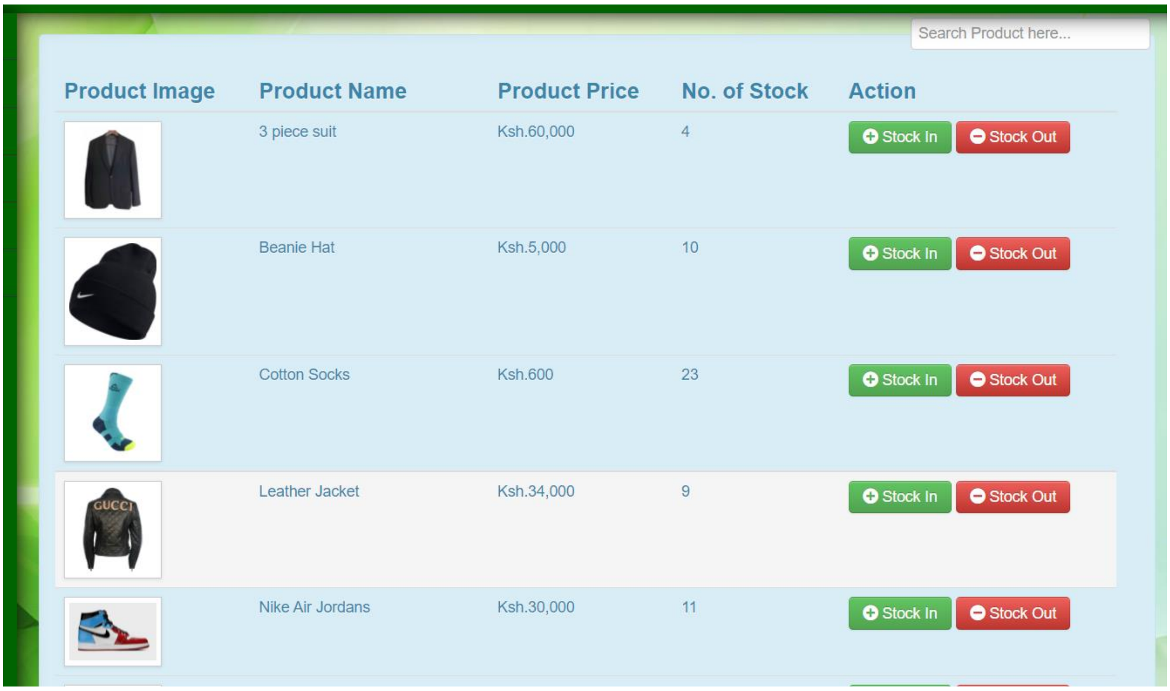



Figure E.20: No. of stock reflecting the stock in action.

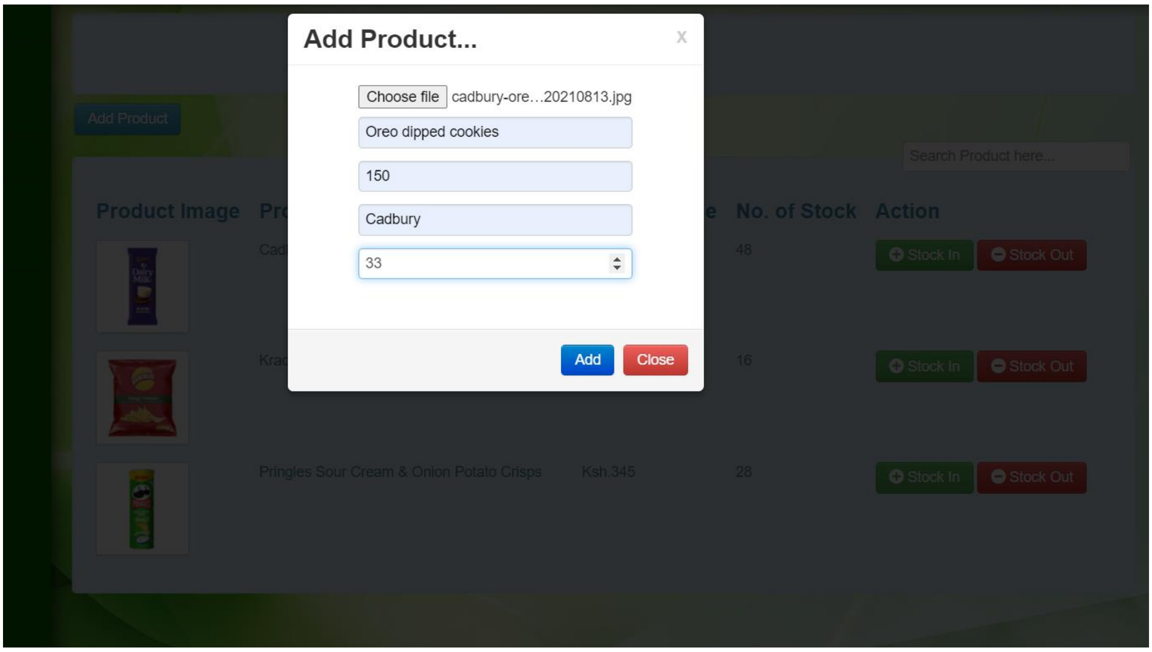



Figure E.21: Adding a new snack product.

