

**FACULTY:** **FACULTY OF COMPUTING AND INFORMATION MANAGEMENT**

**COURSE: BACHELOR OF SCIENCE IN INFORMATION SECURITY AND FORENSICS**

**REG NO. 19/05892**

**NAME: MARWANGA FERNANDES**

**TITLE: TRAN-SMART WEB APPLICATION.**

**SUPERVISOR: MR. ERNEST MADARA**

**DATE: 07/04/2022**

**UNIT: BSIF3500**

**FINAL DOCUMENT**

**Table of Contents**

[**PROJECT PROPOSAL 1**](#_heading=h.gjdgxs)

[**INTRODUCTION 1**](#_heading=h.30j0zll)

[Project Background 1](#_heading=h.3znysh7)

[Problem Statement 1](#_heading=h.1fob9te)

[Proposal Solution 2](#_heading=h.3dy6vkm)

[Project Objectives 3](#_heading=h.1t3h5sf)

[**Project Scope. 3**](#_heading=h.4d34og8)

[**Project Justification. 4**](#_heading=h.iydpnw4gdcjo)

[**Literature Review 4**](#_heading=h.3rdcrjn)

[a. Methodology 5](#_heading=h.lnxbz9)

[**Budget and Resources 6**](#_heading=h.1ksv4uv)

[a. Project Schedule 7](#_heading=h.2u6wntf)

[**Project Risks and Mitigation 8**](#_heading=h.17dp8vu)

[**SYSTEM REQUIREMENTS SPECIFICATION 8**](#_heading=h.nmf14n)

[**INTRODUCTION 8**](#_heading=h.37m2jsg)

[PURPOSE. 8](#_heading=h.1mrcu09)

[ACRONYMS AND DEFINITIONS. 9](#_heading=h.46r0co2)

[OVERVIEW. 9](#_heading=h.2lwamvv)

[**SCOPE. 9**](#_heading=h.111kx3o)

[**USERS AND THEIR CHARACTERISTICS 9**](#_heading=h.3l18frh)

[PRINCIPAL USERS 9](#_heading=h.49x2ik51)

[GENERAL CHARACTERISTICS 10](#_heading=h.2p2csry1)

[**SYSTEM PERSPECTIVE 10**](#_heading=h.2zbgiuw)

[**FUNCTIONAL AND NONFUNCTIONAL REQUIREMENTS. 11**](#_heading=h.1egqt2p)

[FUNCTIONAL REQUIREMENTS 11](#_heading=h.3ygebqi)

[Input. 11](#_heading=h.32hioqz1)

[Output. 12](#_heading=h.1hmsyys1)

[Timing. 12](#_heading=h.41mghml1)

[Processing. 13](#_heading=h.2grqrue1)

[Sequence of operations 13](#_heading=h.vx12271)

[Handling unusual situations. 15](#_heading=h.2r0uhxc)

[Algorithm to transform input to output. 15](#_heading=h.1664s55)

[**Nonfunctional requirements. 16**](#_heading=h.3fwokq01)

[Scalability. 16](#_heading=h.25b2l0r)

[Security. 16](#_heading=h.kgcv8k)

[Availability. 16](#_heading=h.34g0dwd)

[Maintainability. 16](#_heading=h.1jlao46)

[Usability. 16](#_heading=h.43ky6rz)

[Performance. 16](#_heading=h.2iq8gzs)

[Portability and compatibility. 16](#_heading=h.xvir7l)

[**SOFTWARE AND HARDWARE REQUIREMENTS. 17**](#_heading=h.3hv69ve)

[User interfaces. 17](#_heading=h.4f1mdlm1)

[Hardware requirements. 18](#_heading=h.2w5ecyt)

[Server requirements: 18](#_heading=h.1baon6m)

[Client machine’s requirements: 18](#_heading=h.3vac5uf)

[Software requirements: 18](#_heading=h.19c6y18)

[Communication interfaces. 19](#_heading=h.3tbugp1)

[**SYSTEM DESIGN SPECIFICATION 19**](#_heading=h.39kk8xu)

[**INTRODUCTION 19**](#_heading=h.1opuj5n)

[Objective 20](#_heading=h.z337ya)

[Operations of the business relevant to the system being developed. 20](#_heading=h.2nusc19)

[Main inputs: 20](#_heading=h.1302m92)

[Main outputs: 20](#_heading=h.3mzq4wv)

[**Design considerations and constraints 20**](#_heading=h.2250f4o)

[**OVERVIEW 21**](#_heading=h.1y810tw)

[**DEVELOPMENT TOOLS AND STANDARDS 21**](#_heading=h.4i7ojhp)

[**SYSTEM PROCESSES 22**](#_heading=h.2xcytpi)

[Admin Registration Sequence diagram 22](#_heading=h.40ew0vw)

[User Log in 23](#_heading=h.2fk6b3p)

[Filtering of the data to view 24](#_heading=h.upglbi)

[User Log out 25](#_heading=h.3ep43zb)

[User profile update 26](#_heading=h.1tuee74)

[**USER INTERFACE 26**](#_heading=h.1ci93xb)

[**Transactional Interface 27**](#_heading=h.3whwml4)

[Users Login page html code. 27](#_heading=h.2bn6wsx)

[Connecting users page to the database. 29](#_heading=h.qsh70q)

[**Reporting Interface 30**](#_heading=h.3as4poj)

[Html code for the Areas page. 31](#_heading=h.1pxezwc)

[Php code to connect area page to database. 34](#_heading=h.49x2ik5)

[**APPLICATION SECURITY 35**](#_heading=h.2p2csry)

[Authentication 35](#_heading=h.147n2zr)

[Authorisation 36](#_heading=h.3o7alnk)

[**DATABASE DESIGN 37**](#_heading=h.23ckvvd)

[ENTITY RELATIONSHIP DIAGRAM 37](#_heading=h.3jtnz0s)

[DATA NORMALIZATION. 38](#_heading=h.1yyy98l)

[**DATA 39**](#_heading=h.1hmsyys)

[Data Migration 39](#_heading=h.41mghml)

[**DATA DICTIONARY 40**](#_heading=h.2grqrue)

[**Integrating the Maps to the website. 43**](#_heading=h.3x8tuzt)

[**Archiving Policy 44**](#_heading=h.vx1227)

[**IMPLEMENTATION 45**](#_heading=h.rjefff)

[**Assumptions 45**](#_heading=h.1v1yuxt)

[**Challenges and their mitigations 46**](#_heading=h.4f1mdlm)

[**REFERENCES. 46**](#_heading=h.4anzqyu)

# **PROJECT PROPOSAL**

# INTRODUCTION

### Project Background

The world is changing at a rapid pace when it comes to technology and it's uses. According to the Google statistics of internet access, in Kenya alone the number of users accessing the internet has increased to 87% as of 2020. However, even with access to the internet, the economy makes it very hard for people to plan and budget on their living financial costs.

