ASHESI UNIVERSITY COLLEGE

REAL ESTATE MANAGEMENT APPLICATION

APPLIED PROJECT

B.Sc. Computer Science

Benson Murimi Wachira

2016

ASHESI UNIVERSITY COLLEGE

Real Estate Management Application

APPLIED PROJECT

Applied Project submitted to the Department of Computer Science, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science

Benson Murimi Wachira

April 2016

DECLARATION

I hereby declare that this applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature

……..…….……...

Candidate's Name

……..……

Date

…….……..……...

I hereby declare that preparation and presentation of this applied project were supervised in accordance with the guidelines on supervision of applied project laid down by Ashesi University College.

Supervisor's Signature

…….……..……...…..

Supervisor's Name

……..……..….

Date

…….

ACKNOWLEDGEMENT

The completion of this project involved valued contributions from a number of people to whom I greatly thank and would like to express my sincere gratitude.

I would like to thank God for the strength and determination that he gave me while undertaking this project. I wish to express my sincere appreciation to my supervisor, Mr. David Amatey Sampah, for his dedicated time away from his very busy schedule and also for his guidance, comments, suggestions and criticisms that led to the success of this project.

I also wish to thank my parents for their support in all areas. They provided resources and an environment that facilitated the successful completion of the project.

To the Computer Science department staff, I wish to thank Dr. Nathan Amanquah who assisted me greatly during the implementation of this project

Finally, I would like to appreciate the effort of all my colleagues who assisted me streamline my ideas and also those that offered me assistance in various aspects of development of the project.

ABSTRACT

Searching for a property for sale or rent is cumbersome and expensive in Kenya since you reply on the help real estate brokers provide in acquiring a property. This project seeks to find an efficient way of advertising and searching for property. I developed an application that creates contact between property owners or managers and property seekers. The web application can be accessed on both mobile devices and computers. Users can also install an Android or Windows application on their smart phones.

The functionality of this application includes a combination of features from existing solutions. This includes subscription for email or message alerts on properties that a user is searching.

Development and user testing were conducted to evaluate if the application met the functionality stated in requirement specification.

TABLE OF CONTENTS

DECLARATION i

ACKNOWLEDGEMENT ii

ABSTRACT iii

Chapter 1: Introduction 1

1.1. Background 1

1.2. Problem Statement 1

1.3. Project Objectives 2

1.4. Motivation 2

1.5. Assumptions 3

1.6. Significance of the Project 3

Chapter 2: Background and Related Work 4

2.1. Introduction 4

2.2. Real estate technologies 4

2.3. What to look for in a real estate management application 6

2.4. Exiting solutions 7

2.4.1. Buy Rent Kenya 7

2.4.2. Property Kenya 8

2.4.3. Real estate online 9

2.4.4. Property 24 10

2.4.5. Olx 11

2.4.6. Pigiame 12

2.5. Conclusion 12

Chapter 3: Design and Architecture 14

3.1. Requirements 14

3.1.1. User Classes 14

3.1.2. User Requirements 14

3.1.3. System Requirements 15

3.1.4. Functional Requirements 16

3.1.5. Non Functional Requirements 17

3.1.6. System Constraints 18

3.2. System Modelling 18

3.2.1. Activity Diagram 18

3.2.2. Use Case Diagram 19

3.2.3. Sequence Diagram 20

3.3. Architecture Pattern 23

Chapter 4: Implementation 25

4.1. Technologies Used for Development 25

4.2. Classes and Methods 26

4.3. Entity Relational Diagram for Database 28

4.4. Implementation Challenges 40

Chapter 5: Testing 41

5.1. Introduction 41

5.2. Development Testing 41

5.2.1. Unit testing 41

5.2.2. Component testing 43

5.2.3. System testing 43

5.3. User testing 45

5.3.1. User testing results 46

Chapter 6: Conclusion and Recommendation 47

6.1. Introduction 47

6.2. Summary 47

6.3. Conclusion 48

6.4. Recommendations and Future Work 48

References 50

TABLES AND FIGURES

Figure 2.1 Buy Rent Kenya website …….……. 7

Figure 2.2 Property Kenya website .....….. 8

Figure 2.3 Real Estate Online website .....….. 9

Figure 2.4 Property 24 website .....….. 10

Figure 2.5 Olx website ……... 11

Figure 2.6 Pigiame website .....….. 12

Figure 3.1 Activity diagram ……... 18

Figure 3.2 Use case diagram 1 (property owners and seekers)……... 19

Figure 3.3 Use case diagram 2 (administrator) .....….. 20

Figure 3.4 Add property sequence diagram …..…….…….…….…….. 21

Figure 3.5 Subscribe alerts sequence diagram ……..…..……. 22

Figure 3.6 Search property sequence diagram) .....….. 23

Figure 3.7 MVC architectural pattern...…..……..…….…….……. 24

Table 4.1 Backend class methods ……..…….…….26

Figure 4.2 ER diagram 1 (system users, alerts) …..…….…….……. 29

Figure 4.3 ER diagram 2 (users, property, property features, land, pictures) …….…….…...30

Figure 4.4 Add Property (location)...…….……..…….…….……. 31

Figure 4.5 Add Property (details) ……...…..…….. 31

Figure 4.6 Add Property (features) ……..…….……. 32

Figure 4.7 User Login ……..……..... 32

Figure 4.8 Preferred property plan …….……..... 33

Figure 4.9 Upload pictures…….……. 33

Figure 4.10 Review post .....…... 34

Figure 4.11 Property fields .....…... 34

Figure 4.12 User properties …….……... 37

Figure 4.13 Delete property …….…….…….. 38

Figure 4.14 Search property .....…...38

Figure 4.15 Property details page……..…….……. 39

Figure 4.16 Similar properties…….…… 40

Figure 4.17 User registration …….…….…… 41

Figure 4.18 User Login …….…….…… 42

Figure 5.1 user class testing …….……..…….…….. 44

Figure 5.2 Test class implementation ...…….…….…….…….……..44

Figure 5.3 Unit testing result .....…... 45

Figure 5.4 validation (Empty field) …….…... 46

Figure 5.5 validation (Invalid data type)…….…… 47

LIST OF ACRONYMS

KES– Kenya Shillings

USD- United States Dollar

URL- Uniform Resource Locator

MVC - Model View Controller

SQL- Structured Query Language

Chapter 1: Introduction

1.1. Background

Kenya is a country in Eastern Africa and shares boundaries with the Republics of Somalia in the East, Uganda in the West, Tanzania in the South and Ethiopia and South Sudan in the North. With an estimated population of 45 million people, urban population of 24% and urbanisation rate of 4.36%, most Kenyans are looking for houses and land to rent or buy (Mundi, 2015). The most common way of searching for a property to rent or buy is to visit the real estate agencies that are located in major towns and enquire if they have available property for sale or rent. People also get recommendations from friends and relatives about available property in the areas they live. This process is long and cumbersome for the property seeker.