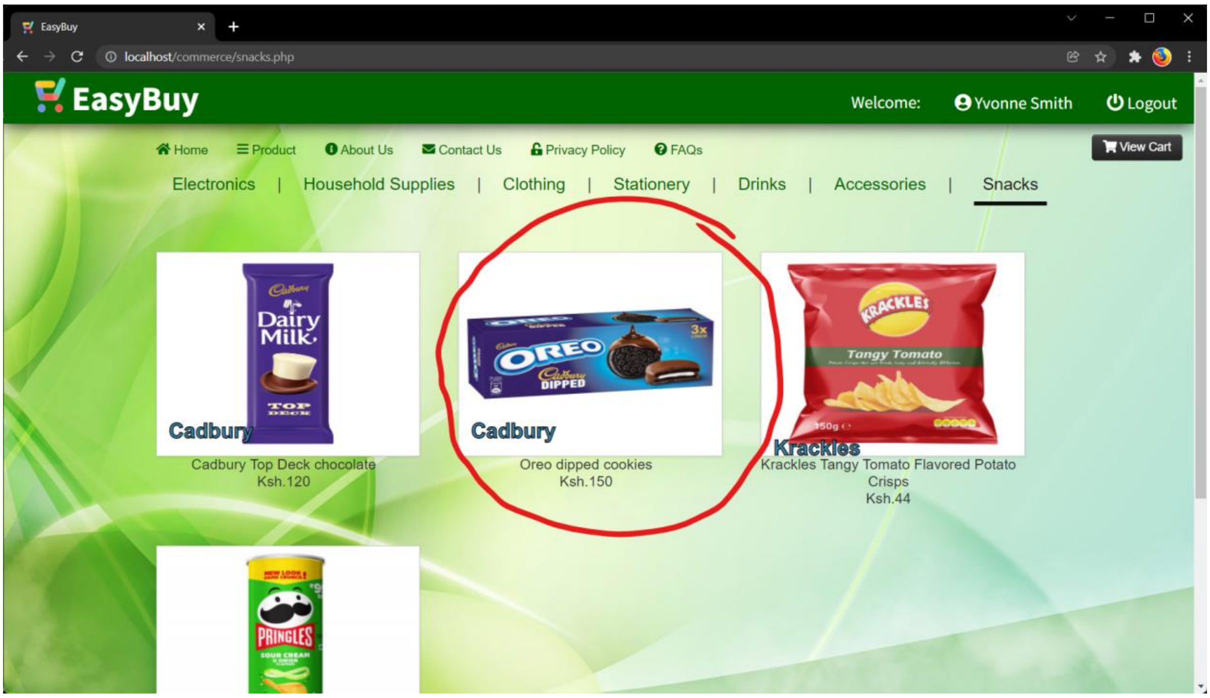



Figure E.22: Newly added product purchasable by customers.