The pandemic for example rendered many Kenyans jobless and some had to relocate. After maintaining the curve, most people who had relocated to their hometowns need to visit the city or towns for reemployment or back to their daily jobs. This shift caused more people to have a hard time since with the economic fluctuation, a lot of things changed from house rent to food costs.

TRANS-SMART refers to an abbreviation to mean Transportation done smartly whereby "Smart" introduces the aspect of technology or rather the input of technology to developing the system. The proposed web application is supposed to ease relocation of its users and prospectus users at large by bringing to their disposal various preferences in terms of what they might need, how to get it and how much it would cost them to make the move successful with minimal or zero hitches.

The application achieves this by giving people the information and knowledge on how much it would take to move stress-free and live in any part of the country from the food, house, clothes, travelling, commuting and even the cost of moving into the town by cargo services. By accessing the website you can get all this information and also trusted contacts to help new people who intend to move into a city without information or a clue about the place.

In conclusion, moving out or into a new place especially to a new city/town can be very hectic. TRANS-SMART is one of the solutions to curb this challenge and reduce if not eliminate the hustle and struggle that comes with the aspect of moving.

### Problem Statement

Different international websites give an estimated value of living in different countries and even some blogs write about the same. The problem however is that most of the information present in these sites is on the higher side than it is on the lower and thus being more relevant or impactful to tourists or people who would not mind spending more money than they expected. This makes it inconsiderate of the low income citizens and also inaccurate to the expense of poor planning by the users accessing this information.

The sites also do not give an all round understanding of the different locations given in a particular town or city. For example where a house/store is located with the relative prices indicated or even recommendations on where to access the areas or what stores for your desired budget.

The user therefore, is limited on their options and how to go about exactly and given that most cities have con artists rather than trusted people, one is likely to get conned by paying for a house to the wrong agent for example. Other than the cost of living and maybe houses on rent or even for sale, it's hard to know the exact place where the property is located until you search for the place by leaving the site or page.

### Proposal Solution

The Trans-Smart Application is a system which allows a user who is either new or a regular user to search for information regarding the cost of living in any town or city in the country by giving them a variety of options regarding housing costs, movement/relocating, food and even clothes.

The system prompts the user to give his name, place to reside, amount of money planned to use and if at all the user is not able to clearly state their preferences, then the system gives a list of recommended listings along with their budget and estimated costs. The user also gets support if necessary when encountered by a challenge while navigating through the system by contacting the organisation on the same.

### Project Objectives

The proposed system is intended to achieve the following objectives:

Feasibility Study - To understand and find the various challenges different people encounter when moving out to a new area or even moving back into their previous place in the city.

Analysis - To get more information about how best the proposed system can be of great importance in terms of solving the problem at hand.

Design - To come up with the relevant architecture of the system and structure which will be used to develop the system effectively.

Testing - To run the system after integrating the different tools required to develop the system and correct the bugs if present.

Implementing - To implement the proposed system and make it accessible to the prospectus users which in this case is the public.

## Project Scope.

Currently, the proposed system is aimed at solving the transition problem of moving the city of Nairobi which is more common and also one that has a lot of people shifting in and out. This however, does not mean that the other es will not be incorporated into the system.

Given that a lot of information is still available and holding this data will require extensive research, the limited time given to develop the system limits the scope to one that can be obtained with ease. In the near future, the system may be developed to incorporate the various cities available.

## Project Justification.

A lot of people have had the chance to move in and out of the city due to various reasons and owing to the most common reasons which are; studying in the capital and seeking employment, it is more relevant to have a point of access that can give required information that is related to your preferences and a reliable source. By providing this information, new migrants or even those who have been to the capital before can be able to have a stress free transition with minimal inconveniences.

## Literature Review

Is the budget planning web application new?. The system is majorly as direct as its name indicates. Majorly offering a variety of options to a user by promoting them to fill in their desired place of residence and the cost of living in these places respectively. There exists such Systems and even blogs that aim mainly to give the relevant information to it's users with time. Examples of these applications and sites include;

These systems provide information by giving various methods of getting it namely:

* The use of True/False choices for people who know what type of place they would like to reside or what they need.
* The use of filtering options for the user to give their scope of what they need and only that.
* The use of images to give information of the products and costs respectively.

What if there was such a system that gave all of the above functionalities and also gave you the exact figure in terms of cost, give you a relative budget as proposed by the system, offer you the options of recommended stores, moving methods and what is flexible for you. For all these functions' execution, the Trans-Smart Web Application gives the solution to this with an accurate and comfortable experience..

### Methodology

The Data collection methods included the following:

**Observations**: Personally, I had to relocate from my campus town location after the pandemic struck the country and move back home since I was enrolled to online classes and this means that most of the months that I did not live in the house I resided in, had to be paid since I live in a different town and this was very costly. However, having knowledge of the town after the COVID Curve was reduced, I got a job back in the city and even this gave me a very hard time to move back into a different house. also Considering the fact that most houses were expensive due to the pandemic effects on the economy. This is just my own observation and one that affected most of my colleagues.

**Document and Records**: I was able to get more information from the documentations done by the previous developers of similar systems and understand more on their grounds of developing the system and the objectives as well.

**Interviews**: Asking Questions was a key factor since I had to learn and understand more from the prospective users. By interviewing most of my colleagues and even random people who lived in my apartment for example, I had the chance to know what impact moving out to a new place or even back to the town had to their lives economically and even socially.

Most people have raised concern on the struggle they encountered and even encountered when they have to move to a new place or even travel back since no information is given that is based on the foreseeable future and therefore planning and budgeting for their stay, was a hard struggle and not a very comfortable experience altogether.

On getting the chance to hear about what the proposed system would aim at accomplishing, most users appreciated this input and promoted its impact in terms of what benefits the system would yield and aid in improving users' experience all together.

## Budget and Resources

The resources for the project include the following:

1. **Html, CSS, Bootstrap**: This will be used to develop the website's interactive user interface such as login forms and fields.
2. **A Database**: The database gives the storage, manipulation and update of client information together with record keeping of the collected data from the system.a
3. **Javascript**: This offers the dynamic web pages to the user when browsing or navigating through the system.
4. **An internet browser**: This is an important tool for any user who would like to access or use the system since it is a web based application. This can be done using any computing device that can connect to the web such as a smartphone, pc or laptop.

