

**Online Auctioning and Bidding System**

**EMMNAUEL CHESIRE**

**CS/M/0466/01/21**

**JEREMY WACHIRA**

**INTE/N/2268/09/20**

**MIRRIAM NEKESA**

**BBIT/MG/1627/09/19**

**A Research Project Submitted to the Institute of Undergraduate Studies and Research for Partial Fulfilment of the Requirements for the Award of Degree in School of Science and Engineering of Kabarak University.**

**DECEMBER 2022**

# **DECLARATION**

We declare that this work without any reasonable doubt has never been presented before to the Faculty of Information Technology or any other Institution. No part of this research document shall therefore be duplicated without prior consent.

Student: Emmanuel Kigen Chesire

Sign: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student: Jeremy Wachira

Sign: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student: Mirriam Nekesa

Sign: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Lecturer’s Approval**

Name: Mr. Chrispus Alukwe

Sign: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **ACKNOWLEDGEMENT**

We praise God for the gift of life, family, friends and provision without which we would not have made it this far in our academic pursuits. Indeed, the very ability to read and comprehend is a gift that only you, God give. All honor praise and glory to you.

Many special thanks to our lecturer and supervisor Mr. Chrispus Alukwe for the guidance and patience throughout our studies. Your wise counsel and input during our research have made this possible.

# **DEDICATION**

We dedicate this work to our parents and friends who gave us financial, psychological and moral support throughout our research and study at large.

# **ABSTRACT**

By definition, an auction is a transaction in which a good or service is put up for sale, the highest bidder is chosen, and the item is subsequently sold. Users can complete the aforementioned process with the use of the web-based Online auction system; they can trade anything they want. Users of this application will be able to list their goods for auction; interested parties can register and place bids on any item that is still on open bid.

The system we are developing allows buyers and sellers to transcend geographic restrictions and purchase products anytime and anywhere via the Internet. The online auction market provides significant benefits to consumers in terms of lower costs associated with physical auctions, greater product selection, and greater efficiency on multiple levels. The current related systems are insufficient due to the use of online market place systems that do not allow for the auctioning process and competitive trade. The system will be able to notify sellers of new bids placed in auctions in which they participate, as well as the winning bidder.

# **TABLE OF CONTENTS**

Contents

[**DECLARATION** 2](#_Toc120682340)

[**ACKNOWLEDGEMENT** 3](#_Toc120682341)

[**DEDICATION** 4](#_Toc120682342)

[**ABSTRACT** 5](#_Toc120682343)

[**TABLE OF CONTENTS** 6](#_Toc120682344)

[**LIST OF FIGURES** 8](#_Toc120682345)

[**LIST OF ABBREVIATIONS** 9](#_Toc120682346)

[**CHAPTER ONE INTRODUCTION** 10](#_Toc120682347)

[**1.0** **Introduction** 10](#_Toc120682348)

[**1.1** **Background** 11](#_Toc120682349)

[**1.2** **Problem Statement** 11](#_Toc120682350)

[**1.3** **Objectives** 12](#_Toc120682351)

[**1.3.1** **Main Objective** 12](#_Toc120682352)

[**1.3.2 Specific Objectives** 12](#_Toc120682353)

[**1.4** **Justification** 12](#_Toc120682354)

[**1.5** **Scope** 13](#_Toc120682355)

[**1.6** **Conclusion** 13](#_Toc120682356)

[**CHAPTER TWO LITERATURE REVIEW** 13](#_Toc120682357)

[**2.0 Introduction** 13](#_Toc120682358)

[**2.1 Online Auction Systems in Kenya** 14](#_Toc120682359)

[**2.3 Related Works** 14](#_Toc120682360)

[**2.3.2 Cheki.com** 15](#_Toc120682361)

[**2.3.3 Bidpesa** 15](#_Toc120682362)

[**2.3.4 jiji.co.ke** 15](#_Toc120682363)

[**2.4 Generals Gaps in The Existing System** 15](#_Toc120682364)

[**2.4.1 Scammers** 15](#_Toc120682365)

[**2.4.2 Hackers** 15](#_Toc120682366)

[**2.4.3 Paying systems** 16](#_Toc120682367)

[**2.4.4 Customer service** 16](#_Toc120682368)

[**2.5 Conceptual Framework** 16](#_Toc120682369)

[**CHAPTER THREE METHODOLOGY** 18](#_Toc120682370)

[**3.0 Introduction** 18](#_Toc120682371)

[**3.1 Research Methodologies** 18](#_Toc120682372)

[**3.2 Design Methodology** 19](#_Toc120682373)

[**3.2.1 Requirement Analysis** 19](#_Toc120682374)

[**3.2.2 Design** 19](#_Toc120682375)

[**3.2.3 Develop** 20](#_Toc120682376)

[**3.2.4 Testing** 20](#_Toc120682377)

[**3.2.5 Maintenance** 20](#_Toc120682378)

[**3.2.6 Feedback** 20](#_Toc120682379)

[**3.2.7 Tools and Techniques** 20](#_Toc120682380)

[**Python** 20](#_Toc120682381)

[**Django** 20](#_Toc120682382)

[**HTML** 21](#_Toc120682383)

[**CSS** 21](#_Toc120682384)

[**JavaScript** 21](#_Toc120682385)

[**3.3 Justification of This Methodology** 21](#_Toc120682386)

[**3.4 Data Collection Method** 21](#_Toc120682387)

[**3.4.1 Primary Data Collection Methods** 21](#_Toc120682388)

[**3.4.2 Secondary Data Collection Methods** 22](#_Toc120682389)

[**3.5 Data Flow Diagram** 22](#_Toc120682390)

[**CHAPTER FOUR: SYSTEM IMPLIMENTATION AND DEPLOYMENT** 23](#_Toc120682391)

[**4.0 Introduction** 23](#_Toc120682392)

[**4.1.1 Activity Diagram** 24](#_Toc120682393)

[**4.1.2 Use Case Diagram** 26](#_Toc120682394)

[**4.1.3 Data Flow Diagram** 28](#_Toc120682395)

[**4.1.3 Context Diagram** 29](#_Toc120682396)

[**4.2 Front End Development** 29](#_Toc120682397)

[**4.3 User Interface Designs and Test cases** 32](#_Toc120682398)

[**4.3.2 Reports Generation Test-Case** 36](#_Toc120682399)

[**4.4 Backend Development** 39](#_Toc120682400)

[4.4.1 Views.py file 40](#_Toc120682401)

[4.4.2 Urls.py file 40](#_Toc120682402)

[**4.5 Testing** 43](#_Toc120682403)

[**4.5.1 Unit Testing.** 43](#_Toc120682404)

[**4.5.2 Integration Testing** 43](#_Toc120682405)

[**4.5.3 System Testing** 44](#_Toc120682406)

[**4.5.4 Acceptance Testing** 44](#_Toc120682407)

[**4.6 Deployment Methods** 44](#_Toc120682408)

[**CHAPTER 5 CONCLUSION AND RECOMMENDATIONS** 46](#_Toc120682409)

[**5.0 Conclusion** 46](#_Toc120682410)

[**5.1 Recommendations** 47](#_Toc120682411)

[**5.2 Future Works** 47](#_Toc120682412)

[**APPENDIX** 48](#_Toc120682413)

[Time Tabling 48](#_Toc120682414)

[**Budget** 48](#_Toc120682415)

[**REFERENCES** 49](#_Toc120682416)

# **LIST OF FIGURES**

**No table of figures entries found.**

# **LIST OF ABBREVIATIONS**

CSS – Cascading Style Sheets

DBMS – Database Management System

HTML – Hyper Text Markup Language

IT – Information Technology

MTBF- Mean Time Between Failure

PC – Personal Computer

SDLC – System Development Life Cycle

SMS – Short Message Service

# **CHAPTER ONE INTRODUCTION**

## **Introduction**

To enable buyers and sellers overcome geographical constraints and to purchase products from anywhere at any time via the internet, the online bidding and auctioning system we intend to create will help. Compared to traditional offline markets, the online auction market offers significant advantages to consumers in terms of lower costs related to physical auctions, greater product selection, and greater efficiency on many levels.