Most of the properties are managed by real estate agencies and if you want to rent a house that is not managed by an agency it becomes very difficult to let people know about it. The property seeker does not have a lot of options to choose from because the information about property is limited.

Often people like staying around places where they work or do business so that commuting is easier, especially in major towns and cities that have a lot of traffic.

1.2. Problem Statement

Looking for a residential house, business house and plots for sale or rent is costly and cumbersome for the property seekers. This is because it involves moving from place to place to enquire of vacant offices and houses or relying on the network that the person has.

The property owners also go through a lot of struggle while informing the public about their properties. The techniques used by the property owners to sell and rent their property include giving the property to a real estate agency to manage or contracting brokers. This

increases the cost of acquiring the property since the brokers charge both the property owners and property seekers a commission.

The solution to this problem is to develop a platform that the property owners can upload their properties and the property seekers can browse to find the properties that they want.

1.3. Project Objectives

The objectives that this project seeks to achieve are;

i. Reduce the time taken while searching for property by the property seekers.
ii. Give the property owners a platform to market their property
iii. Create a connection between the property owners or managers and the prospective people who need their properties.
iv. Reduce the cost that is involved in paying the property brokers to search the properties.
v. Give the property seeker multiple options to choose from

1.4. Motivation

During my summer break 2014, I wanted to rent a house for four months while doing my internship. Since I was not in Kenya, I searched online for houses to rent within the area I was going to work. I did not get any house that met my needs since there were only few properties and rent was very high. The process of getting the house was long and I had to rely on the help of a friend.

During summer 2015, I contacted a friend who was in Nairobi two months prior to my summer break to help me search for a house around the area I was going to work. I couldn't get a house to rent in my desired location hence I visited a real estate agency in search of a house. They charged me a total of 30 USD to facilitate the process.

1.5. Assumptions

To use this solution, I made the following assumptions:-

i. The user of the system can read and understand basic English.

ii. The user has a computer or a smart phone or a feature phone.

iii. The user has access to internet connection.

1.6. Significance of the Project

The aim of this project is to bridge the gap between the property owners and the property seekers in all regions of Kenya. This will reduce the cost involved in searching for property by providing a platform where the two parties can interact at their comfort.

Chapter 2: Background and Related Work

2.1. Introduction

Real estate is land and anything that is fixed such as building, road and structures (Webfinanceinc, 2006). Real estate management is the business of managing land and buildings, including activities such as keeping buildings in good condition and facilitating the renting and sale of property (Webfinanceinc, 2006).

A real estate management application is a computerized system that facilitates the management of property, including maintenance of the property, using a software. These applications have replaced the old-fashioned paper based methods of managing properties, which are cumbersome and inefficient (propertymatrix, 2003).

The application that I am developing does not focus on keeping records of the properties such as maintenance, rent, managers or even residents, but focuses on marketing the property to potential people who might want to rent or buy the property. It also helps the property seekers search for the property they want.

2.2. Real estate technologies

There are two major technologies used in developing real estate management applications.

1. **Desktop applications**.

Desktop applications are mainly used in offices to manage real estate by keeping records about the state of the properties, the tenants, rent payments, accounting, employees and other details about the properties (reviews.com, 2015). They are used by real estate owners and managers.

The programming languages that can be used to develop a real estate management desktop application include;

i. Python- It can be used for SWING GUI application development because it has libraries such as Jython for GUI development.
ii. Java- It is object oriented and has powerful IDE such as NetBeans which ease the design of user interface.
iii. C, C++, Visual basics and C# are examples of languages that can be used to create desktop applications for Windows operating system.

2. Web applications.

The web applications are accessible to the general public. They connect the property owners and property seekers by helping the property owners and managers advertise their property and also help property seekers to easily search for property.

The programming languages that can be used to develop a real estate management web application include;

i. Html- Hypertext marker language is used to provide layout and format of content on a web page (Shiotsu, 2014).
ii. CSS – Cascading Style Sheets is used to design the look of the application by manipulating elements such as color and fonts (Shiotsu, 2014).
iii. JavaScript – JavaScript is used to increase interactivity of the website as well as handle asynchronous communication (Martin, 2015).
iv. PHP- Open source server side scripting language that is compatible across multiple platforms (Shiotsu, 2014).

v. Python- It is a high level programming language that has multiple libraries that ease the work of a programmer since they can achieve more with only a few lines of code (Martin, 2015).

vi. Java- it is a server side language that is mostly suitable for use in websites that have high traffic. This is due to its robust frameworks such as JSP (Shiotsu, 2014).

vii. Ruby – This language differs with other languages because it has multiple ways of doing the same thing. Ruby on Rails which is an open source framework, enables programmers to create dynamic websites quickly and efficiently (Martin, 2015).

This project focusses on the web applications that manage real estate properties. This chapter discusses the functionalities of existing real estate management web applications. It compares the features of these applications and finally introduces a real estate management application that is suitable for use in Kenya. This application combines the features from the existing solutions, introduce new features and also modify features from the existing solutions.

2.3. What to look for in a real estate management application

Below are features that are important and should be considered when choosing a real estate management application to use.

a. The application should be easy to use and navigate.
b. The application should be secure and keep user details confidentially.
c. The application should be available and reliable.

2.4. Exiting solutions

Below is a list of real estate management web applications that are used in Kenya. Each application has been analysed by exploring its features.

2.4.1. Buy Rent Kenya

Website URL: http://www.buyrentkenya.com/

Figure 2.1 shows the add property page of Buy Rent Kenya website.