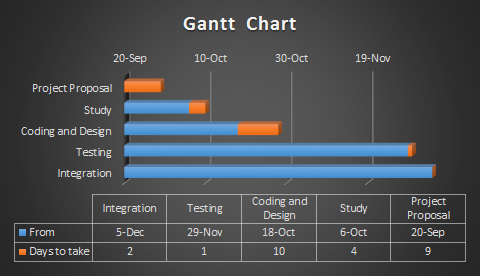
| TOOL | DESCRIPTION | COST |
| --- | --- | --- |
| HTML, CSS, PHP | This are languages to use in developing the website interface | Free |
| A Database | MySQL, XAMP/WAMP Server | Free (Accessible on the internet) |
| Java Script | This will enable coding of the dynamic web pages | Free |
| An Internet Browser | Mozilla Firefox, Google Chrome | Free(Accessible on the internet as a free software) |
| Web Host | Hostwinds, Blue Host, Hostinger, AWS | Ksh. 750 - Ksh. 1000 |
| Maps | Snazzy Maps, Mapme, Maptive | Free(Accessible on the internet as a free software) |

### Project Schedule

| Activity | From | Days to take |
| --- | --- | --- |
| Project Proposal | 20-Sep | 9 |
| Study | 6-Oct | 4 |
| Coding and Design | 18-Oct | 10 |
| Testing | 29-Nov | 1 |
| Integration | 5-Dec | 2 |

The following Gantt Chart shows the progress of the project and is as follows:

* **Project Proposal**: From 20th September 2021 to 29th September 2021
* **Study**: From 6th October 2021 to 10th October 2021
* **Coding and Design**: From 18th October 2021 to 28th November 2021
* **Testing**: From 29th November 2021 to 30th November 2021



## Project Risks and Mitigation

* + The nature of the application to be developed involves receiving personalised data from the targeted users for example; their name, email address, phone number, budget and the city they live in or that which they would like to move into.
  + This therefore poses the risk of losing their data due to the common cyber attacks which might lead to corruption of the data. This risk can be minimised by using legitimate software tools such as the database to store the data and secure code (java script)which limits infringement of the data especially through session hijacking.

## 

| **Risks** | **Mitigation** |
| --- | --- |
| * Loss of personalised data collected from the users of the system through cyberattacks. | Enhancing security of the user accounts by prompting the user to use a strong password before registering. |
| * Corruption of the data stored in our database which will hold the users’ information | Ensuring that the database is encrypted and only authorised users will be able to access the database |
| * Failure or shutting down of the system due to huge traffic to the server | Advancing to a hire server or a more powerful server that can handle high traffic with time |

# SYSTEM REQUIREMENTS SPECIFICATION

## INTRODUCTION

### PURPOSE.

The purpose of this document is to give a more detailed description of what requirements specification is needed to develop the system and also give a detailed view of what the system entails. This document will illustrate the different parts of the system in relation to what their functions are and what the effect of the parts to the system.

### ACRONYMS AND DEFINITIONS.

TN Web App - Tran-Smart Web Application.

SRS- System Requirement Specification.

GUI - Graphical User Interface.

PC – Personal Computer.

DB- Database.

PL- programming language

### OVERVIEW.

The aim of coming up with this document is to give the relevant stakeholders involved in developing the application a clear understanding of what is required to design, implement, test and operate the application. The purpose of this document is to provide an understanding of the requirements that are applicable to the design, implementation, testing and operation of the Trans-Smart web Application.

This will entail the users, functional, software and hardware requirements of the system. For example the use case diagrams, sequence diagrams.

## SCOPE.

Tran-Smart will give prospective users the functionality of searching for various destinations in the city and filter out their results based on different aspects. Users will be able to also view the different areas’ locations and a vast knowledge of the area before making a decision to move into the place or visit the place. This will be achieved by simply having access to the internet and having a web browser.

The system is based on a relational database which will hold relevant information about the various areas to be visited or aimed at visit and also collect the information collected from the various users that will access and use the website.

## USERS AND THEIR CHARACTERISTICS

### PRINCIPAL USERS

The main users of the system include the administrator and registered users or regular users with an account to the system.

### GENERAL CHARACTERISTICS

The users should have access to the internet and have prior knowledge of using an internet browser to access the system from the web. The users should also have an understanding of the various terms used in the system for their swift interaction.

The first page of the site will have two buttons, one two prompt the user to login to the system and a different button to prompt the user to sign up or register to the system. Registration however is an option and therefore if this is not the user’s interest, they can just view the available information without advanced filters.

Below is a description of the different users and their characteristics;

**User: Regular (registered) users.**

Students should have:

* active internet access
* A device connecting to the internet.
* A web browser.

They will have the following permissions:

* Registration of their credentials to the system
* Logging in to the application.
* Filter out the desired results in accordance to different options.
* Perform normal operations of the system such as navigate and view the data and add comments.

**User: administrator.**

The administrator of the system will be able to carry out the following functions;

* Creating, deleting and editing of user accounts.
* Viewing of user information.
* Updating or making changes to the system.
* Maintaining and monitoring of the system.

## SYSTEM PERSPECTIVE

Given that the system will be running on the internet, it is necessary to have a web browser that accesses the internet. The user interface is developed with focus on simplicity as one of the aspects that will enforce ease of using the system. The different parts that constitute of the user interface include;

**User log in**: This window displays the user the point that can enable users access the system by entering their credentials for authentication.

**System overview**: This window provides the administrator with the information about the users that have accessed or regularly access the system and also a general view of the operations of the system.

**Areas**: This will show the users or rather display the various areas available and the different properties and facts about the places.

**The hardware**: The required hardware will mainly be a personal computer or a smart mobile phone to access the internet via a web browser.

**User interfaces**: The application’s user interface incorporates buttons, toolbars, menus and grids to provide the user with an easy and flexible option of navigating the system.

## FUNCTIONAL AND NONFUNCTIONAL REQUIREMENTS.

### FUNCTIONAL REQUIREMENTS

This illustrates the requirements that have effect on performance or operations of the web application that is, the functionalities.

The main modules to the system are, the user and the administrator.

### Input.

* Log in: All users registered to the system will be required to log in to the system every time they would like to use or access their profiles in the system by providing valid credentials to the input boxes displayed to their screen.
* The passwords will then be hashed and saved to enhance their security.
* Sign up: Every new user who wishes to have an account and register to the system’s database will be able to do this by clicking or rather selecting the sign up button and they will be prompted to input some required details such as their name and a password for purposes of later access to their profiles.
* Viewing and navigation: The menus and buttons will be tools for the user to use when navigating the site and searching for relevant information to them.
* Commenting: There will be a text box to collect views and insights from the users and visitors about the site mostly to help improve the system.
* Log out: If the user wishes to leave the site, they can do so by selecting the log out option.

### Output.

* Log in: After verifying that the login credentials are correct, the system displays a message to show that the login was successful. Otherwise an error message will appear to outline the error encountered and how to solve the issue.
* Filter search: If the filter selected is valid and a number of areas that meet those specifications are available, the system will display the results. Otherwise an error message will be displayed and an alternative solution offered.
* Comments: The comments to be typed will be displayed to the user as they type as well and the various ways they can communicate and send their feedback to the administrator.
* Changing the user profile: In case the user decides to change their profile in terms of editing their information, they can be able to see the various changes made reflected to their accounts.
* Log out: After logging out the user will get a message to show that the log out attempt was successful.

### Timing.

Authentication of the users and access to the system should not take more than four seconds. Given that the system depends on the user's access to the internet as well, then if their internet connection is strong, it should take less than two seconds to navigate across the various pages and perform various operations while logged on the system.

### Processing.

Login: The authorization of the user's credentials function is called to check whether the credentials entered by the user are valid or not.

Authorization and authentication: The credentials are checked and matched to the stored data in the database and if at all the details match, access is granted to the user. Otherwise, access is denied.