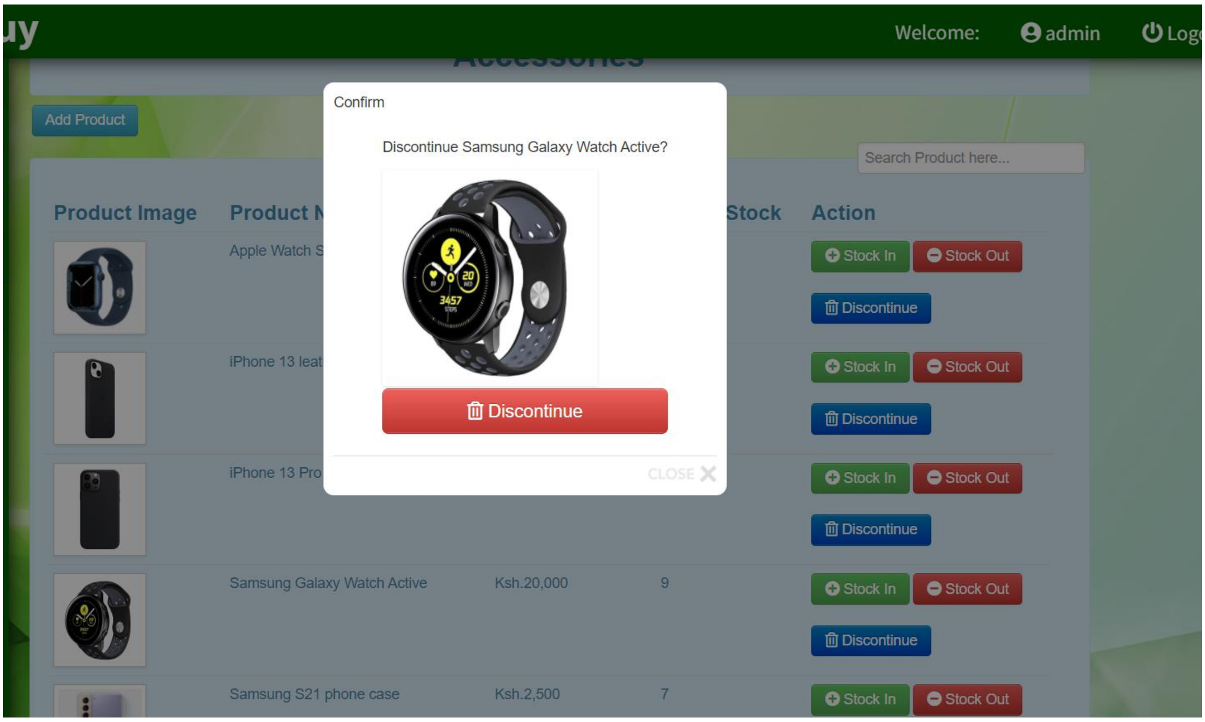



Figure E.23: Discontinue product window.

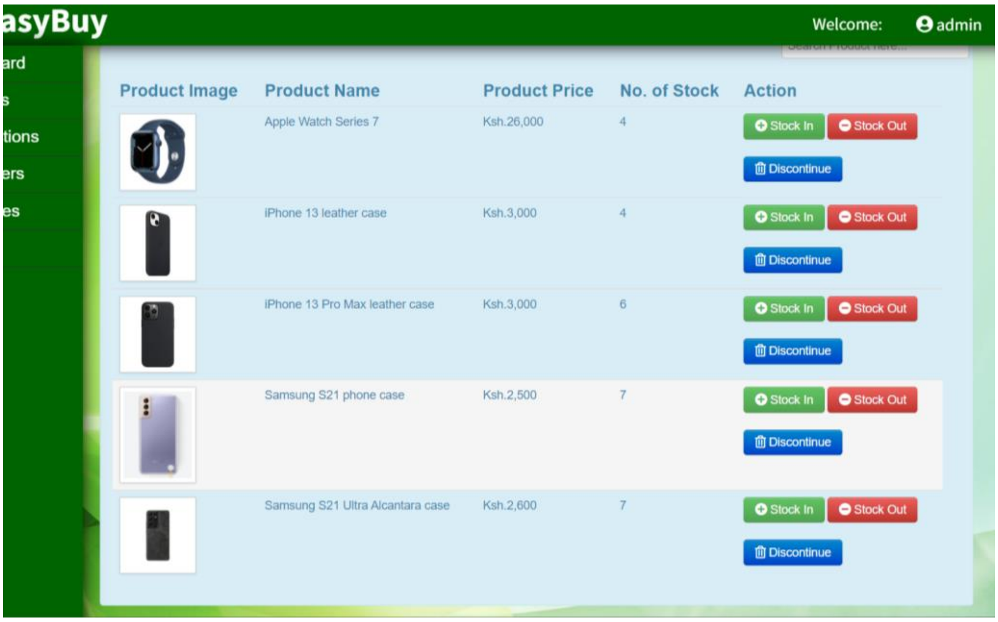



Figure E.24: Discontinued product not shown on inventory.

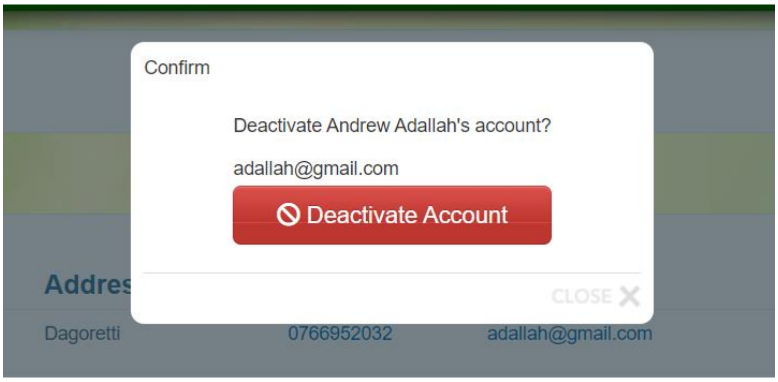



Figure E.25: Deactivating a customer account.