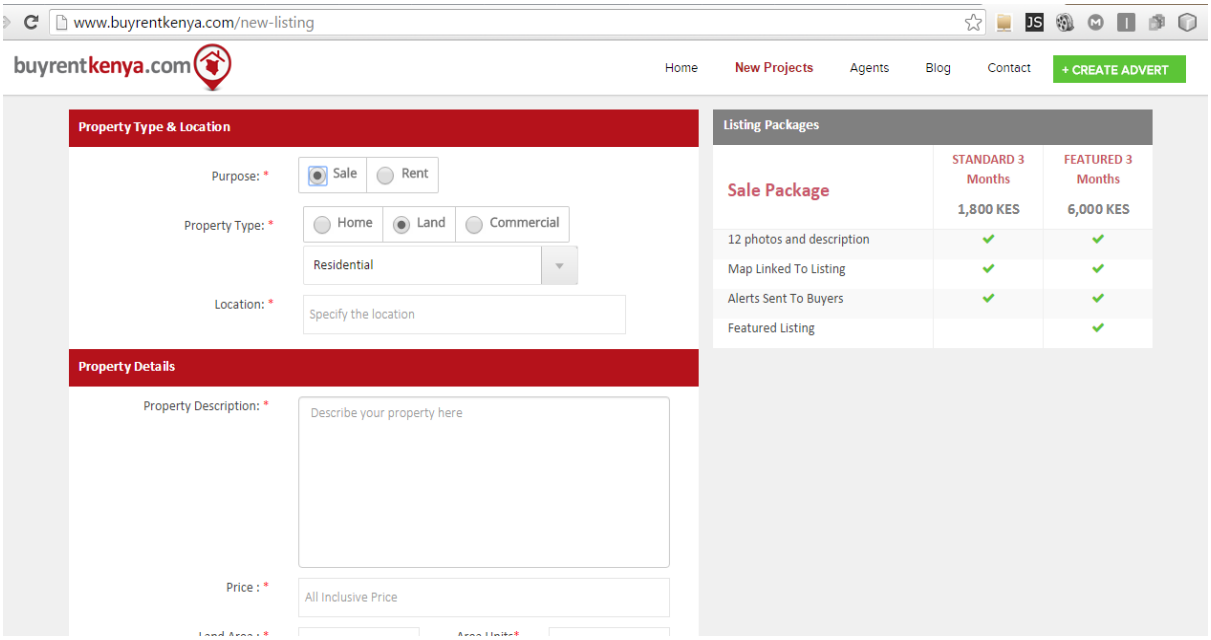

Figure 2.1: Buy Rent Kenya website

Buy Rent Kenya is an ecommerce website that posts ads on real estate creating a platform to advertise and search property throughout the country.

Posting an ad on this platform is expensive while it offers a short time to advertise the property. As a result of the high price of posting an ad, majority of the property on the sites are very expensive hence it does not attract average property seekers who are the majority in the country.

The sale package costs 1800 Kenya Shillings (KES) (18 USD) for standard 3 months listing and 6000 Kenya Shillings (KES) (60 USD) for featured 3 months listing. The rental package costs 300 Kenya Shillings (KES) (3 USD) for standard 1 month listing and 800 Kenya Shillings (KES) (8 USD) for featured 1 month listing. When you post an ad on the site it does not appear immediately on the listing. The user receives an email detailing the steps to follow in order to complete the transaction. After the payment, the user has to send an email or text message to Buy Rent Kenya with the confirmation text they received after making the payment. After completing these steps, the property is activated to appear on the site.

2.4.2. Property Kenya

Website URL: http://www.propertykenya.com/

Figure 2.2 shows the home page of Property Kenya website.

Figure 2.2: Property Kenya website

Property Kenya is an ecommerce website that posts ads on real estate properties.

It has a bad user interface that lacks consistency in the organization of text and contains too many adverts that hinder navigation.

The search for the property is only categorized by cost and hence a user cannot narrow down the search to the property features they want to search.

2.4.3. Real estate online

Website URL: http://www.realestateonline.co.ke/

Figure 2.3 shows the home page of Real Estate Online website.

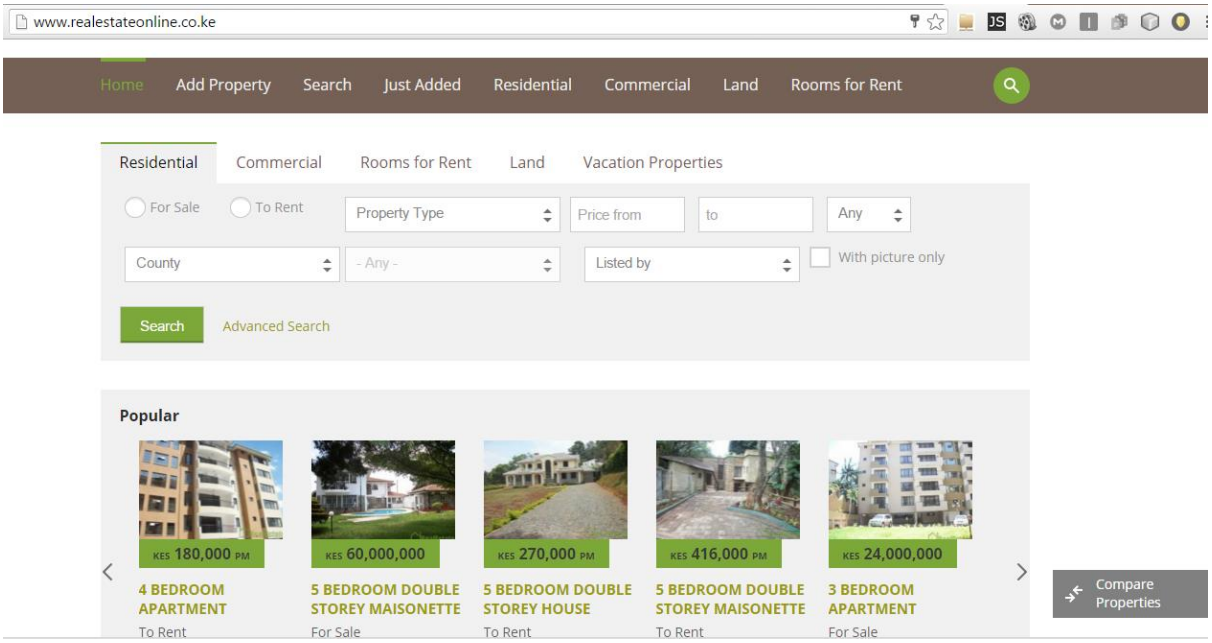

Figure 2.3: Real Estate Online website

Real estate online is an ecommerce website that posts ads on real estate. It has a nice user interface that is easy to navigate. There is a slide show that displays popular and featured ads and a search that is dynamic depending on desired property features.

Real estate online offers both free and featured ads. Under free package, the property is listed for two months with a maximum of 6 pictures. The featured package costs 750

Kenya Shillings (KES) (7.5 USD) and the property is listed for four months with a maximum of 12 pictures.

2.4.4. Property 24

Website URL: http://www.property24.co.ke/

Figure 2.4 shows the search page of property 24 website.