Using product and sellers’ information we describe the product through the use of textual and visual elements, a description of the product's characteristics and a primary book value

## **Background**

Auctions are usually held as special events with the presence of a reception and the auctioneer delegates the bids. This method necessitated the bidder's physical presence, which had several drawbacks. This resulted in the use of online bidding and auctioning, which allows auctions to be conducted via the internet. The majority of current online market place systems don’t incorporate the auctioning process. Everything (products and the state of authenticity of the sellers) is legitimate and confirmed by the administrator before any of the products are put up for bidding and auctioning. Necessary documents that will be proof-checked will be used to submit a bid and only after being approved by the administrator will they proceed to the auctioning process.

This project shall handle this issue by creating an online platform where a user will be able to post items online for auction. The items will accompany item name, selling price, and a picture presentation for the bidder to see (Dutta, and Ramamoorthy, 2009). The bidder, if interested in the item, will auction for the product and will be able to inspect the item physically to approve the product then complete the business with the seller.

The system is a forum where the bidders meet their respective product sellers in their locality on the internet to solve the problem of item auctioning.

## **Problem Statement**

Currently the bidding and auctioning of things mostly happens in person. Hence it would demand potential participants to travel lengthened distances just to engage in a bidding or auctioning process. Time is emaciated along the way and this is inconvenient to most people.

On the other hand, available bidding systems are not reliable due to the disparity of the buyers, cone-men have always taken the advantage to offer item delivery to the customers. Many fake items have found their way into the hands of the people, or buyers remain in the same condition of lack, as they don’t get the right items from the sellers. Sometimes buyers struggle to find the right items, in failure, and they seek to get back to their homes. On the other hand, we have suppliers and business people who are qualified to supply and sell the items yet they have very few people who can come to them.

## **Objectives**

### **Main Objective**

To put in place an online bidding and auctioning system for buyers and sellers (using a web-based application engineered with Django).

### **1.3.2 Specific Objectives**

1. Design and develop a system that will ensure the trust of both the bidder and seller. The information of the buyer, seller and the products is well vetted.
2. Notifying bidders and sellers of new bids made in auctions in which they participate.
3. Potentially enable the setting of longer lasting auctions that would conveniently make it easier for the schedules of bidders to participate in the auction.

## **Justification**

The use of online market place systems that do not allow for the auctioning process and competitive trade is why the current related systems are insufficient.

The goal of this new system is to enable an online and transparent bidding and auctioning system that has both buyer and seller’s trust in their identities, products and process for which they choose to make bid and auction. Using visual and textual product descriptions, third-party product certification and product base value. The successful implementation of this project results in a functional much needed online bidding and auctioning system.

## **Scope**

This online market place system only supports the sale of products with only one set market price that will be final. This system accepts more bids for a product increasingly setting new minimum bidding prices. The auctioning process is exclusively open to registered potential buyers and sellers. The system allows for viewing of the auctioning process with activities of previous bidders and their bids on a product since the set starting price. It makes the process feel inclusive for the buyers and sellers.

## **Conclusion**

Following the identification of the project background, problem statement, objectives, scope and justification it is revealed that the successful implementation of this online auction system will result in a more effective way of conducting auctions.

# **CHAPTER TWO LITERATURE REVIEW**

## **2.0 Introduction**

This chapter looks at the current available online systems and state their strengths and weaknesses. The purpose of this chapter is to analyze the evolution of online auctioning systems over the years.

This chapter will look at the available systems and show our new system is different from the rest of the systems and how it is of importance the development of this new system. The new system will address the weaknesses in the current available systems thus resulting in development of this new effective system.

## **2.1 Online Auction Systems in Kenya**

Electronic markets especially those based on the internet have become platforms for providing environment to conduct business transactions. This represents the change of business sector to ecommerce over the years.  The users for online auction management web site will be bidder, seller, or admin (Konia, 2007). Auctioning systems are some of the major components in enabling users to buy and sell products online. The sellers set up an auction for the different products while the purchaser who bids the highest amount wins the right to purchase that product.

There are different types of auctioning systems classified on how the handles the requests. The first type provides the function of registering goods for an auction to the sellers *Zhang, Yang*, *IEEE Communications surveys & tutorials* 15.3 (2012): 1020-1041. This design maximizes the probability that the product auctioned sells. The second type analyzes the bidding price of another competitor and suggests the best price to the user. The final type, the system plays the role of the auctioneer an enables the client to look at the other persons history of auction while bidding for the product.

Clients access the auction system from one of these computers. The system allows the clients to buy and sell items by means of auctions.

## **2.3 Related Works**

This section of this chapter takes a step further in investigating the current existing auction systems and provides different descriptions of the technology used. Before starting with the project, considerations of current existing system and their functionalities are taken into account, then decide on the kind of system architecture and software technology to use.

The first part of the project is an investigation of already existing on-line auction systems around the internet. Three famous auction systems considered for this project are eBay, cheki.com and bidpesa.com

### **2.3.2 Cheki.com**

Established in 2010, Cheki.co.ke is Kenyan Auto website with more car buyers and car sellers than any other site in Kenya.Cheki.co.ke is an online marketplace where all of Kenya's leading car dealers, importers and private sellers post their cars for sale in Kenya. We do not sell cars directly but rather offer a state of the advertising medium for Kenyans to buy and sell their cars, 4WDs, vans, bikes and trucks and other vehicles. (www.cheki.com)

### **2.3.3 Bidpesa**

The latest e-commerce platform in Kenya is Bidpesa that uses bidding as an approach to sell its items. It provides free delivery the following day after your purchase. Users are required to place a bid on the advertised product, consequently raising the auction price by Ksh. 1 for each bid made. (www.bidpesa.com)

### **2.3.4 jiji.co.ke**

This is one of the most famous online selling platforms in Kenya. This site is not actually a bidding site but incorporates some of its capabilities. It allows an auctioneer to set up an account and upload their item which will be viewed by registered users. Bidders interested on the product will contact the auctioneer to finalize their purchase. The auctioned chooses the buyer based on their set conditions but not necessarily according to their bid price. (www.jiji.co.ke)

## **2.4 Generals Gaps in The Existing System**

### **2.4.1 Scammers**

Scamming has been one of the major issues affecting online auctioning. Scammers can take a very long time to be not notice which leads to loss of millions of shillings by buyers.

### **2.4.2 Hackers**

There has been no way to contain hackers who get away with user accounts. Buyers can be scheming, Phishing schemes are also frequent on online users receive false emails stating that their eBay account will be suspended, directing them to a fake login page. Once the user logs in, his login data and eBay identity are susceptible to theft.

### **2.4.3 Paying systems**

There has been no method you used to come up with one common paying system. Some buyers and sellers disagree on which method to use for payments. There are a lot factors affecting the mode of payment that will be used by used, such as the cost of each transaction.