After logging out, the session is terminated and the dynamic memory of the user’s device is cleared.

### Sequence of operations

**Login:**

**Searching, filtering of the different areas by the users.**

**No**

**yes**

**Updating the user profile.**



### Handling unusual situations.

If the system is too slow in responding to the inputs and commands, check the internet connection bandwidth.

When logging in and an error message is displayed, try checking if the credentials are correct and valid.

### **Algorithm to transform input to output.**

* User log in correct details authenticated by the system.
  + User selects the area(s) they are interested in.
  + Filtering of the results is done where necessary.
* The information on areas is displayed to the user’s interface.
* The user picks the areas they want to know more about and group the data to a more detailed list.
* The system checks the database for more information.
* Input from the user is stored in the database.
* The stored input can be viewed by the administrator.
* The user logs out of the system.

## Nonfunctional requirements.

The requirements below show the different capabilities that system will have and that will enhance its operations functionalities.

### Scalability.

The system enables adding of new features and updating the current properties using new technologies making it up to date and relevant with future changes in technology.

### Security.

The system offers encryption of the passwords used by users to access the website using hashing algorithms and storing the passwords and user information in a secure database that can only be accessed by the administrator.

### Availability.

The system administrator, through the maintenance of the system, will confirm and ensure that the system is available to the users for as long as the user needs to use it at all times and with minimal or zero setbacks.

### Maintainability.

Installation of new features and updating or editing of the system will be made possible by the system administrator as a form of maintaining the web application.

### Usability.

One of the factors put in thought while designing and developing the system is simplicity in terms of usage especially on the user interface. This is important so as to make the system more easy and flexible to use.

### Performance.

The web application will not have the common performance issues such as lagging and hanging as long as the internet connection is strong and the browser is also secure.

### Portability and compatibility.

Accessing the system will be possible from any device that can access the internet using a web browser. This is because the system will run on the net and thus no interruption or affecting of other applications or permissions will be needed.

## SOFTWARE AND HARDWARE REQUIREMENTS.

### User interfaces.

The interface is the primary tool which will enhance the experience and use of the system to the user and this can be done by accessing the system through a web browser and a smart device connected to the internet.

### Hardware requirements.

### Server requirements:

* Operating system: windows 10
* Processor: Intel(R) Core (TM) i5-4600U CPU @ 2.10GHz, 2.70 GHz or higher for actual computation done by the computer.
* RAM:4GB, 8GB or more
* Hard drive: 160GB or more for storage.
* Internet connection: 15mbps or higher speeds.
* Monitor: 14.00 inch: To display the program.

### Client machine’s requirements:

* Operating system: windows 8 and above, MAC or LINUX, android.
* Processor: Intel(R) Core (TM) i3-3600U CPU @ 2.0GHz, 2.70 GHz or higher
* RAM: 4GB or more
* Internet connection: 5mbps, to be used to connect to the browser and the application.

### **Software requirements**:

Below are the different software requirements used in developing the system:

* Operating System: Parrot Linux Os
* Front-End: JavaScript, HTML, CSS.
* Backend Software: MySQL Database, PHP, Python Programming Language.
* Database: The database to be used is MySQL due to its availability, cost effectiveness, reliability in terms of storage and ease of configuration.
* HTML, Bootstrap and CSS can be used for styling of the pages and sheets.
* Programming language: JavaScript is used in order to offer the user the dynamic web pages option while navigating the system.
* A web browser: Chrome, Firefox, google app or Microsoft edge.

### Communication interfaces.

To access the system, the user should have a stable connection to the internet either on a wireless network, cabled or even via a modem.

* A dial up modem of 52 kbps.
* Broadband internet.
* A dial up or broadband connection with an internet provider.

# SYSTEM DESIGN SPECIFICATION

## INTRODUCTION

This document is the reference point of the solution attained. Through describing the requirements, environment, system architecture, files, database design, output and inputs, interfaces, design and processing illustrations, the document is very important in developing the Tran-smart web application system.

### Objective

The main purpose of this document is to give the developer or an interested technical expert in developing the system in terms of how the application architecture is designed and the necessary requirements needed to achieve the or rather develop the system.

### Operations of the business relevant to the system being developed.

The main operations that are to be performed by the system are:

### Main inputs:

* User’s details during registration and a password.
* Regular users will need their username and password during login.
* Login details from the user (both new and regular users).
* Search for specific areas, houses, products and budget selection.
* The area data and cost information done by the developer or the system administrator.

### Main outputs:

* A page displaying the data of the areas after selection by the users.
* A connection link which will display the respective maps to the area to give the location of the area as selected by the user.
* The view or comments box to gather the user’s views and comments which will aid in improving the website better.

## Design considerations and constraints

* The client can use any operating system using any browser.
* The number of users on the system is not limited to a specific number.
* The client is expected to have stable network connection since it is an online platform.
* The system will only give data of the capital city for now due to the reason that the data of the whole country can be very tedious to collect and that will also require greater storage space.
* The system will not be able to give the user an option of interacting with a telemarketer or agent (human) for guidance because the system is not currently based on an enterprise firm or organization.

## OVERVIEW

This document will give a clear and precise view of the system in terms of the design approach and architecture in relation to how it is developed and highlighting the challenges encountered with the relevant solutions.

It is important to note, the document is not the final representation of the actual system after its implementation but an overview of the system design.

The various software and hardware used to implement the idea and architectural design can also be understood better by some of the class and entity relationship diagrams.

The system will be developed using mainly PHP,Javascript and MySQL. Whereby, MySQL will act as the main database to store the information from the user and the data which will be accessed by the user.

PHP and JavaScript on the other hand will be used to code the back-end connection of the web pages (designed using HTML and CSS) to the database to enable ease of communication from the database to the website and vice versa.

Additional tools include QGIS which will enable the use of google maps to navigate through the system, Python as the programming language to code the login page for security reasons and a web browser to access the internet with.

## DEVELOPMENT TOOLS AND STANDARDS

Developing the system involves the use of various tools. The tools are highlighted below along with the clear description of the exact tool to use in terms of version and the type.

The tools which will be used to develop the application include;

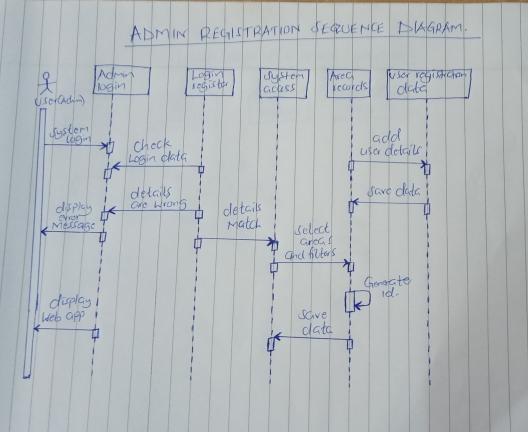
1. XAMPP - This is a software package that has a collection of various applications that will be useful in developing the website for example MySQL and PHP.
2. PHP - This tool is resourceful in developing the code to connect the website to the database (MySQL) to enable sending of data to the database and viewing the data stored in the database.
3. HTML and CSS - This language will be used to design the web pages and add content that will appear on the website.
4. Google Maps - This will be used to give the user easy navigation through the area they would like to view according to their choice of destination.
5. Python - This language will be used to code the login/sign page of the website to enhance the security of the website.
6. MySQL - This will be the main database to be used in storing the data that will be collected from the user and that will be accessed by the user.