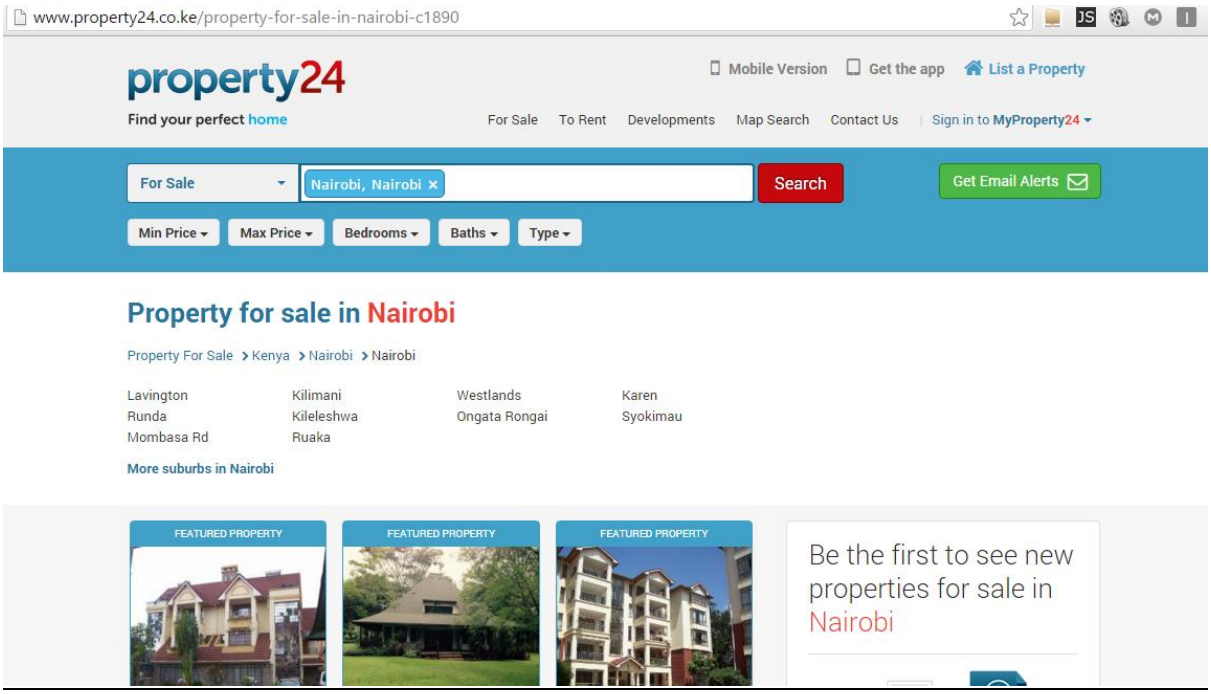

Figure 2.4: Property 24 website

Property 24 is an ecommerce website that posts ads on real estate. The web application can be accessed on computers as well as mobile devices. The mobile version of the web application has a different user interface that allows the user to search the property using locations as well as refining the search by including other parameters such as price range. The device detection feature directs the user to the respective domain depending on the device. Property 24 also has both iOS and Android mobile application that contains additional features such as finding property via map search. The user can save contacts of the property owner or manager directly to the address book if they are

using a mobile phone and contact them on the go. The web application has a refined search that enables the user to sort the searched property by price, date posted, property type and size of property.

Users can also set email alerts for properties from a certain region but it does not allow the user to provide the features of the property they want to receive the alerts on.

2.4.5. Olx

Website URL: http://www.olx.co.ke

Figure 2.5 shows the add property page of Olx website.

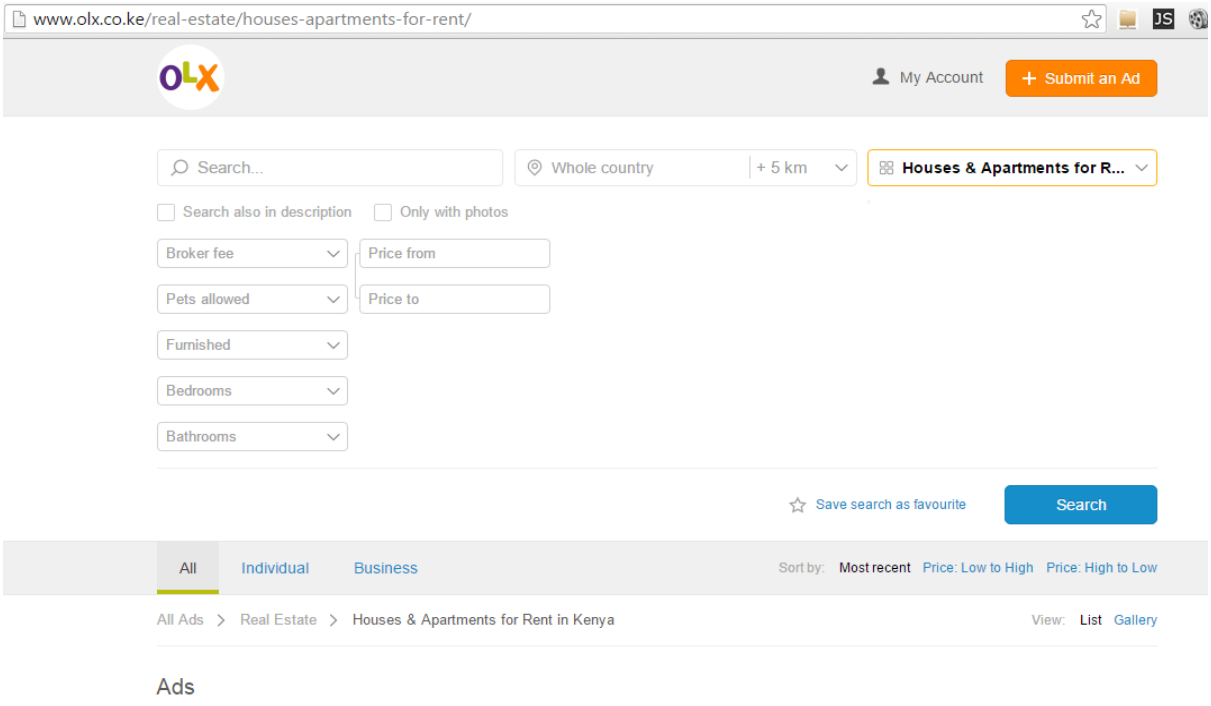

Figure 2.5: Olx website

Olx is an ecommerce website that posts ads of any goods or services but does not focus on real estate property. As a result, there are fewer property details hence the user cannot provide details such as amenities and security features of the properties.

2.4.6. Pigiame

Website URL: http://www.pigiame.co.ke/

Figure 2.6 shows the real estate category page of Pigiame website.