### **2.4.4 Customer service**

The users who are dissatisfied with sites undesirable policies or with any problems they have encountered cannot forward their opinions though this medium. This results in lack of communication and users receive automated email responses because a direct communication link is not established.

# **2.5 Conceptual Framework**

The figure below illustrated the functionality of the real time auction system. The diagram was an overview of how the system operated from when a user searches for a product in the system. The first step showed the input process where the user gave information to the system to be processed. The user entered the details of the product they wish to purchase mainly the name of the product. The system analyzed the data through a process of determining the availability of the product and based on the available ratings. The last thing that took place was the output of the system which specified the exact dispatch for an ambulance and estimating the arrival time to the destination. Lastly a report on the emergency was done.

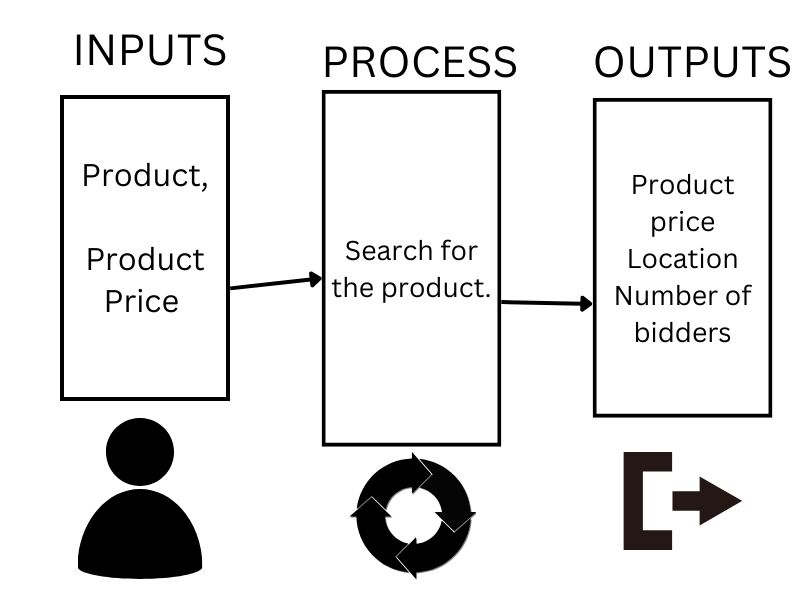
****

Figure ‑ Conceptual Framework

# **CHAPTER THREE METHODOLOGY**

## **3.0 Introduction**

The methodology used in development of the system was captured in this chapter. The functional and non-functional requirements have been illustrated in order to understand the development of the system.

## **3.1 Research Methodologies**

Research methodology is a collective term for the structured process of conducting research. There are many different methodologies used in various types of research and the term is usually considered for data gathering and data analysis. Part of the research methodology is concerned with how the research is conducted and this is called the study design and typically involves research conducted using questionnaires, interviews, observations and experiments. The term research methodology, is also referred to as research methods, it usually encompasses the procedures followed to Analyze and interpret the data gathered. These often use a range of sophisticated statistical analyses of the data to identify correlations. In this project we carried out different research methodologies that best suite our system.

## **3.2 Design Methodology**

The Agile methodology provided a modified approach where it assumes that requirements and solutions evolve through collaboration between self-organizing cross-functional teams and their end user.



Figure 3.1 : Agile Methodology

### **3.2.1 Requirement Analysis**

The system requirement for the online auctioning system will require different things for it to work efficiently. This requirement entails user requirements who is required to list their items for auctioning or bid for items. Other functional requirements include registration of users and authentication. Non-functional requirements include security, availability, accountability and portability.

### **3.2.2 Design**

A design was a plan or specification for the construction and building of a project. In this case of our system, the problem was thoroughly analyzed and a logical design for this is set up. The details needed for the design were collected and a blueprint of the system can be made. The system was designed in a way that was efficient and favored the users of the system.

### **3.2.3 Develop**

After the design of the system were put into place, developing the system concurrently began. Developing involved the hands-on work which was coding and building up the system.

### **3.2.4 Testing**

The implementation of the system was tested. It happened once the system was completed and ready for implementation. Testing was majorly done to ascertain whether the user requirements were met, this included both the functional and non-functional requirements.

### **3.2.5 Maintenance**

Maintenance involves the process of checking whether the system runs properly and also checking on what needs to be added to the already existing system.

### **3.2.6 Feedback**

This phase provided information about a product and was more of a rating on how the system fairs in the market that it operates in.

## **3.2.7 Tools and Techniques**

### **Python**

Python has become one of the most popular programming languages in the world in recent years. It's used in everything from machine learning to building websites and software testing. It was used for backed development.

### **Django**

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

### **HTML**

HTML (Hypertext Markup Language) is the code that is used to structure a web page and its content. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables.

### **CSS**

CSS in full is cascading style sheets and it was a language used to describe how a document was going to be presented. It helped in formatting and displaying a good interface for the AADS system.

### **JavaScript**

JavaScript is a lightweight programming language that web developers commonly use to create more dynamic interactions when developing web pages, applications, servers, and or even games.

## **3.3 Justification of This Methodology**

It has been implemented because clients can make minor objective changes without huge amendments to the schedule. It aided the dynamic characteristics of business processes particularly the consumer wants

## **3.4 Data Collection Method**

### **3.4.1 Primary Data Collection Methods**

Proper choice of data collection methods is key. Primary data collection methods will consist of data collected for the first time. For this study, closed-ended questionnaire and one-on-one interviews will be utilized in collection of information. To begin with, two different questionnaires will be written up where one is for auctioneers while the other is for buyers. This will allow for the collection of quantitative data. Furthermore, one-on-one interviews will be used in order to identify the extent to which the problem being solved by this solution affects both auctioneers and buyers. The data collected through these means will be quantitative hence, easy to Analyse for trends hence informing the development of the system.

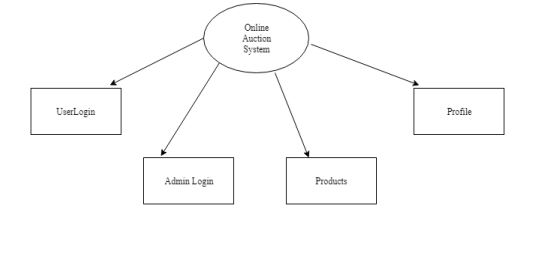
### **3.4.2 Secondary Data Collection Methods**

Secondary data will consist of information that has already been collected. For the purpose of this study, data was collected from various websites and online articles that highlight the problem of auctioning facing different sectors.

## **3.5** **Data Flow Diagram**

A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various subprocesses the data moves.

## 



# **CHAPTER FOUR: SYSTEM IMPLIMENTATION AND DEPLOYMENT**

**4.0 Introduction**Front End implementation - It was implemented using Hypertext Mark-Up Language, Cascading Style Sheets and JavaScript.  
Back End implementation - The Back-End database was developed using Python Django for framework.  
Implementation Strategy- We used the phased changeover strategy which took place in stages, where we implemented a part of the system done and ensured that it is probably working before going to the next. The tools which were used for development included sublime text for coding the system, the browser for running it and visual paradigm for  
designing the UML diagrams.  
In this stage, the agile model proved its importance due to its ability of allowing addition of new requirements as they were recognized at every stage of development. The model also helped me reduce inherent project risks by breaking a project into smaller segments and providing more ease-of-change during the development process. We involved the users as well as my supervisor throughout the process which helped increase the likelihood of user acceptance of the final implementation product

**4.1 System Architecture**

**4.1.1 Activity Diagram**Figure 4.1.1: A bidder’s Activity diagram

Show collection of items

Login

Activate Account/Deposit

Select a desired Item

Bid on item

Show Item chosen

No

Yes

If the auction closes with your bid as the highest

Admin contacts both the seller and bidder to organize the transactions and transactions are made

Process Updates and Update Status

Yes

No

Figure 4.1.2: A Seller’s Activity Diagram

Show collection of items

Login

Contact the admin with necessary documents and details relevant to the auctioning of his item

Is the seller’s and his/her item authenticity verified

Updating of a product and it’s status on the website

If the auction closes with a bid

Admin contacts both the seller and bidder to organize the transactions and transactions are made

Process Updates and Update Status

Yes

Yes

Activity diagrams are UML diagrams that describe the dynamic aspects of a system under development.