## SYSTEM PROCESSES

The various processed carried out in the system will be;

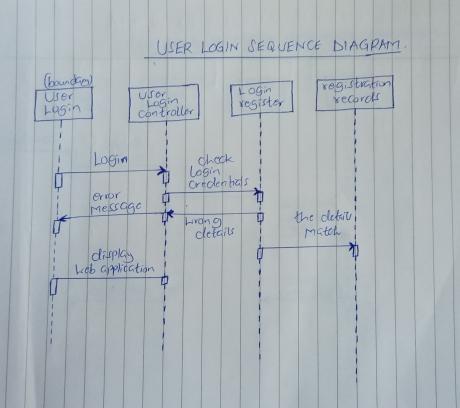
### Admin Registration Sequence diagram

Below is a sequence diagram, to show the process of executing the registration action initiated by the user hitting on the sign up or the register button.



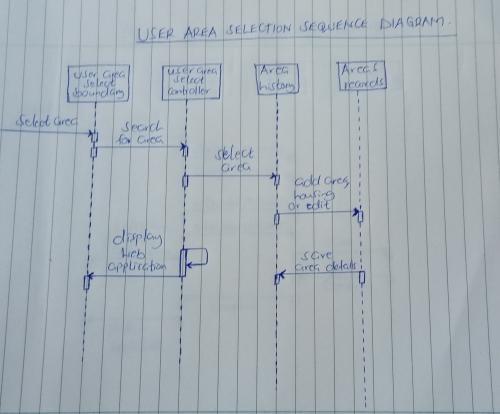
### User Log in

Below is an illustration of the user login process done by the user if they are registered already to the system.



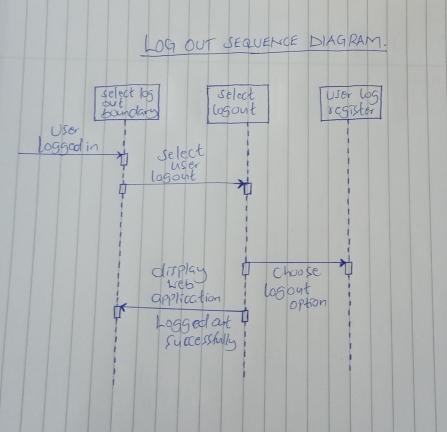
### Filtering of the data to view

The sequence diagram below shows how the execution done by the user of selecting the areas and filtering the areas and housing is done by the system.



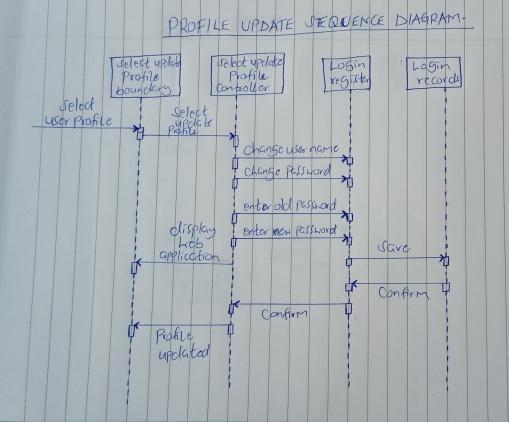
### User Log out

The sequence diagram below illustrates the user log out processed after the user is done with their activity and they would like to exit the site.



### User profile update

Below is a diagrammatic illustration of the process of updating the profile of a user which involves changing their personal details and saving those details in a database. For example changing their username and passwords.



## USER INTERFACE

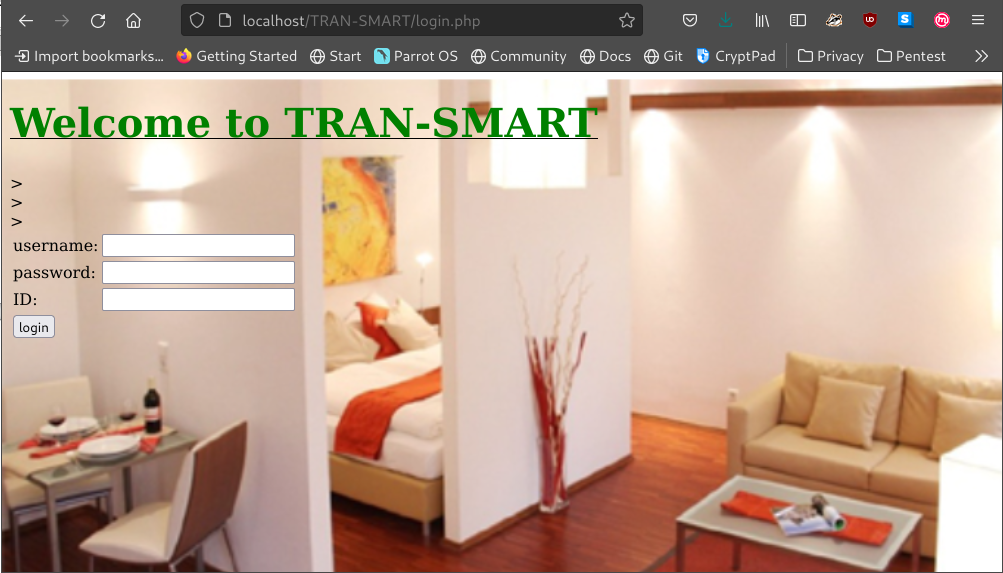
The various interfaces used to develop the system are as listed below. They consist of; transactional interface, Login interface, navigation to various areas interface and the PhpMyAdmin interface used to organize and manage the database used (MySQL)





## Transactional Interface

The diagram below gives an illustration and a source code to show the process of transaction once a user select the amount or budget range which the user intends to use.