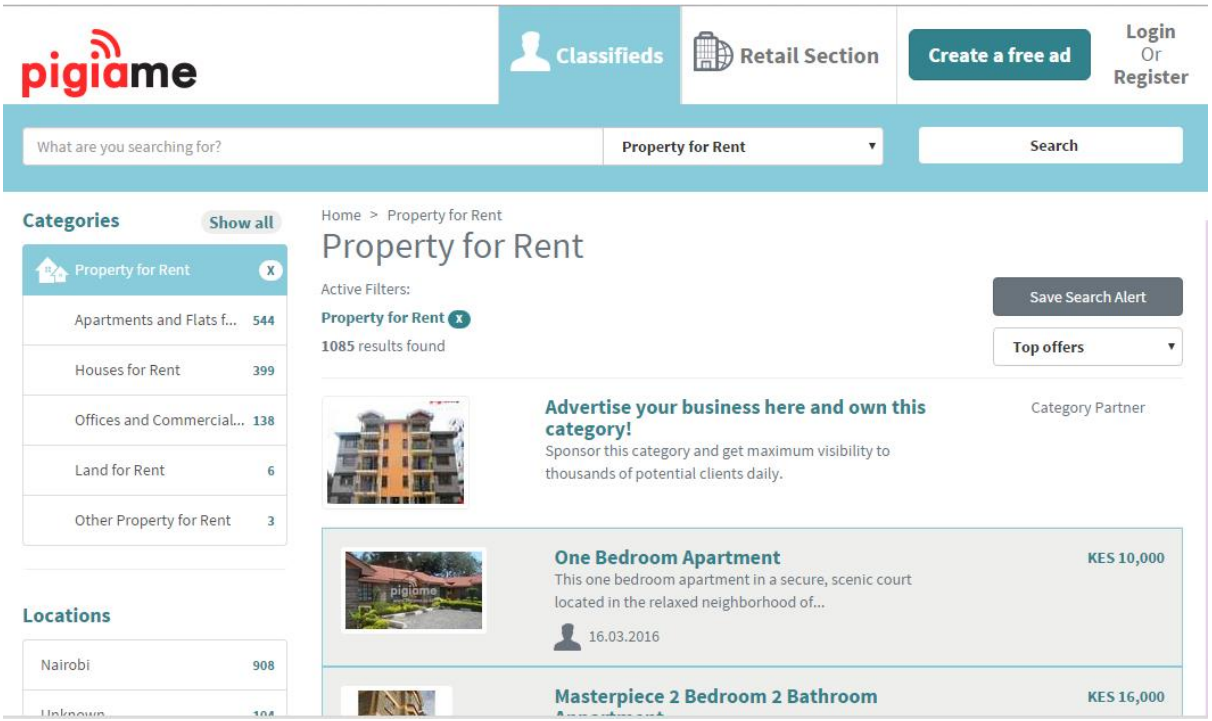

Figure 2.6: Pigiame website

Pigiame is an ecommerce website that posts ads of any goods or services but does not focus on real estate property. It has a friendly user interface that makes it easy to navigate. It sorts the properties by location, property type and price. The user can also order property by price and the date the property was added to the site.

2.5.Conclusion

The web applications listed above have the following advantages to the users

i. Property seekers can obtain and maintain first contact with the real estate owners and managers.

ii. Property owners and managers can advertise their property to a large number of people at a cheap rate.

iii. The property seeker can avoid unnecessary cost incurred during property search.

iv. They give flexibility to property managers and property seekers to list and search property at their comfort zone.

This real estate management application seeks to support the users in the following ways.

i. It will reduce the time taken while searching for property by the property seekers.

ii. It will give the property owners a platform to market their property

iii. The application will provide a connection between the property owners or managers and the prospective property seekers who are searching for properties.

iv. It will reduce the cost involved in paying the property brokers to search for a property.

v. The application will give the property seeker multiple options to choose from

vi. The property seekers will be able to subscribe to alerts by providing the features they want in a property. They will receive an alert via messages or emails.

vii. The users can search for a property based on specific property features.

In conclusion, by developing and deploying this application, I hope to ease the burden of property owners and property seekers in marketing their property and searching for property in Kenya respectively.

Chapter 3: Design and Architecture

3.1. Requirements

3.1.1. User Classes

There are three user classes

i. Property owners or managers- These are the people who own the properties and can add their property on the application either for sale or rent.
ii. Property seekers (general public)- These are individuals who are searching for properties to rent or buy
iii. System administrator -These are individuals who are responsible for maintaining the software and managing all the privileges in the system

3.1.2. User Requirements

i. The real estate management application will list all the properties added by the property managers or owners.
ii. The real estate management application will provide an interface to add property.
iii. The real estate management application will store the details of all the property managers and owners.
iv. The application will allow the property seekers to subscribe to alerts on property they are interested in.
v. The real estate management application will alert the users when a property they are interested in is uploaded.
vi. The application will allow users to register, login and to reset their password.
vii. The application will show the location of the property.
viii. The application will allow the users to edit and delete the posts that they add.

3.1.3. System Requirements

The property owners and managers can create a post of any property they want to market on the application by providing the required features of the property. The user will have to create an account before they can post any ad on the application. The registration involves providing the user details such as first name, last name, date of birth, gender, phone number, email address, username and password. All the details provided by the users are stored in the database. After registering, the user can log in using the username and password provided during registration. Authentication will be done to check if the credentials are correct then a session will be started and the user redirected to a page where they can add properties, view their posts and browse through all the properties on the site. In case the user forgets their password, the user can reset their password by clicking on the reset password link on the login page. The user is required to enter the email or phone number used during registration, after which a four-digit verification code is sent to the user if email or phone match is found in the database. The verification code is also saved to the database, to enhance authentication during password reset. The user enters the verification code and the new password and then clicks on save password button. If the code matches the code stored on the database, the user password is changed and then the verification code is deleted from the database.

The user can edit or delete the properties they have added on the site. A status of the property is also displayed (active/ disabled). Free ads will become inactive after two months while featured ads will become inactive after three months. Disabled ads will not be visible to the property seekers. The user can change the status of the property to list it back on the site by clicking on activate button. This reduces the struggle of the property owner to add the details of the same property back to the site if it gets deactivated before it is bought or rented.

The properties will be displayed with the most recently added property at the top. Each post will contain all the features of the property. The featured posts will be ranked higher than the free posts during the search. When the user clicks on the post, all the details of the property will be displayed including the contact information for the property owner or manager. The property seeker can send an email to the property owner from the site.

The property seekers who are registered members of the site can create a customized alert with the features of a property that they want. When a property that matches those features is uploaded on the site, the user gets a message or email notification to check it up on the site. The user can also unsubscribe for the alert to stop the notifications. The alerts are automatically deactivated after a period of three months from the day they were created. The user does not continue to receive the alerts when they are deactivated. A user can also deactivate the alert manually by selecting the alert and clicking on the deactivate button. The user can also activate previously deactivated alerts by clicking on the activate button on the list of subscribed alerts.