This is just a flow chart showing the flow from one activity to another. Activity in this case is a representation of the operation of the system.

## **4.1.2 Use Case Diagram**

<<uses>>

<<uses>>

<<uses>>

<<uses>>

<<uses>>

<<uses>>

User

Figure 4.2.1: A user USE CASE diagram

<<uses>>

<<uses>>

<<uses>>

<<extends>>

<<extends>>

<<extends>>

<<extends>>

<<extends>>

<<extends>>

<<extends>>

<<extends>>

Administrator

Figure 4.2.2: An Administrator USE CASE diagram

Use case diagram is the basic form of software requirements for a new software program under development. They represent the expected behavior of a system.

It’s used to design the system from the end user's point of view. This is a summary of some relationships

### **4.1.3 Data Flow Diagram**

A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various subprocesses the data moves.

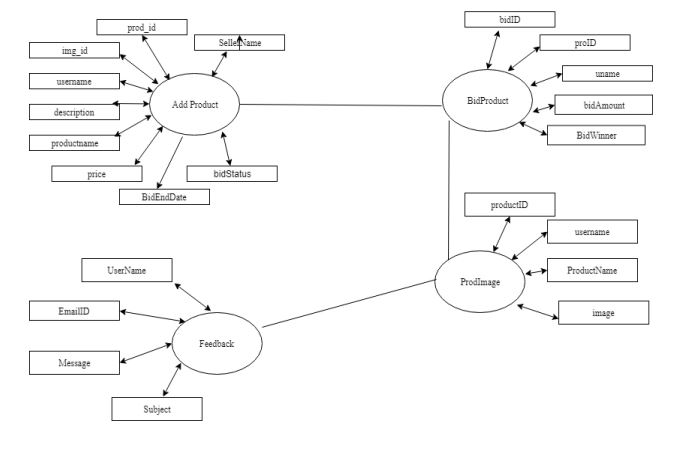


Figure 4.3: DFD Diagram

### **4.1.3 Context Diagram**

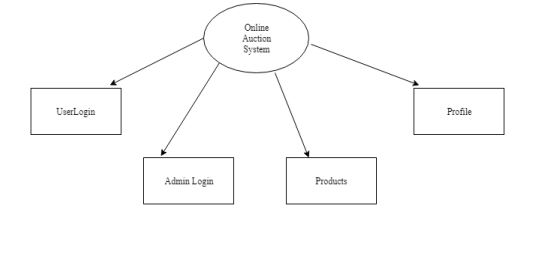
A context diagram is the first step in the process of creating data flow diagrams. DFDs illustrate how data moves through an information system. They, however, do not show programming logic or a program’s processing steps. Therefore, a context diagram is a simplified representation of an informationDsystem. 

Figure 4.5: Context Diagram

## **4.2 Front End Development**

**Example of code snippets used to develop the front end**

Below is the html code for the navigation bar for accounts

<div class="container">

    <nav  class="navbar navbar-default">

      <div class="container-fluid">

        <div class="navbar-header">

          <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar">

            <span class="icon-bar"></span>

            <span class="icon-bar"></span>

            <span class="icon-bar"></span>

          </button>

          <a style="color:black" class="navbar-brand" href="{% url 'catalog:product\_list' %}">BIDSHOP</a>

        </div>

        <div class="collapse navbar-collapse" id="myNavbar">

          <ul class="nav navbar-nav">

            <li><a href="{% url 'catalog:product\_list' %}">{% trans 'Home' %}</a></li>

              {% if request.user.is\_authenticated %}

              <li><a href="{% url 'account:dashboard' %}">{% trans 'My dashboard' %}</a></li>

              {% endif %}

          </ul>

          <ul class="nav navbar-nav navbar-right">

                  {%if request.user.is\_authenticated %}

                  <li>

                      <a>

                          <span style="color:#2770cf">Hello {{request.user.username}},</span>

                      </a>

                  </li>

               <li>

                   <a href="{% url 'logout' %}">

                      <span class="glyphicon glyphicon-log-out"></span>

                       {% trans 'Logout' %}

                   </a>

               </li>

              {% else %}

                <li><a href="{% url 'account:register' %}"><span class="glyphicon glyphicon-user"></span>{% trans 'Sign Up' %}</a></li>

                <li><a href="{% url 'login' %}"><span class="glyphicon glyphicon-log-in"></span>  {% trans 'Log-in' %}</a></li>

              {% endif %}

          </ul>

        </div>

      </div>

    </nav>

</div>

<body>

Below is the html code for the navigation bar for the catalog section

<div class="container">

    <nav  class="navbar navbar-default">

      <div class="container-fluid">

        <div class="navbar-header">

          <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar">

            <span class="icon-bar"></span>

            <span class="icon-bar"></span>

            <span class="icon-bar"></span>

          </button>

          <a style="color:black" class="navbar-brand" href="{% url 'catalog:product\_list' %}">BIDSHOP</a>

        </div>

        <div class="collapse navbar-collapse" id="myNavbar">

          <ul class="nav navbar-nav">

            <li><a href="{% url 'catalog:product\_list' %}">{% trans 'Home' %}</a></li>

             {% if request.user.is\_authenticated %}

            <li><a href="{% url 'account:dashboard' %}">{% trans 'My dashboard' %}</a></li>

              {% endif %}

          </ul>

          <ul class="nav navbar-nav navbar-right">

              {% get\_current\_language as LANGUAGE\_CODE %}

              {% get\_available\_languages as LANGUAGES %}

              {% get\_language\_info\_list for LANGUAGES as languages %}

              {%if request.user.is\_authenticated %}

              <li>

                  <a>

                      <span style="color:#2770cf">Hello {{request.user.username}},</span>

                  </a>

              </li>

               <li>

                    <a href="{% url 'logout' %}">

                      <span class="glyphicon glyphicon-log-out"></span>

                      {% trans 'Logout' %}

                   </a>

               </li>

              {% else %}

                <li><a href="{% url 'account:register' %}"><span class="glyphicon glyphicon-user"></span> {% trans 'Sign Up' %}</a></li>

                <li><a href="{% url 'login' %}"><span class="glyphicon glyphicon-log-in"></span> {% trans 'Log-in' %}</a></li>

              {% endif %}

          </ul>

        </div>

      </div>

    </nav>

  </nav>

{% %} is used to write python code within the html code

As seen below (in the code for payments), extends tag is used for inheritance of templates in Django. One needs to repeat the same code again and again. Using extends we can inherit templates as well as variables. {% block %} The block tag is used to define a block that can be overridden by child templates. In other words, when you define a block in the base template, you're saying that this area will be populated with content from a different, child template file. Consider this base.