### Users Login page html code.

<html>

<head>

<title> users</title>

</head>

<body>

<body background="image3.PNG";background style=background-size:100% 100%">

<ins><h1 style="font-size:250%;color:green;font-family:forte">Welcome to Tran-Smart</h1></ins>

<form action="Connect.php" method="POST">

<table>

<tr>

<td>First\_name:</td>

<td><input type="text" name="First\_Name"></td><br>

</tr>

<tr>

<td>Last\_name:</td>

<td><input type="text" name="Last\_Name"></td>><br>

</tr>

<tr>

<td>Current\_Location:</td>

<td><input type="text" name="Current\_Location"></td>><br>

</tr>

<tr>

<td>ID:</td>

<td><input type="text" name="ID\_NO"></td>><br>

</tr>

<tr>

<td>Phone\_NO:</td>

<td><input type="text" name="Phone\_NO"></td>

</tr>

<td>Desired\_Location:</td>

<td>

<select name="Desired\_Location">

<option

value="Nairobi">Nairobi</option>

<option

value="Nakuru">Nakuru</option>

<option

value="Kisumu">Kisumu</option>

<option

value="Mombasa">Mombasa</option>

</select>

<input type="number"Budget" name="Budget"><br><br>

</td>

</tr>

<tr>

<td>

<a href="Area.php">

<input type="submit"/>

</a>

</td>

</tr>

</table>

</form>

<h3><b><a href="Area.php"><</a></b></h3>

</body>

</html>

### Connecting users page to the database.

<?php

$First\_Name = $\_POST["First\_Name"];

$Last\_Name = $\_POST["Last\_Name"];

$Current\_Location = $\_POST["Current\_Location"];

$ID\_NO = $\_POST["ID\_NO"];

$Phone\_NO = $\_POST["Phone\_NO"];

$Desired\_Location = $\_POST["Desired\_Location"];

$Budget = $\_POST["Budget"];

if (!empty($First\_Name) || !empty($Last\_Name) || !empty($Current\_Location) || !empty($ID\_NO) || !empty($Phone\_NO) || !empty($Desired\_Location) || !empty($Budget)){

$host = "localhost";

$dbUsername = "root";

$dbPassword = "";

$dbname = "Tran\_Smart";

//create connnection

$conn = new mysqli($host, $dbUsername, $dbPassword, $dbname);

if (mysqli\_connect\_error()){

die("Connect Error(".mysqli\_connect\_errno().")". mysqli\_connect\_error());

}else {

$SELECT = "SELECT ID\_NO From Users Where ID\_NO = ? Limit 1";

$INSERT = "INSERT Into Users (First\_Name, Last\_Name, ID\_NO, Current\_Location, Desired\_Location, Phone\_NO, Budget) values(?, ?, ?, ?, ?, ?, ?)";

//prepare statement

$stmt = $conn->prepare($SELECT);

$stmt->bind\_param("s", $ID\_NO);

$stmt->execute();

$stmt->bind\_result($ID\_NO);

$stmt->store\_result();

$rnum = $stmt->num\_rows;

if ($rnum==0){

$stmt->close();

$stmt = $conn->prepare($INSERT);

$stmt->bind\_param("ssisssi", $First\_Name, $Last\_Name, $ID\_NO, $Current\_Location, $Desired\_Location, $Phone\_NO, $Budget);

$stmt->execute();

echo "New record added";

}

else

{

echo "This users exists";

}

$stmt->close();

$conn->close();

}

}

else{

echo "All fields are required";

die();

}

?>

<?php

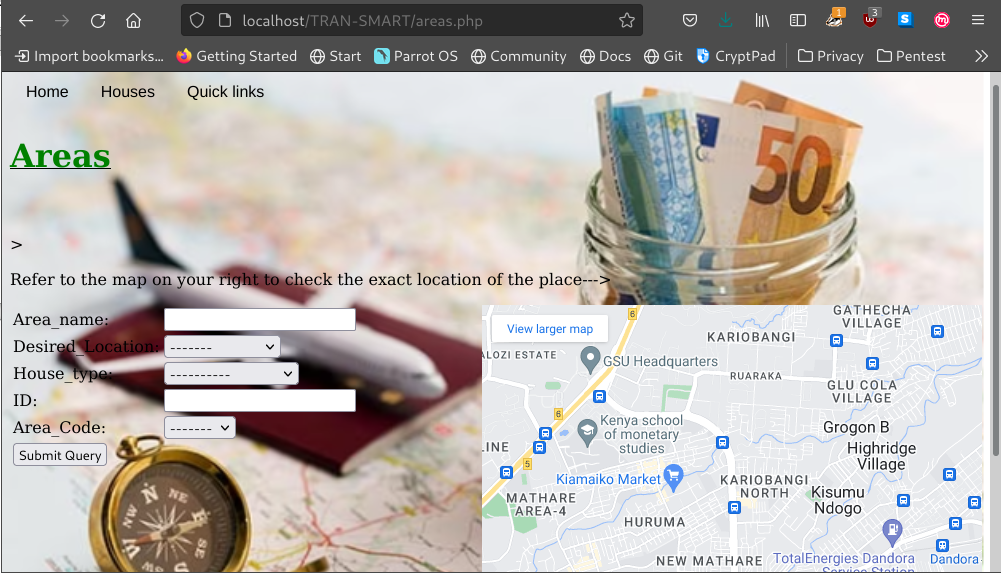
header("Location: Area.php");

exit;

?>

## Reporting Interface

Below is the source code to the connection of the website to the database which collects the users views and information on preferred areas to move in to or visit.





### Html code for the Areas page.

<html>

<head>

<title> Areas.php </title>

</head>

<body>

<body background="image3.PNG";background style=background-size:100% 100%">

<ins><h1 style="font-size:250%;color:green;font-family:forte">Areas</h1></ins>

<form action="Connectareas.php" method="POST">

<table>

<tr>

<td>Area\_name:</td>

<td><input type="text" id="Aname" name="Aname"></td><br><br>

</tr>

<tr>

<td>Desired Location:</td>

<td>

<select name="Nairobi"

id="Nairobi">

<option

value="Mathare">Mathare</option>

<option

value="Muthaiga">Muthaiga</option>

<option

value="Eastleigh">Eastleigh</option>

<option

value="Allsops">Allsops</option>

<option

value="Jua Kali">Jua Kali</option>

<option

value="Baba dogo">Baba dogo</option>

</select>

<select name="Nakuru"

id="Nakuru">

<option

value="Milimani">Milimani</option>

<option

value="Section58">Section58</option>

<option

value="Free-Area">Free-Area</option>

<option

value="Barnabas">Barnabas</option>

<option

value="Mzee wa Nyama">Mzee wa Nyama</option>

<option

value="Mawanga">Mawanga</option>

</select>

<select name="Mombasa"

id="Mombasa">

<option

value="Kizingo">Kizingo</option>

<option

value="Kibokoni">Kibokoni</option>

<option

value="Makadara">Makadara</option>

<option

value="Ganjoni">Ganjoni</option>

<option

value="Tudor">Tudor</option>

<option

value="Nyali">Nyali</option>

</select><br>

<tr>

<td>House\_type:</td>

<td>

<select name="House\_type"

id="House\_type">

<option

value="Bedsitter">Bedsitter</option>

<option

value="Single">Single</option>

<option

value="One Bedroom">One Bedroom</option>

<option

value="Two Bedroom">Mombasa</option>

</select><br>

<tr>

<td>Area\_Code:</td>

<td>

<select name="Area\_Code"

id="Area\_Code">

<option

value="001">001</option>

<option

value="002">002</option>

<option

value="003">003</option>

<option

value="004">004</option>

</select>

<tr>

<td>

<a href="Area.php">

<input type="submit"/>

</a>

</td>

</tr>

</form>

<h3><b><a href="houses.html"><</a></b></h3>

<h3><b><a href="homepage.html">home</a></b></h3>

</body>

</html>

### Php code to connect area page to database.

<?php

$Area\_name = $\_POST["Area\_name"];