3.1.4. Functional Requirements

To add a property on the site, the user will provide the location details of the property, general features of the property and the advance features of the property (which are optional). The user will then choose between a free and featured plan. For the free plan, the property is listed normally for a period of two months. A maximum of four pictures can be added under this plan. The featured plan costs 500 Kenya Shillings (KES) (5 USD) and the property is listed on the site for three months and the user can upload a maximum of six pictures. Under this plan, the ad will get more views since it is ranked on top during the searches and normal listing.

The property owners can post properties on the site. The property owners can also edit the details of a property they have added to the site.

The property owners can also delete their properties.

A property seeker can subscribe for email or text message alerts.

The users can also search and browse through the properties in the application.

3.1.5. Non Functional Requirements

3.1.5.1.Usability and learnability

The application should be easy for the user to learn and use even when they are using it for the first time. The interface design should be simple with a clear navigation menu and icons on input fields.

3.1.5.2.Security Requirements

The system should be secure and the communication between the clients and the server should be protected from attacks.

During registration, the user is expected to provide personal details such as email and phone number. The password provided during registration is encrypted using MD5 before storage in the database. The SQL queries are written as prepared statements to prevent MySQL injection.

The login will authenticate the users to prevent unauthorized access to pages such as profile details of registered users. This will prevent random users from editing or deleting property that they did not add to the system.

3.1.5.3.Scalability

The system should be able to scale and support large number of active users at a given point in time. The server that hosts the software should be up and running at all times.

3.1.5.4.Dependability

The system should be able to deliver services to the users at any point in time. It should be able to deliver the minimal services to the users in case of any failures. Since this software is expected to serve large number of users, it should be easy to maintain without stopping the services of the software.

3.1.6. System Constraints

i. The user cannot access some pages that are not within their access rights. This is to enforce security of the system such that only authorized users can have access.eg: deleting and editing properties.

ii. The users will not be able to use all the functionalities of the system. Only administrators will have the rights to control every functionality of the system.

iii. The application cannot be accessed without network connection

iv. The application cannot be accessed on phones that do not have internet access capabilities.

3.2.System Modelling

3.2.1. Activity Diagram

Figure 3.1 below shows an activity diagram for a property seeker while searching for a property on the website.

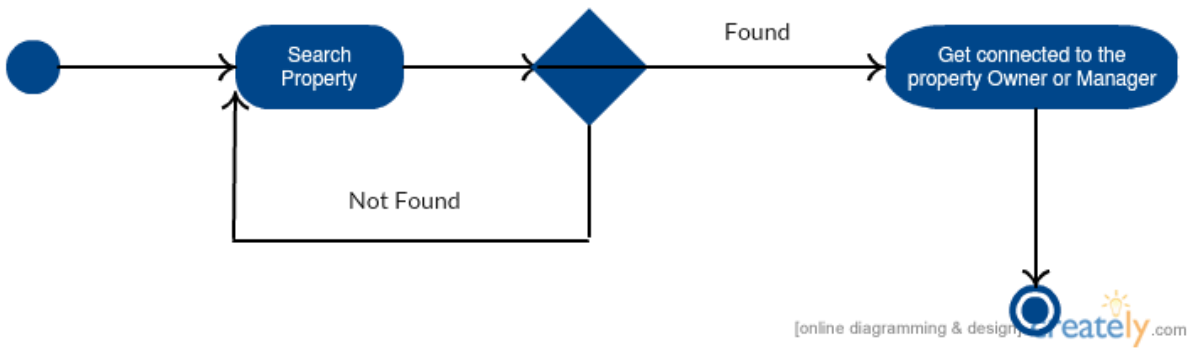

Figure 3.1: Activity diagram

3.2.2. Use Case Diagram

Figure 3.2 below shows the use case diagram for property owners or managers and property seekers.