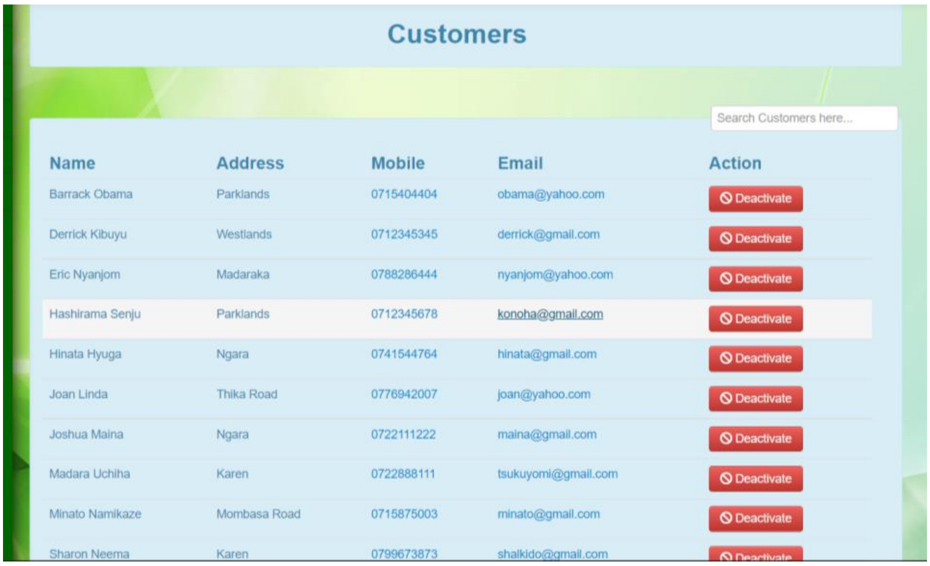



Figure E.26: Deactivated accounts not shown in customer list.

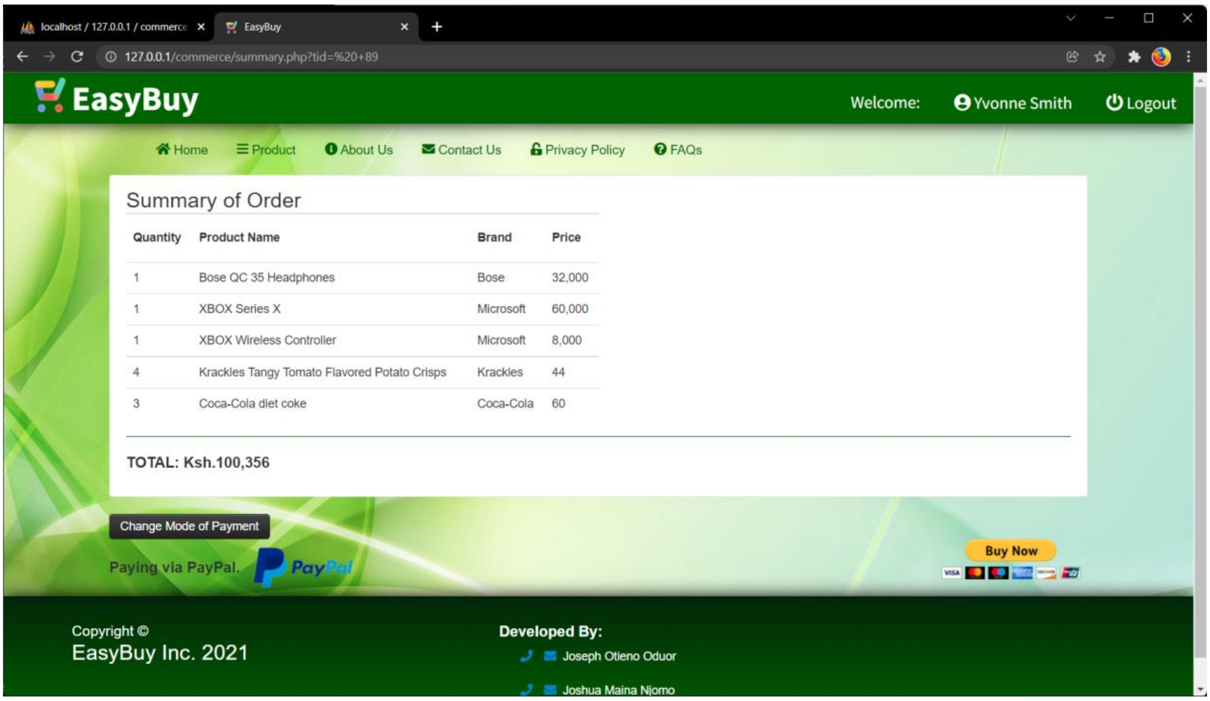



Figure E.27: Paying via PayPal.