$Desired\_Location = $\_POST["Desired\_Location"];

$House\_type = $\_POST["House\_type"];

$Area\_Code = $\_POST["Area\_Code"];

if (!empty($Area\_name) || !empty($Desired\_Location) || !empty($House\_type) || !empty($Area\_Code)){

$host = "localhost";

$dbUsername = "root";

$dbPassword = "";

$dbname = "City Living Budget";

//create connnection

$conn = new mysqli($host, $dbUsername, $dbPassword, $dbname);

if (mysqli\_connect\_error()){

die("Connect Error(".mysqli\_connect\_errno().")". mysqli\_connect\_error());

}else {

$SELECT = "SELECT Area\_Code From Areas Where Area\_Code = ? Limit 1";

$INSERT = "INSERT Into Areas (Area\_name, Desired\_Location, House\_type, Area\_Code) values(?, ?, ?, ?)";

//prepare statement

$stmt = $conn->prepare($SELECT);

$stmt->bind\_param("s", $Area\_Code);

$stmt->execute();

$stmt->bind\_result($Area\_Code);

$stmt->store\_result();

$rnum = $stmt->num\_rows;

if ($rnum==0){

$stmt->close();

$stmt = $conn->prepare($INSERT);

$stmt->bind\_param("sssi", $Area\_name, $Desired\_Location, $House\_type, $Area\_Code);

$stmt->execute();

echo "New record added";

}

else

{

echo "This users exists";

}

$stmt->close();

$conn->close();

}

}

else{

echo "All fields are required";

die();

}

?>

## APPLICATION SECURITY

*The application’s security plays a very important role in terms of securing the user’s details. Both in the database and the website. These is illustrated in terms of authentication and authorisation of the users.*

### Authentication

Below is the algorithm used to execute the users when registering or logging in to the system.

| 1 | START |
| --- | --- |
| 2 | Declare Variables First\_Name, Last\_Name, ID\_NO, Current\_Location, Desired\_Location, Phone\_NO, Budget |
| 3 | Input First\_Name, Last\_Name, ID\_NO, Current\_Location, Desired\_Location, Phone\_NO, Budget |
| 4 | Submit Data |
| 5 | For ID\_NO=1  5.1. Add new record  Else  5.2. Display “User exists”  5.3. Update ID\_NO=ID\_NO+1 |
| 6 | END IF |
| 7 | STOP |

### Authorisation

*After authenticating or rather confirming that the user is valid, access is given to the user and the user is able to login to the system.*

| 1 | START |
| --- | --- |
| 2 | User details confirmed |
| 3 | Input password |
| 4 | Submit Data |
| 5 | For Password=1  5.1. Give access to system  Else  5.2. Display “Incorrect details”  5.3. Update ID\_NO=ID\_NO+1 |
| 6 | END IF |
| 7 | STOP |

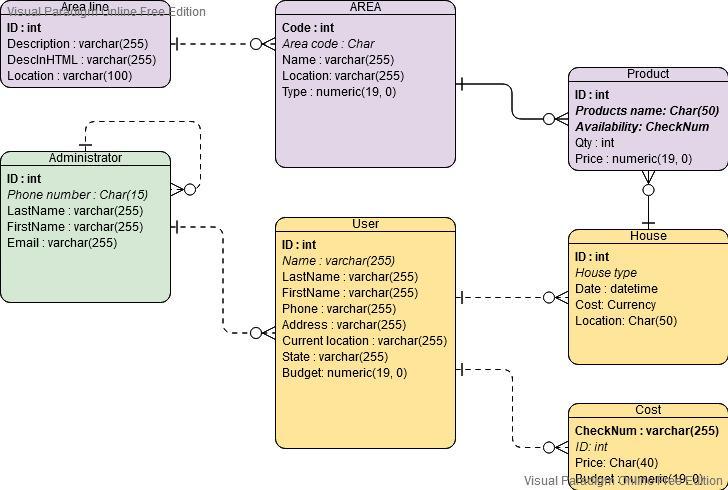
## DATABASE DESIGN

The database used in developing the system is MySQL hosted under phpMyAdmin. Below is a descriptive design of how the database design or architecture is.

### ENTITY RELATIONSHIP DIAGRAM

The entity relationship diagram below gives a structural view used in the design of the database. This shows the relationships of the entity sets stored in a database. This gives a logical structure of databases.

The diagram is a structural diagram that is used in designing database. It shows the relationships of entity sets stored in a database. This illustrates the logical structure of the databases and can be viewed as a conceptual and representational model of data used to represent the entity framework infrastructure.



### DATA NORMALIZATION.

Normalization is the process of organizing data in a database. Its main use is to minimize the redundancy from a relation or set of relations. Its also used in removing some undesirable characteristics like insertion, update and deletion anomalies. This makes database more flexible.

| UNF | 1NF | 2NF | 3NF |
| --- | --- | --- | --- |
| **NEW USER** | **ADMIN** | **REGULAR USER** | **USER** |
| Email address | Admission Number | Username | First Name |
| Password | Password | Password | Last Name |
| First name | Phone number |  | Email |
| Last name |  | **ADMIN** | Password |
| ID number | **USER** | ID | ID Number |
| Area | ID | First Name | Phone number |
| Phone number | Password | Last Name | **HOUSE** |
|  | Area | Area | House ID |
| **AREA** | House | Password | House Type |
| Type | Email Address |  |  |
| Location | First name |  | **LOCATION** |
| Distance | Last name | **COST** | State |
| Code | ID number | Product | Name |
|  | Phone number | Price | Distance from town |
|  |  | Type |  |

## DATA

### Data Migration

Given that most of the data used in the system is outsourced and based on research, the various types of data collected from research and also as views from the users that access the website has to be integrated to the system on a regular basis to keep it up to date.

The migration of the data will involve the steps listed below;

* Transforming the data collected and that collected from the user to the desired format of use.
* Making analysis of what need to be migrated to the system and its importance to the system in terms of impact.
* Extraction of the data from the research fields and the legacy system to use it in the system especially during updates.
* Cleaning of the transformed data from errors after collecting it and doing research before loading it to the system.
* Loading the data to the system. This will be the last step of the migration processes as it is done after confirming that the data is ready to use and transformed to the right mode.

Google maps is a notable tool used in the system and the mapping of the tool to the system is important in giving the user a view of the location in terms of where exactly the area is. The data dictionary gives a detailed illustration of what data is used in the system and the data types.