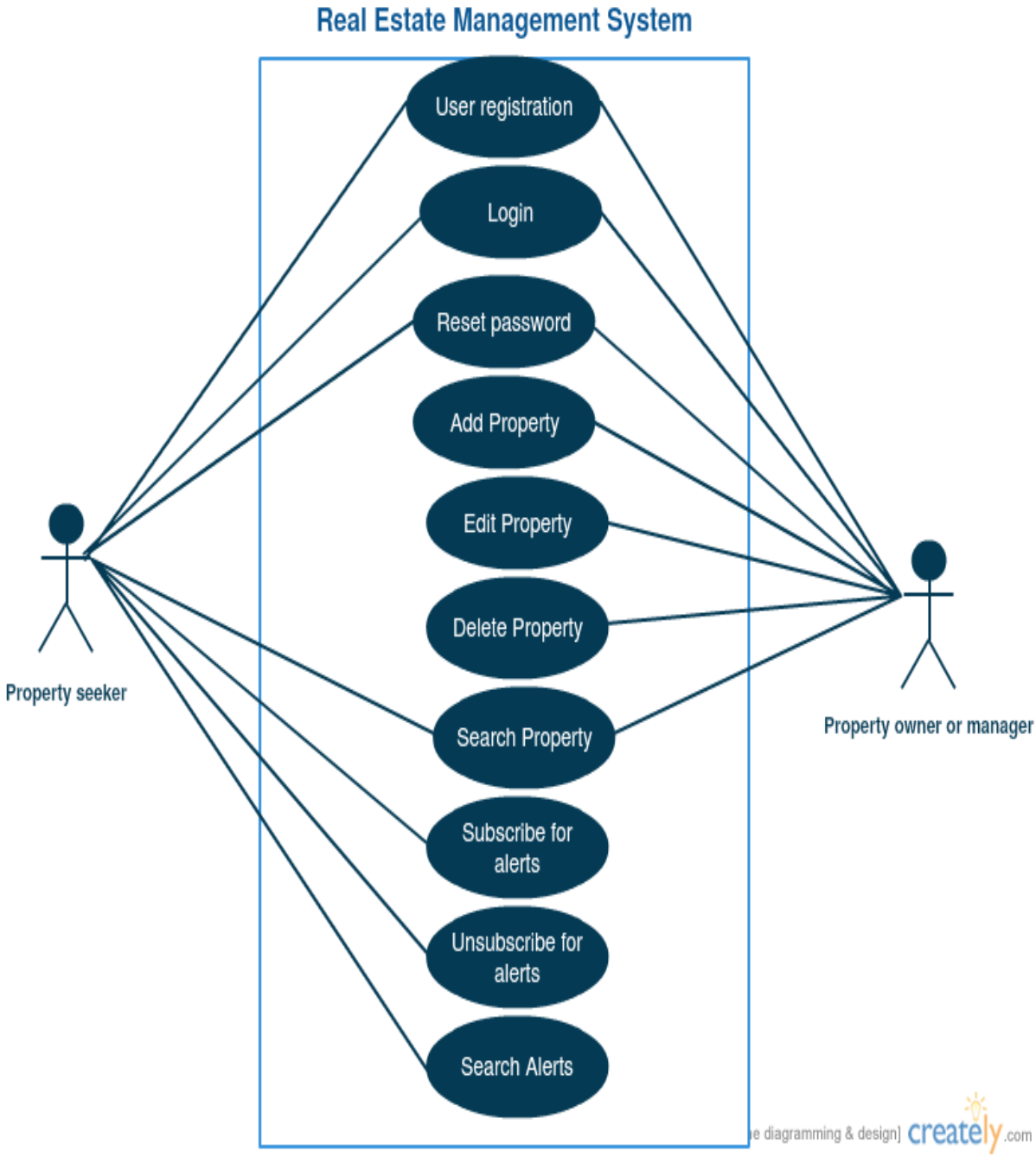

Figure 3.2: Use case diagram 1 (property owners and seekers)

Figure 3.3 below shows the use case diagram for the system administrator.

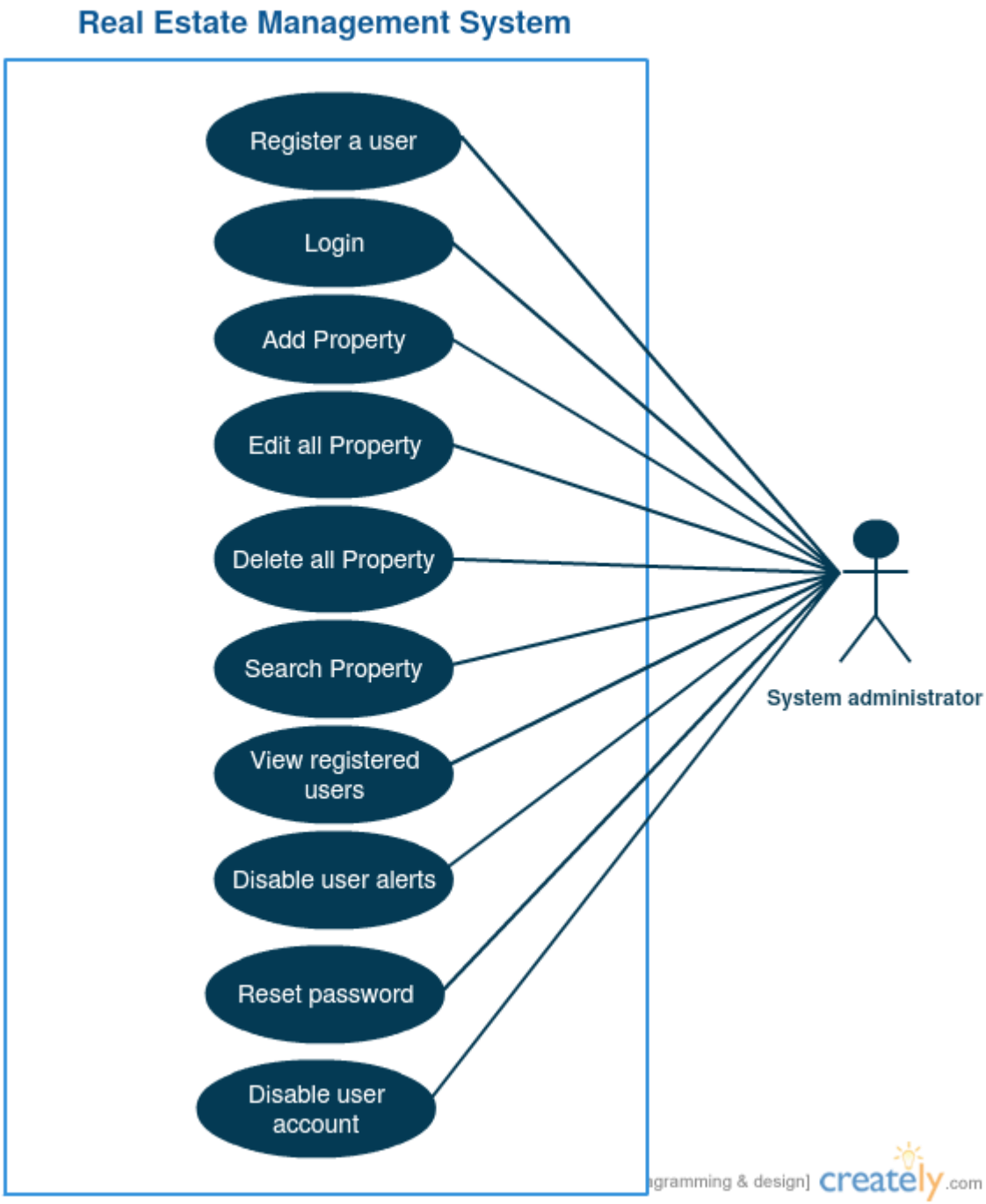

Figure 3.3: Use case diagram 2 (administrator)

3.2.3. Sequence Diagram

Figure 3.4 below shows a sequence diagram when a property owner is adding a new property on the website.