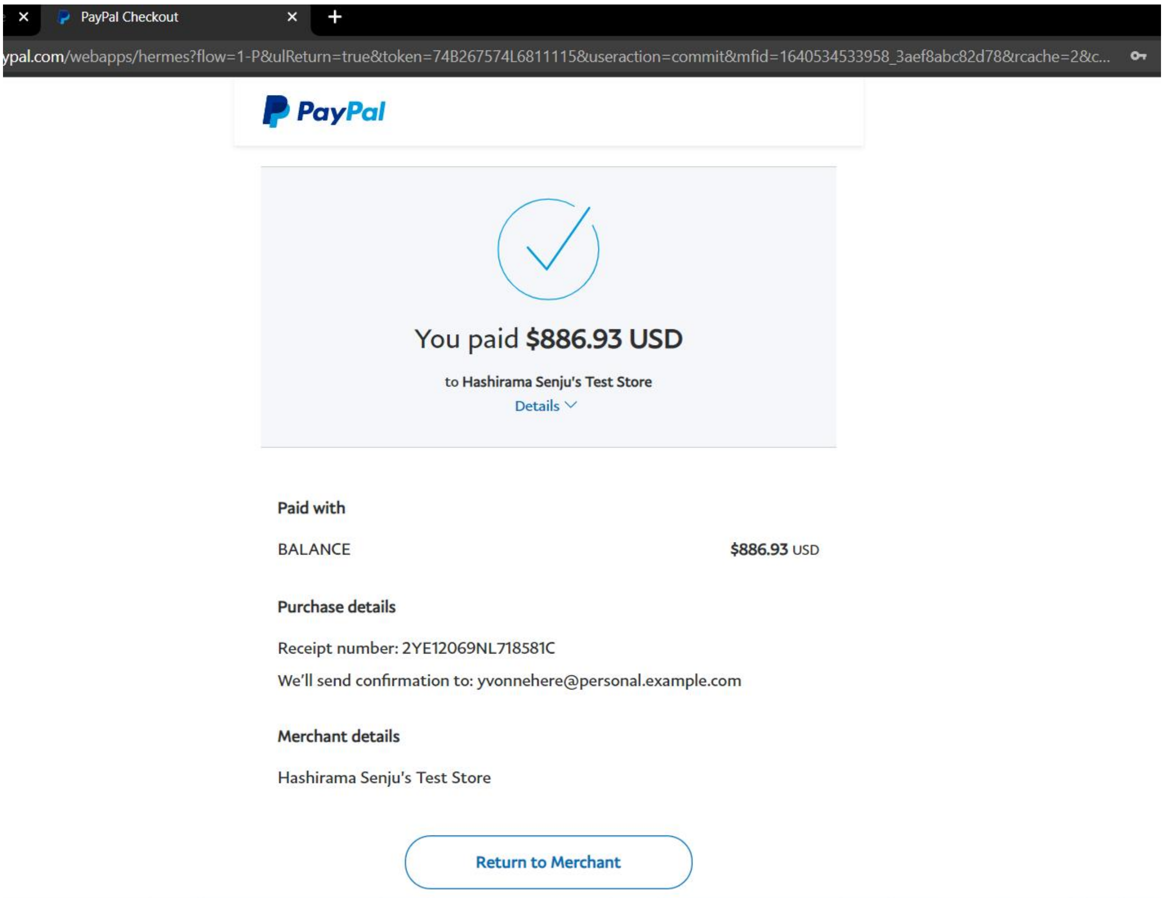



Figure E.28: PayPal sandbox API payment confirmation.