{% extends "catalog/base.html" %}

{% block title %}Top up your account{% endblock %}

**4.3 User Interface Designs and Test cases**  
User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions.

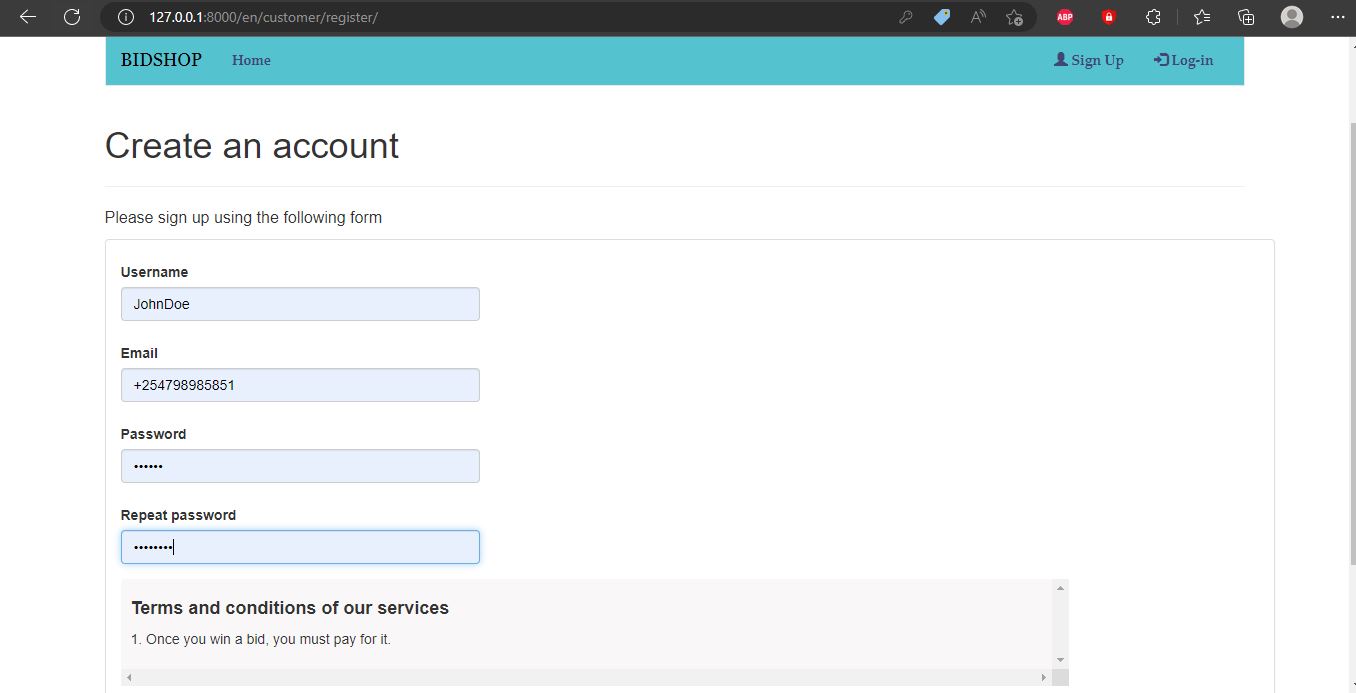
TEST CASE 1**4.3.1 Registration and login test case**The system allows registration of users by creating an account, username and passwords which will be used during the log in process.  
1. Test procedure - Enter the username and password then click the sign in button.  
2. Test data - phone number/ registration number, password.  
3. Expected results - The user logs in on entering correct username and password.  
4. Actual results - If the user details are correct, the user logs in, otherwise he/she is denied access to the system during login.

Figure 4.4.1

**Browse**The system allows the user to browse products that are available for auction, it displays the items on the dashboard and allows to bid on the desired item.  
1. Test procedure – Go to websites dashboard and browse available items.  
2. Test data – products available for auctioning.  
3. Expected results - The system displays the result of the search and allows the user to bid on the desired product.  
4. Actual results –the user gets to bid on the desired products.