It is important to note that the system admin has an important responsibility in making sure that the data used in the system is up to date and that the data collected is up to date as well. The responsibility of the user on the other hand is to ensure that their views are consistent and use the right credentials to access the system

## DATA DICTIONARY

The data dictionary offers a reference and illustration of each data element used in the system. It also gives a clear definition of the data model used by associating the various elements Data dictionary is a reference and description of each data element.

| **ENTITY** | **FIELDS** | **ENTITY DESCRIPTION** | **DATA TYPE** | **LENGTH** | **PRIMARY KEY** | **FOREIGN KEY** | **NULL** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Regular user User | User ID | Unique Identification of user registration used to login. | integer | 20 | yes | no | no |
|  | Names | Official names of users | varchar | 20 | no | no | no |
|  | Email address | Unique electronic address | varchar | 30 | no | no | no |
|  | ID number | Citizen’s Identification number | integer | 10 | no | no | No |
|  | Phone number | Mobile number for communication | Integer | 13 | no | no | no |
| New User | ID number | Unique identification number given in institution | Integer | 10 | yes | no | no |
|  | First name | Official first name | varchar | 20 | no | no | no |
|  | Second name | Official Surname | varchar | 20 | no | no | no |
|  | Email address | Unique electronic mail for the user | varchar | 20 | no | no | no |
|  | password | Unique characters to access the system | Varchar | 20 | no | no | no |
|  | Phone number | Mobile number for communication | Integer | 13 | no | no | no |
|  | Current residence | Area where the user lives when registering | char | 10 | no | yes | yes |
|  | Area code | Unique code of the area | lookup | 10 | no | no | no |
| Area | Area code | Unique identifier of the areas | varchar | 10 | Yes | no | no |
|  | Area name | Name of the area selected | varchar | 10 | no | no | no |
|  | Location | This gives the situated area or where exactly the area is | varchar | 10 | no | no | no |
| Houses | House ID | Unique identification of houses | varchar | 10 | yes | no | no |
|  | House type | The various types of the areas that the user can select from | varchar | 10 | No | no | no |
|  | cost | The various costs of the houses based on the type and location | integer | 10 | Yes | no | no |
|  | Availability | This shows whether or not the house of choice is available or not | varchar | 15 | no | no | no |
|  | Location | Location where the house is situated based on the areas data | integer | 10 | no | yes | no |
| Products | Product ID | Unique identification of products | integer | 10 | yes | no | no |
|  | Type of product | The various types of products that a user can select from | char | 10 | no | yes | no |
|  | Price | The cost of the different products the user selects | Currency | 100 | no | no | no |
|  | Amount | This gives an illustration of the different types of the products and their amount | integer | 10 | no | no | no |
| Cost | Cost ID | Unique identification number for the costs of the different products and housing | integer | 10 | no | yes | no |
|  | Area | This describes the area selected by the user to get the cost of living there generally | Varchar | 100 | no | no | no |
|  | Budget | This gives the user a range of what amount of money they plan on using | integer | 10 | no | yes | no |

## Integrating the Maps to the website.

The source code below shows the code used to import the maps to the system. The mapping will enable the navigation of the user in the website to allow the user view the exact area.

<iframe width="425" height="350" frameborder="0" scrolling="no" marginheight="0" marginwidth="0" src="https://www.openstreetmap.org/export/embed.html?bbox=35.60806274414063%2C-1.5777119568340103%2C37.14065551757813%2C-0.7703930452206605&amp;layer=mapnik" style="border: 1px solid black"></iframe><br/><small><a href="https://www.openstreetmap.org/#map=10/-1.1741/36.3744">View Larger Map</a></small>

## Archiving Policy

The information used in the system is collected from both users and research done by the developer. The nature of the application involves the use of data that is dependent on various factors such as the economy at the time of research or when the data is collected. This therefore means the data or information collected might be irrelevant with time and therefore storing of this data will defer in terms of what is more necessary and what isn’t.

The data of users that is collected from the website for example their names, passwords and other personal information will be stored on the cloud and on a hard drive for easy retrieval in case of any loss or tampered to the data. This is information that is sensitive and needs to be archived and backed up on a regular basis to avoid loss or corruption of the data by unauthorised persons.

Archiving of the data can be more described according to the exact data to be stored as shown below;

1. Areas – The data about areas does not need to be deleted or discarded because it is unlikely that the basic information of a place such as the name, will change. Therefore this data can be archived for a longer period of time and to the cloud to save on the disk space.
2. Houses – The information on the housing is greatly affected by the economy in terms of the cost since it is known to change almost after every yearly budget is done especially in developing countries. This therefore means that, the information will need to be stored and updated after every year.
3. Users – The personal information of the users can be updated and stored both on the cloud and the hard drive of the admin for reference purposes and also to secure the data from unauthorised access.
4. Cost – The data on costs of products and housing is also influenced by the economy and should be updated after every year as well to avoid inconsistencies.

The system will also require regular maintenance routines to make sure that the system is up to date running with minimal or if possible no errors.

## IMPLEMENTATION

Deploying the system to the web will involve the use of a web host which will hold the website and make it available to the web for user access. My preferred host for the system is 000hostnger web host due to the fact that it is free for a start-up and also it offers flexible options of payment.

Testing and debugging of the system will involve going through the various steps of using the system as a user whereby, the system developer will register to the system as a new user and use the different functions of the website, confirm that the records are sent to the database, logout and login again as a registered user the again confirm that the details are saved to the database.

Implementation of the system to the desired environment (web) will face some challenges and this section illustrates the challenges to be faced and how to mitigate them along with the assumptions made of system by the developer.

## Assumptions

The following assumptions are in place:

* The users aimed at using the system have prior knowledge of how to use the website and search for the website.
* The users will be able to interpret and understand the information displayed in the site using a bit of previous knowledge about the areas.
* The system administrator or the developer will update the system regularly to make sure that the data collected is correct and up to date.
* The resources or data used for migration will be available even after the migration for reconciliation.
* The system will take a maximum of two seconds to load pages on the click of the user.
* Monthly maintenance routines will be done to ensure that the system is running with minimal or no errors.

## Challenges and their mitigations

Implementing the system for its use to the web may involve some challenges. Below is a list of some of the expected challenges and the various solutions that can be used to counter those challenges.

* The storage space of the database holding the website data may be overwhelmed by the content collected. This can be avoided by backing up the data to the cloud storage and part of it to the hard drive especially the sensitive data such as users’ credentials.
* Legal permits – Having to collect images of houses and architectural designs will involve getting a legal permit especially from the owners. This can obtained by approaching the agents or alternatively using common design sketches online after visiting the area.
* Lagging – The website may experience a lag or being slow in executing the user commands and some functionalities if the server or database pc lacks internet due to power outages or slow internet connection. This can be avoided by using reasonably good speeds of connection and having a backup UPS to avoid the power loss affecting the operations of the pc.
* Errors in data cleaning. This can be experienced after the user makes errors in their input by entering irrelevant data in relation to the specified data type. This can be avoided by providing an error message to the user after entering an invalid input.
* Full Month End snap shots will be available from legacy system for cut over testing and reconciliation
* Cut over will take place immediately following month end
* Once the legacy system is closed off for month end and reconciled no further transactions processing will take place until the new system is available
* The legacy system(s) will be locked for further transactions

## REFERENCES.

Below are some examples of the available web applications that offer the cost of living in

different places along with the respecting sites:

* https://www.numbeo.com/
* https://www.daveramsey.com/
* https://www.nerdwallet.com/
* https://www.bestplace.net/
* https://www.estimatemyapp.com