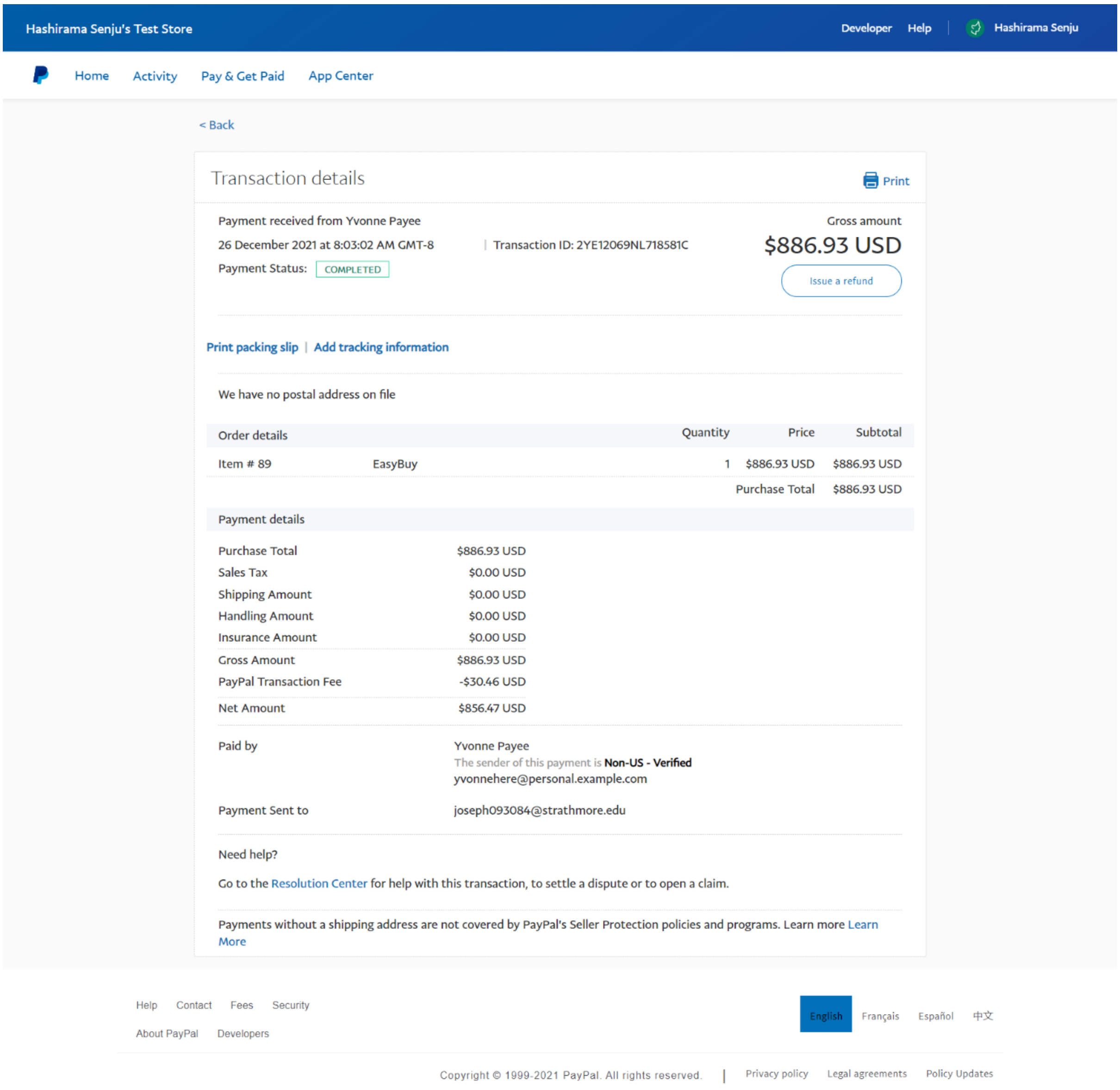



Figure E.29: Payment reflected in the PayPal business merchant account.

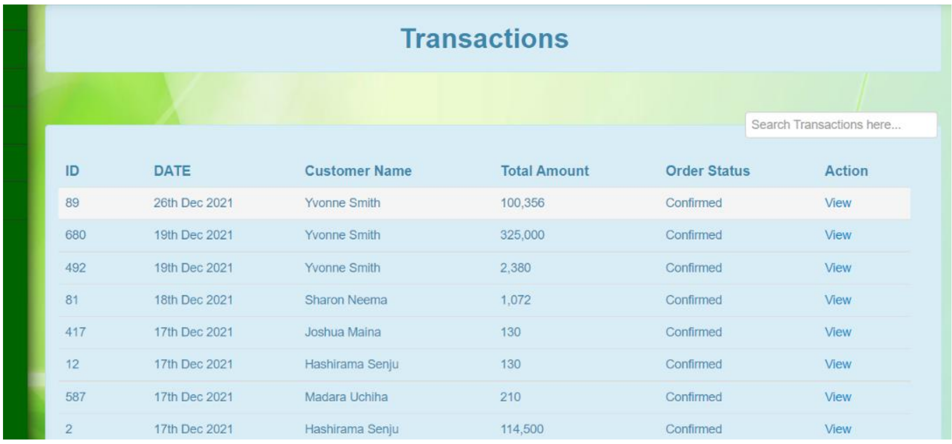



Figure E.30: Transaction auto-confirmed by the PayPal sandbox API.