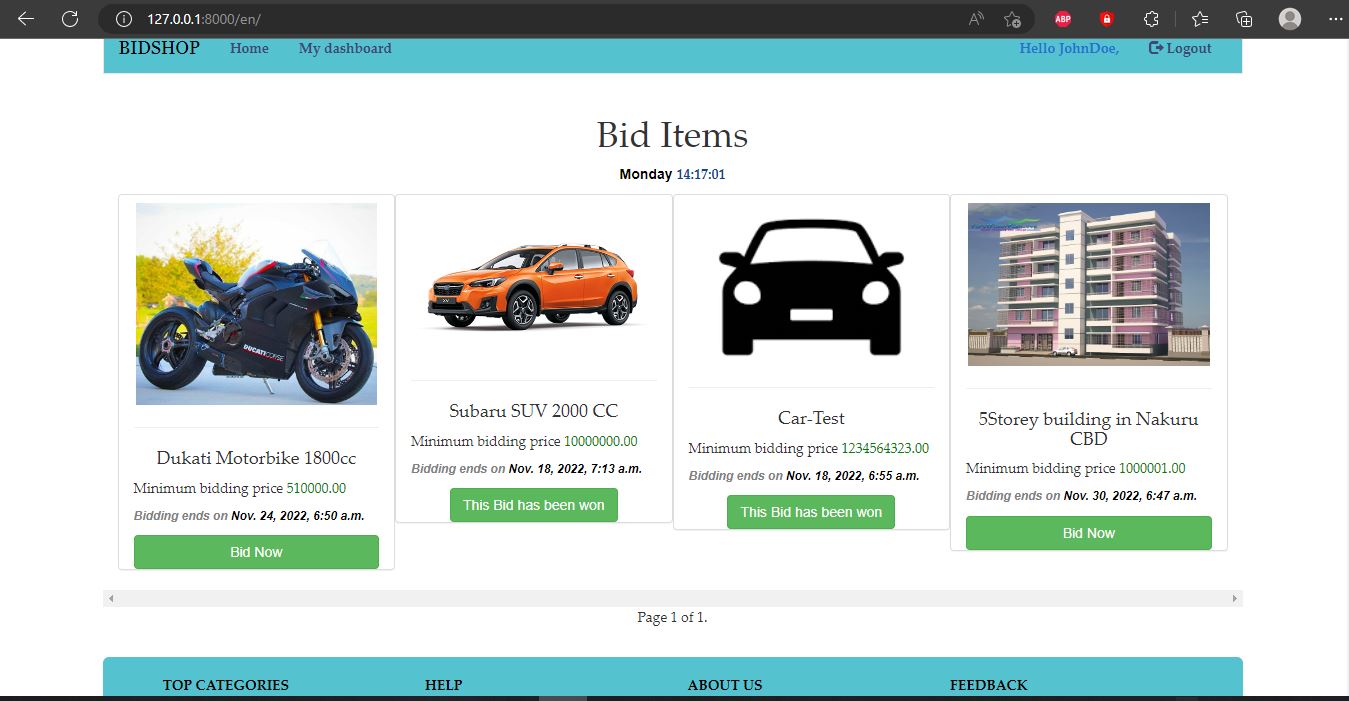


Figure 4.4.2

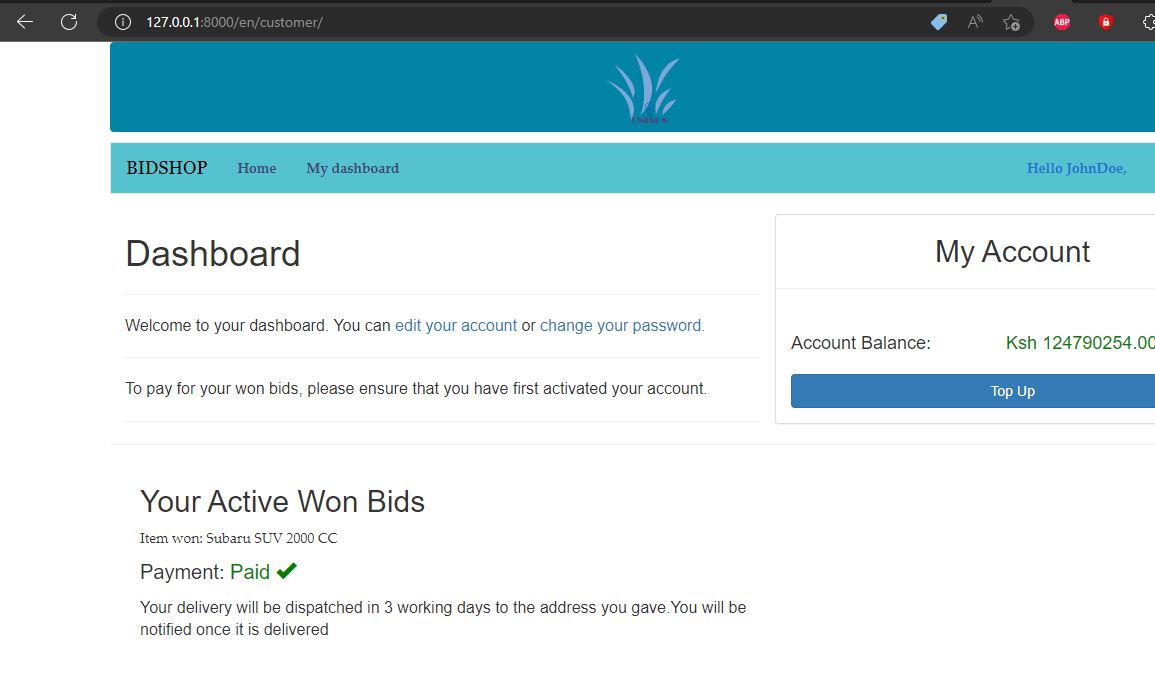
**my auction**The system allows the user to post the ad for the product they want to sell, view their active bids that are in progress and inactive bids that have expired.  
1. Test procedure – After logging in check your account dashboard for active bids  
2. Test data – products to be sold, active and inactive bids.  
3. Expected results –ability to post the ad on the product to be sold and get an access to view active and inactive bids.  
4. Actual results –the user gets access to view active and inactive bids and successfully posts the ads for the product they want to sell.

Figure 4.4.3

**My account**The system allows add money from their credit cards to their account for bidding and purchasing of the items.  
1. Test procedure – After logging in check your account dashboard for balances.  
2. Test data – account balance.  
3. Expected results –ability to add and view balances in your account.  
4. Actual results –the user gets access to view account balances.

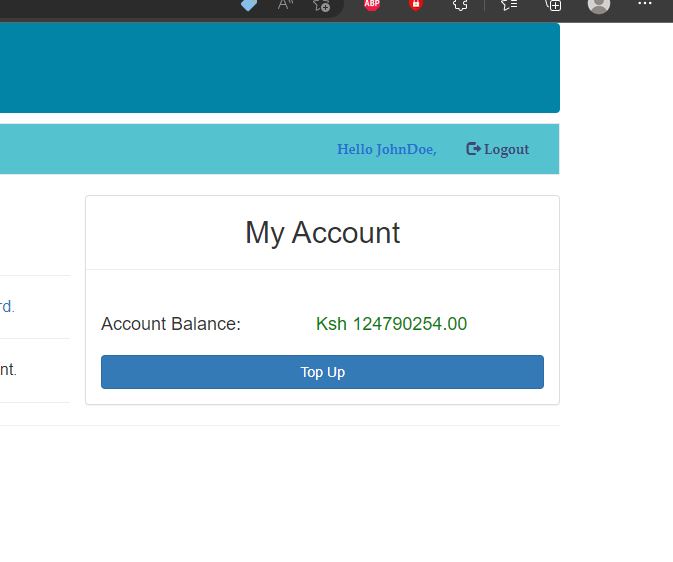
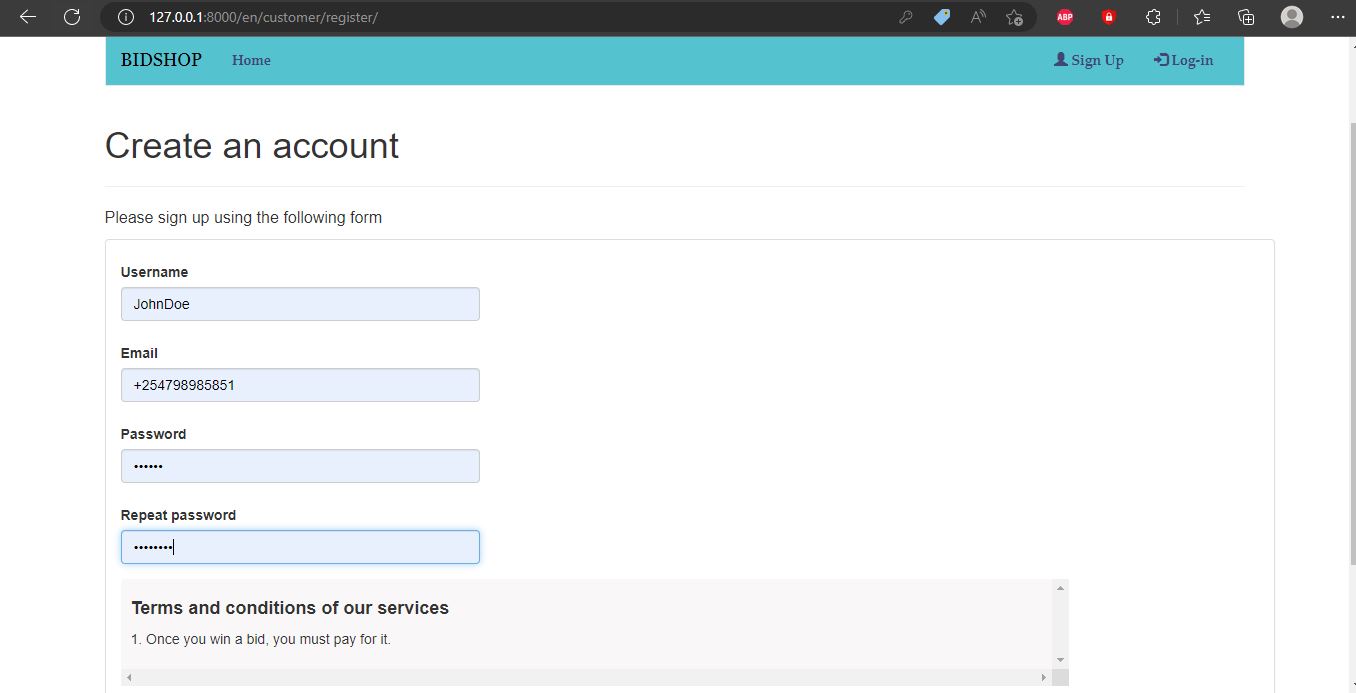
****

Figure 4.4.4

**4.3.2 Reports Generation Test-Case**The admin gets a report whenever wanted stating various products up for bidding and users registered on the website.  
1. Test procedure – Log in as administrator to generate reports.  
Figure 4.4.5

2. Test data – Report generation  
3. Expected results – The necessary reports generated successfully.  
4. Actual results -The requisite updated reports are successfully generated.