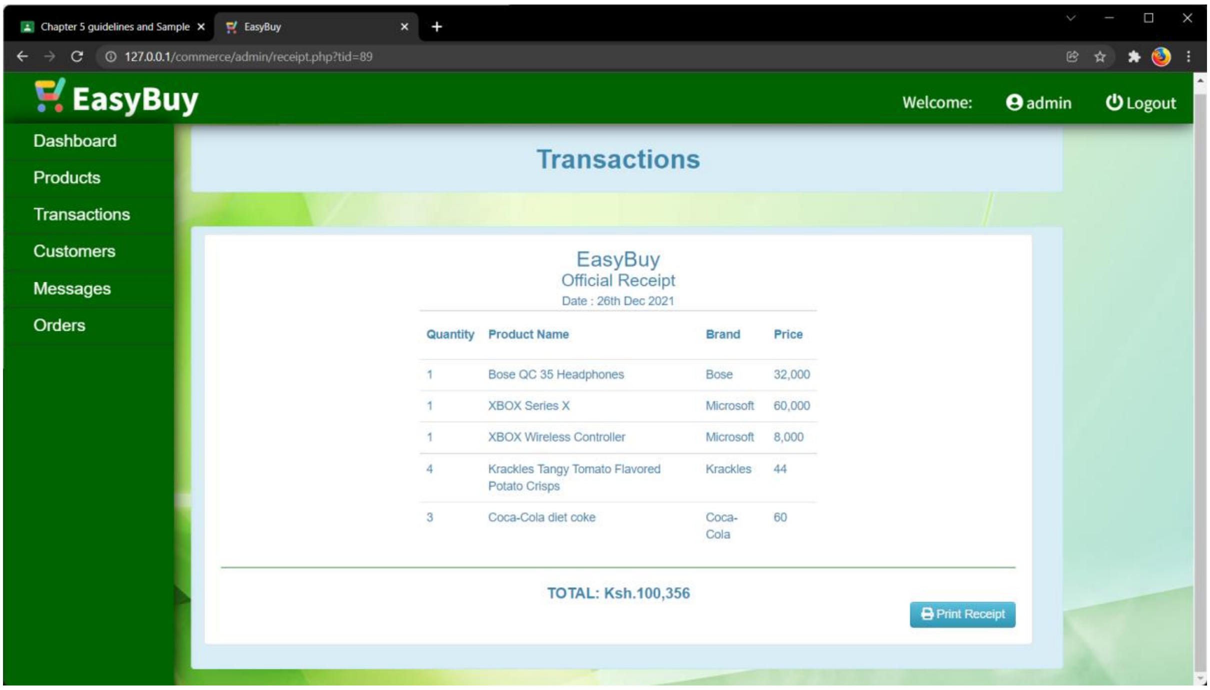



Figure E.31: Official transaction receipt.

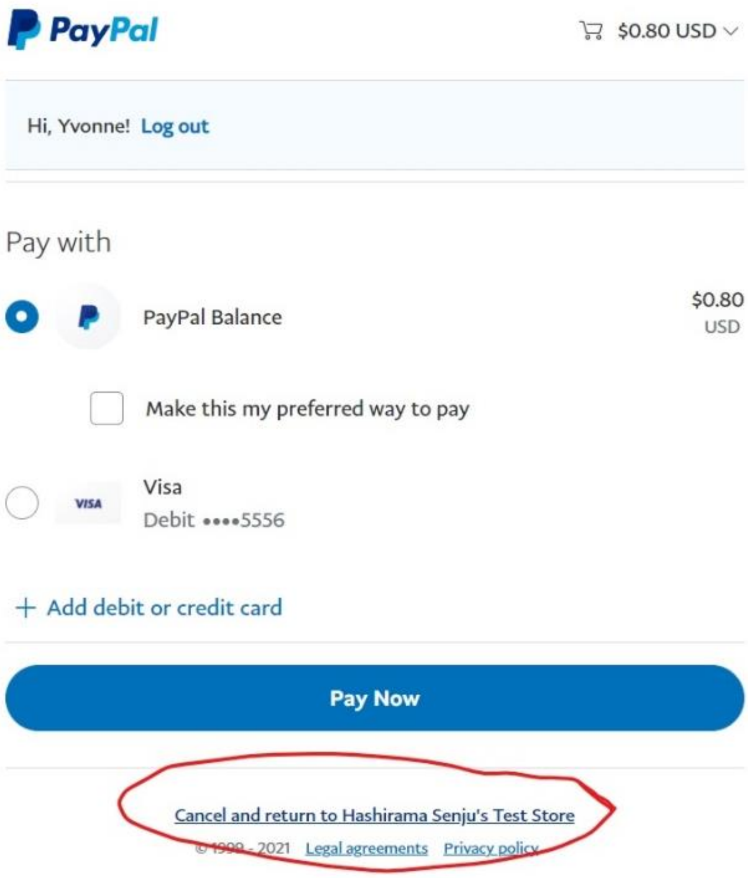



Figure E.32: Cancelling an order while in the payment module.


Figure E.33: Cancelled order in the administrator dashboard.

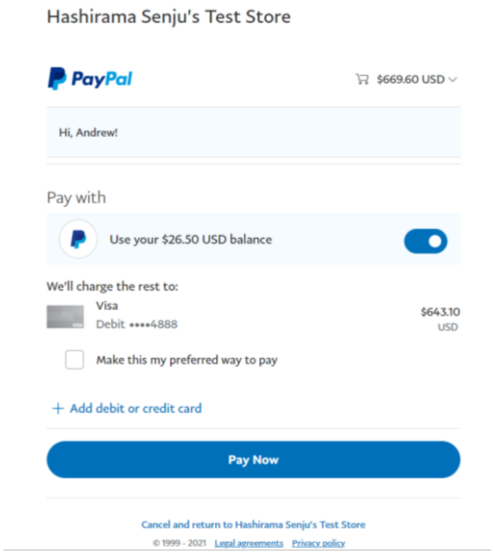



Figure E.34: Paying with a registered credit/debit card.

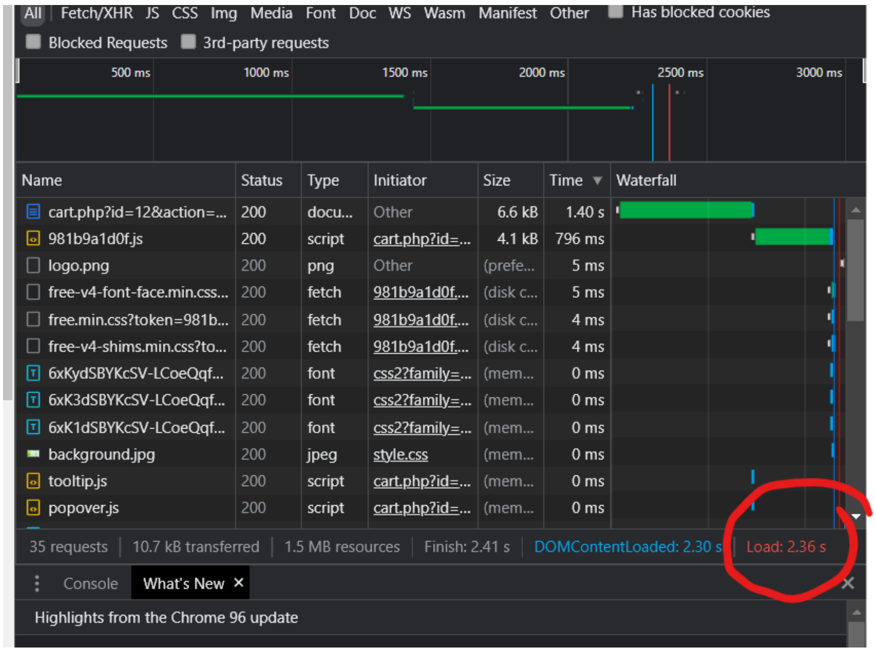



Figure E.35: Longest load times from sleep.

# DBIT IS PROJECT DOCUMENTATION MARKING GUIDE

| | |
|---|---|
| Name of Candidate: JOSEPH OTIENO ODUOR | Student Number: 093084 |
| Name of Candidate: JOSHUA MAINA NJOMO | Student Number: 138065 |
| Title of Project: A RECOMMENDER SYSTEM FOR A WEB BASED E-COMMERCE PLATFORM USING K-MEANS CLUSTERING | |

| No. | Areas of assessment | Potential score | Actual Score |
|---|---|---|---|
| **0** | **Preamble (5 Marks)** | | |
| | Title page: Informative, concise and appropriate | 2 | |
| | Abstract: Gives proper summary of the project including solution, methodology and expectations | 3 | |
| **1** | **Introduction: (8 Marks)** | | |
| | Background of the Study | 2 | |
| | Problem statement (Clear Gap Identified) | 3 | |
| | Objectives (SMART and linked to Problem statement) | 3 | |
| **2** | **Literature Review: (12 Marks)** | | |
| | Relevant literature (related to the project) was consulted | 3 | |
| | Literature cited properly within text and in APA style | 2 | |
| | Literature consulted is adequate for the research | 3 | |
| | Conceptual framework included and well discussed | 4 | |
| **3** | **Approach and Methodology (8 Marks)** | | |
| | The choice of methodology discussed | 2 | |
| | Clear discussion of functional and non-functional requirements | 2 | |
| | Clear and well planned approach to project work; Discussion of milestones and Deliverables | 2 | |
| | Clear explanation of the tools and techniques used | 2 | |
| **4** | **System Analysis and Design (10 Marks)** | | |
| | Description of system architecture | 2 | |
| | Analysis diagrams (use case, Sequence, ER model) | 4 | |
| | Design diagrams (Interface wireframes, DB schema) | 4 | |

| | | | |
|---|---|---|---|
| **5** | **System Implementation and Testing (12 marks)** | | |
| | Implementation of core modules | 5 | |
| | Test cases of the core modules | 5 | |
| | Report modules to inform decision making | 2 | |
| 6 | Conclusions and Recommendations (4 Marks) | | |
| | Recommendations and future works | 2 | |
| | Evidence the student has learned | 2 | |
| 7 | Documentation Structure (11 Marks) | | |
| | Well formatted Table of contents | 1 | |
| | References provided and formatted in APA style | 2 | |
| | Clear and proper use of language | 2 | |
| | Effective chapter structures and layout | 2 | |
| | Appendices / Supervisor Signing Sheet (At least 10 signatures) | 4 | |
| | Total Score | 70 | |

Comments

…………………………………………………………………………………………………………

…………………………………………………………………………………………………………

…………………………………………………………………………………………………………

…………………………………………………………………………………………………………

…………………………………………………………………………………………………………

…………………………………………………………………………………………………………

…………………………………………………………………………………………………………

…………………………………………………………………………………………………………

…………………………………………………………………………………………………………

Signed: ________________________________ Date: _______________

(Name of the Examiner)