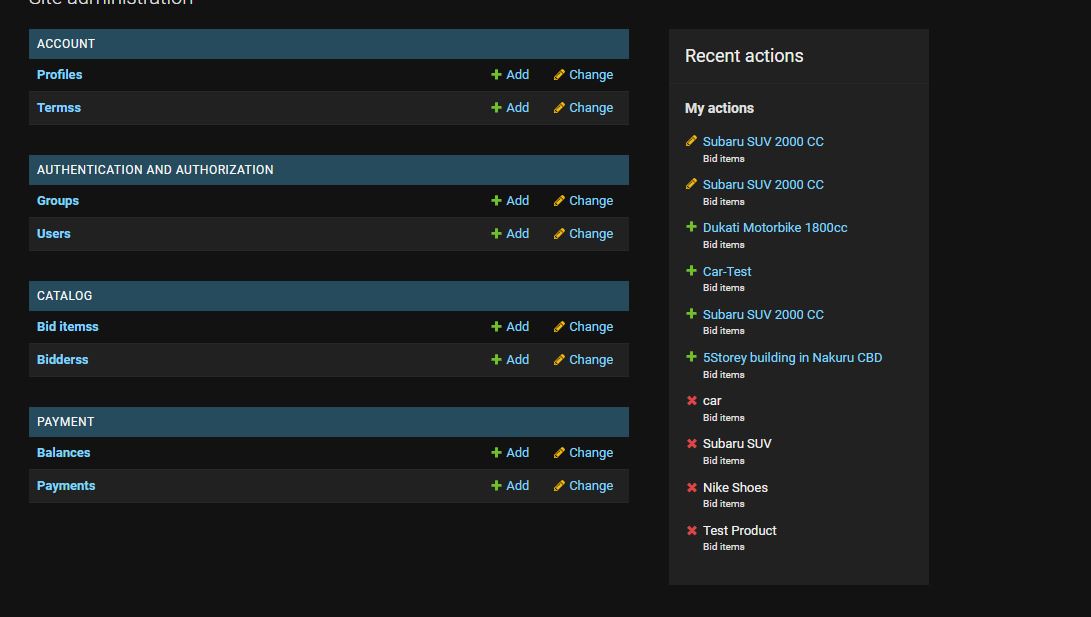
****

Figure 4.4.6

#### **SMS Notification Test**

Auction winners will get an SMS notification whether they won and auctioneer will a notification of who won the auction.

1. Test procedure- A message will be sent to registered mobile number

2. Test data-SMS notification

3. Expected results-the winners and losers gets an SMS notification.

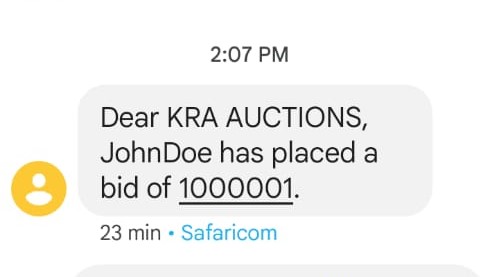


Figure 4.4.7

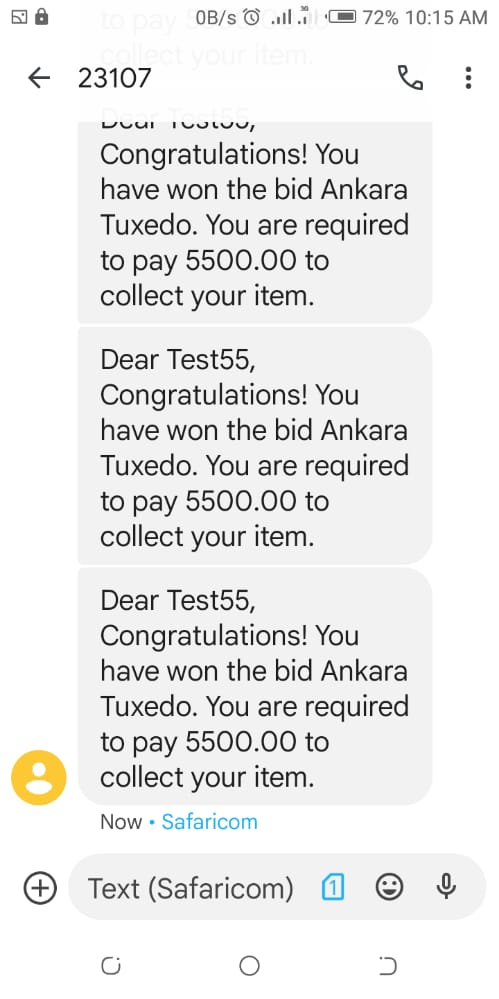
4.Actual results-SMS notification successfully sent to winners and losers.

Figure 4.4.8

## **4.4 Backend Development**

There are several boiler plate and default files and codes that comes with creating a python project( Running :

django-admin startproject app

)

Examples of files that will should be used in creating an apps in the project are:

### 4.4.1 Views.py file

Actually whenever a url is called in Django it finds corresponding function that is in views.py. Using a function you can do following things: Show a webpage with some information passed to it. Submit some information to database. Perform actions such as edit, delete, update. You can also write function for any other actions.

### 4.4.2 Urls.py file

When a web request is made to a Django application, it is the job of the urls.py file to determine what to do with that request. This is what is known as routing in a web application. So, you might have URLs set up in your project like /contact, /stats, /about-us and so on. Each of those routes will trigger a different function in the views.py file.

We run our migration at the beginning stages of development even without implemented data models, this step is necessary for the first creation of a super user/administrator account. After this, you won’t have to run this code again if you want to create another super user. The code below is used.

python manage.py migrate

The following are the apps we defined and used in our project. They are in the settings.py file

INSTALLED\_APPS = [

    'account',

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'whoosh',

    'haystack',

    'catalog',

    'cart',

    'orders',

    'payment',

    'rosetta',

]

The Database we used is defined below.

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.sqlite3',

        'NAME': os.path.**join**(BASE\_DIR, 'db.sqlite3'),

    }

}

Below are the URLS that will be in use in the project. All the routing is done through the links in the list. In the urls.py

urlpatterns = **i18n\_patterns**(

**re\_path**(r'^admin/', admin.site.urls),

**re\_path**(r'^cart/', **include**('cart.urls', namespace='cart')),

**re\_path**(r'^customer/', **include**('account.urls', namespace='account')),

**re\_path**(r'^account/', **include**('django.contrib.auth.urls')),

**re\_path**(r'^search/', **include**('haystack.urls')),

**re\_path**(r'^order/', **include**('orders.urls', namespace='orders')),

**re\_path**(r'^payment/', **include**('payment.urls', namespace='payment')),

**re\_path**(r'rosetta/', **include**('rosetta.urls')),

**re\_path**(r'^', **include**('catalog.urls', namespace='catalog')),

)

The paths/routes are imported for each app as below. It is in all apps using the paths in the url.py file.

from django.urls import **re\_path**

The code below confirms the winner and also send the winning bidder their message. It is in the catalog/views.py

def **confirm\_bid\_winner**(request, id):

    bidItem = **get\_object\_or\_404**(BidItems, id=id)

    now = timezone.**now**()

    bid\_end = bidItem.end\_date

    bidder = Bidders.objects.**filter**(bidItem=bidItem, active=True).**aggregate**(Max('amount'))

    if now > bid\_end:

        # send sms to the winner

            if bidder.**get**('amount\_\_max') is not None:

                bidder\_obj = Bidders.objects.**get**(amount=bidder.**get**('amount\_\_max'))

                winner\_phone\_number = Profile.objects.**get**(user=bidder\_obj.user).phone\_number

                winner\_name = bidder\_obj.user.username

                bid\_item = bidder\_obj.bidItem

                bid\_amount = bidder\_obj.amount

                message = f"Dear {winner\_name}, Congratulations! You have won the bid {bid\_item}. You are required to pay {bid\_amount} to collect your item."

                api\_url = 'https://api.mobitechtechnologies.com/sms/sendsms'

                headers = {"Content-Type": "application/json", "h\_api\_key":"6bb06b8724fcbd880285addf6d22b9b41139f7fd4f5f5ea0318404035afcc44c"}

                request\_body = {

                        "mobile": winner\_phone\_number,

                        "response\_type": "json",

                        "sender\_name": "23107",

                        "service\_id": 0,

                        "message": message

                        }

                response = requests.**post**(api\_url,json=request\_body,headers=headers)

**print**(response.content)

                bidder\_obj.won = True

                bidder\_obj.active = False

                bidder\_obj.**save**()

                bidItem.available = False

                bidItem.**save**()

                messages.**success**(request, f"The winner of the bid is {bidder\_obj.user.username}")

                return **redirect**('catalog:product\_list')

            else:

                messages.**error**(request, "No Winner! This bid ended without a single placed bid")

                return **redirect**('catalog:product\_list')

    else:

        messages.**error**(request, "Error confirming winner")

        return **redirect**('catalog:product\_list')

## **4.5 Testing**

This phase was done especially on the system users. It mainly involved a number of randomly picked college students. The main aspect under scrutiny was requirement inclusion as well as system functionality. The relationship between the system and the objectives earlier laid down was crosschecked too.  
This phase is crucial in ensuring the system is developed under the system, user and functionality framework

**4.5.1 Unit Testing.**Involved testing minimal software components and subcomponents or modules such as users log in, registration and creation of new account, among other modules. Each module was tested individually to verify that they function correctly as per the functionality specification. For instance, a module like for registration and authentication was tested to see  
whether it actually allowed for registered users to log into the system using the assigned credentials. Here also, the test was about confirming that the registration details were captured and stored in the database.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Test Area** | **Expected Results** | **Actual Results** |
| Unit Testing | Testing individual modules of the system | Every module having the capability of running independently and give the expected output | The modules worked appropriately and were able to give the correct output |

*Table 4.1*

### **4.5.2 Integration Testing**

Different modules of the system were combined together and tested as a whole. In this process, we verified that the individual components integrated were able to work together and interact well without any conflicts. We also tested that the integrated modules were able to meet the stated user needs.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Test Area** | **Expected Results** | **Actual Results** |
| Integration Test | Modules Relationships | Different modules in the system working together in the expected way | The modules were able to work together and produce the expected results |

*Table 4.2*

**4.5.3 System Testing**

Involved testing the entire system components as a whole.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Test Area** | **Expected Results** | **Actual Results** |
| System testing | Entire system | The system meeting all the objectives that were listed in the first chapter | The system was able to meet all the objectives that had been proposed |

*Table 4.3*

**4.5.4 Acceptance Testing**Activities involved taking the final system to the real users of the system to test the system for themselves. After the testing it was found that the system had good usability and could be easily understood even by people with little computer literacy

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Test Area** | **Expected Results** | **Actual Results** |
| Acceptance Testing | System’s acceptability and usability | Farmers being able to navigate through the site with minimal effort. | Farmers with the lowest level of education were able to use the system with no assistance. |

*Table 4.4*

**4.6 Deployment Methods**Listed is the overall operations that were performed during the deployment of the system.  
1. Release – The system prototypes were released for user testing and reviews.  
2. Installation - Involved both hardware and software installation for the use. We acquired and installed the required hardware for instance laptops and configured the new system on them. We also created space for the system and database files on the machines and configured application system directories. We also used a free hosting website for  
some components of the system that were to be accessed through internet.  
3. System Execution - Execution phase was performed to confirm the performance of the system as required by the users after which we did collection of the reports on processing, performance and output results of the new system.  
4. User Training and Orientation - This involved introducing the users willing to use or learn more about the system, training them on how to use the attachment management system comfortably.  
5. Security - For security purposes, the following were implemented:  
**Password Security** - We Used passwords for authentication.   
**Web System Security** - We Used POST instead of GET to submit data of sensitive information.  
**Server Security** - We Updated and will continue to update the control panel regularly. We removed software not in use from the server that could lead to security breach.  
6. Installed System - After the finalized Installation of the System, We checked to ensure that the new installed system was in the appropriate working server and environment.  
7. Maintenance - We were involved in the monitoring and reviewing of the new system’s performance and problems. We compared the results of the new system with existing system to assess performance difference.  
8. Evaluation - Upon comparison of the performance and reliability of both existing and new system, we conducted an analysis to evaluate their differences and similarities and the service delivery efficiency.  
9. Post Implementation Review Summary - Finally, we wrote a report which identified any techniques and practices used during the development of the project that worked extremely well, and which would benefit current and future projects.

# **CHAPTER 5 CONCLUSION AND RECOMMENDATIONS**

## **5.0 Conclusion**

The entire process of system development has not only been mind-taxing but resource-demanding as well. This is however not to say it hasn’t been fun having to endure it. The satisfaction with which being part of a solution to a problem in the society using the knowledge and skills acquired through learning is a huge complement that accompanies the process. This process has enabled us to practice and improve on our programming skills. It has proven to be a strong foundation to our future in our careers.

## **5.1 Recommendations**

Our recommendations for future developers wishing to further improve on the system include:  
1. Improve on the industry usage of the system. Perhaps the system should be linked to an active bank account system.  
2. Additional integration for delivery services.  
3. An integration of rating system for auctioneers.

## **5.2 Future Works**

The work presented provided different possibilities for further work. Whereas some of them concern straightforward extensions of the presented approach and its tool support, there are possibilities to combine the results of the thesis with another research. Future researchers should deploy an independent application system whereas functionalities could be added.

# **APPENDIX**

## **Time Tabling**

|  |  |  |  |
| --- | --- | --- | --- |
| Activity | August | September | November |
| Documentation |  |  |  |
| Requirement analysis |  |  |  |
| System Design |  |  |  |
| Implementation |  |  |  |
| Testing |  |  |  |
| Deployment and presentation |  |  |  |

## **Budget**

|  |  |
| --- | --- |
| **Item** | **Budget (Ksh)** |
| Laptop | 36500 |
| Internet connectivity and SMS service | 8000 |
| Smartphone | 12400 |
| **Total** | **56900** |

# **REFERENCES**

Zhang, Yang, et al. "Auction approaches for resource allocation in wireless systems: A survey." *IEEE Communications surveys & tutorials* 15.3 (2012): 1020-1041.

Hasker, Kevin, and Robin Sickles. "eBay in the economic literature: Analysis of an auction marketplace." *Review of Industrial Organization* 37.1 (2010): 3-42.

Dutta, R. and Ramamoorthy, K., International Business Machines Corp, 2009. User rating system for online auctions. U.S. Patent 7,552,081.

Konia, B.S., MARKET MY SITE Inc, 2007. Online auction bid management system and method. U.S. Patent 7,225,151.

Milunovic, S. and Filipovic, J.,2013. Methodology for quality management of projects in manufacturing industries. Total Quality Management & Business Excellence, 24(1-2), pp.91-107.