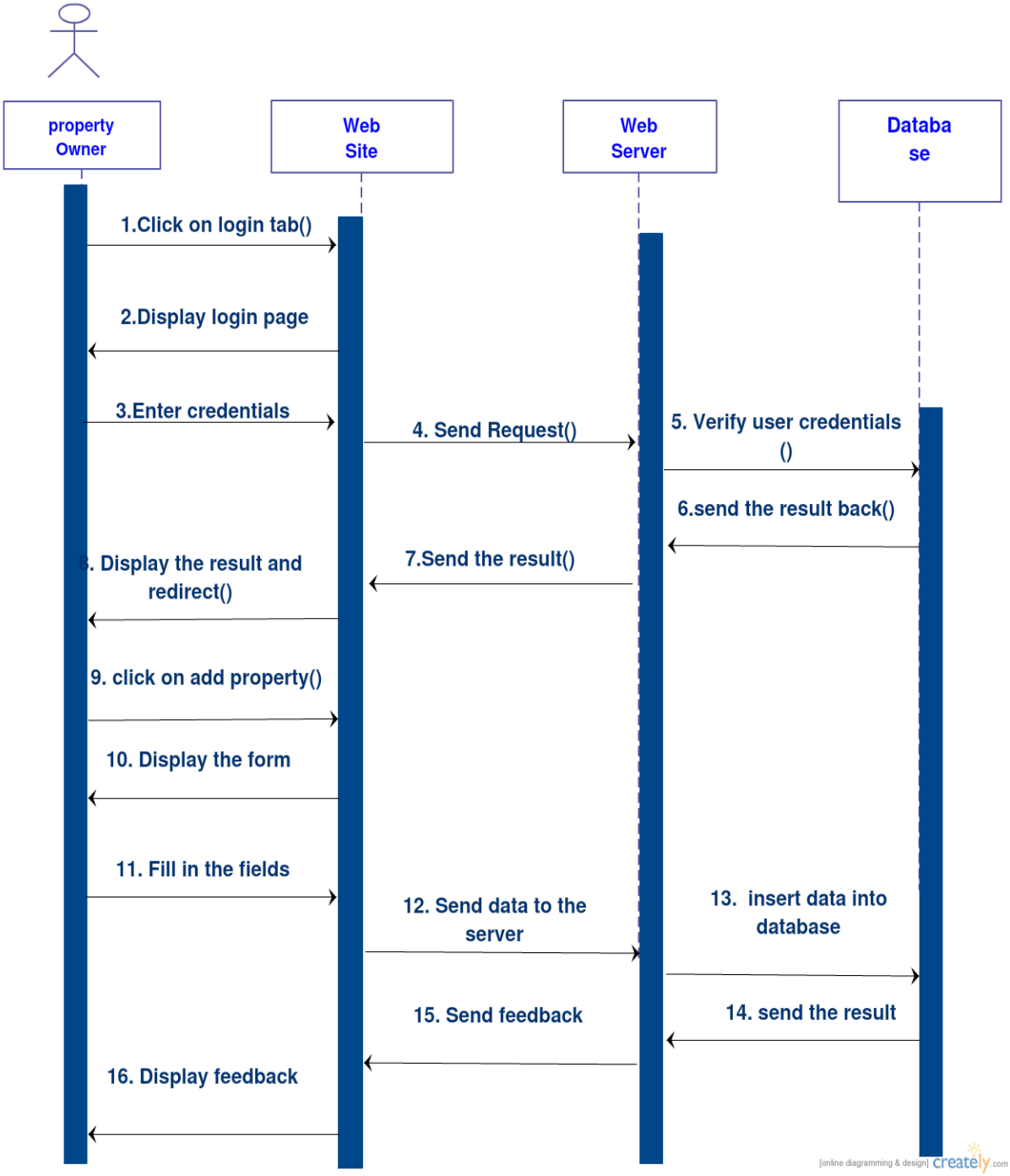

Figure 3.4: Add property sequence diagram

Figure 3.5 below is a sequence diagram when a property seeker is subscribing for a message or email alert in the application

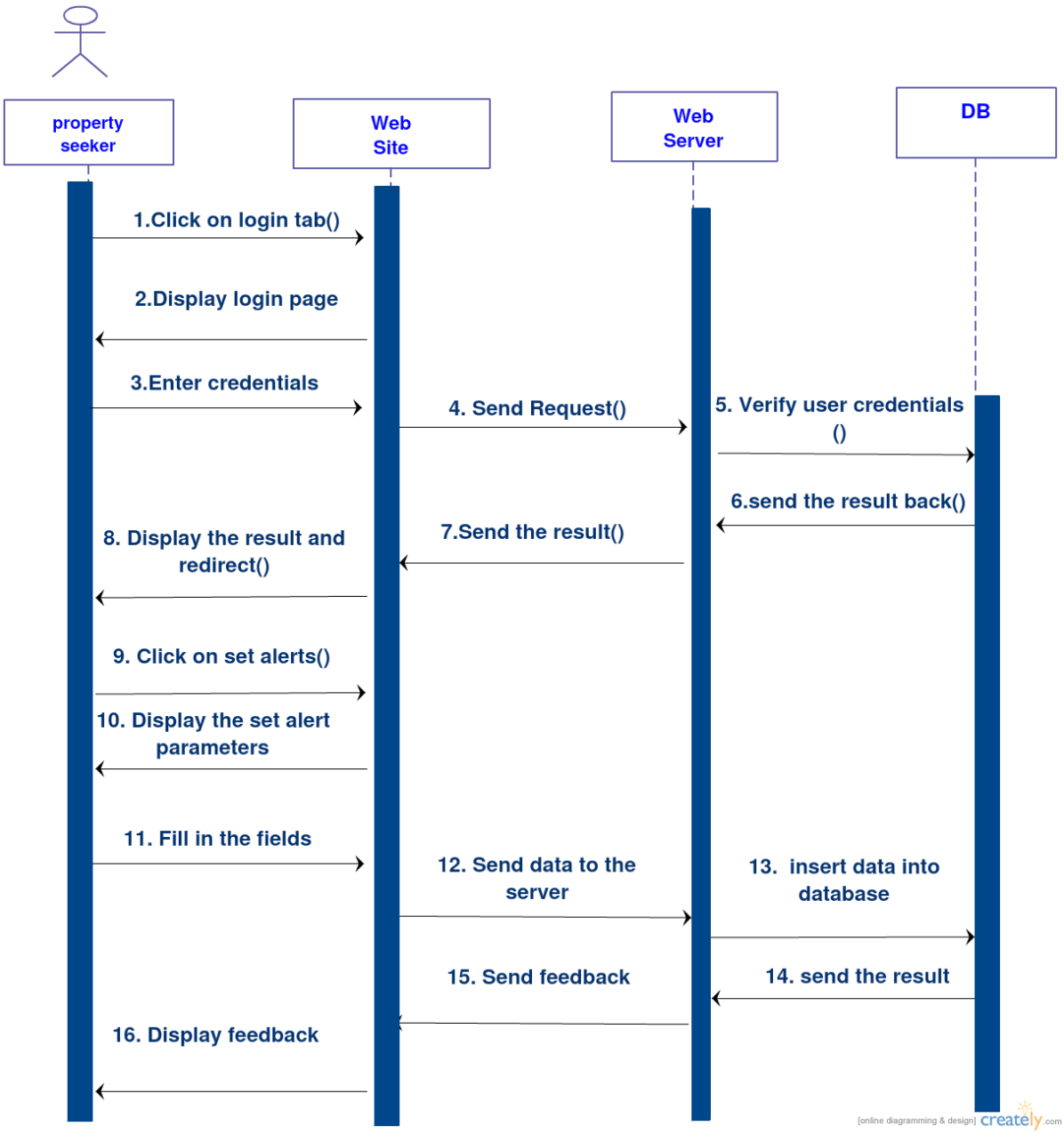

Figure 3.5: Subscribe alerts sequence diagram

Figure 3.6 below is a sequence diagram when a property seeker is searching for a property on the website

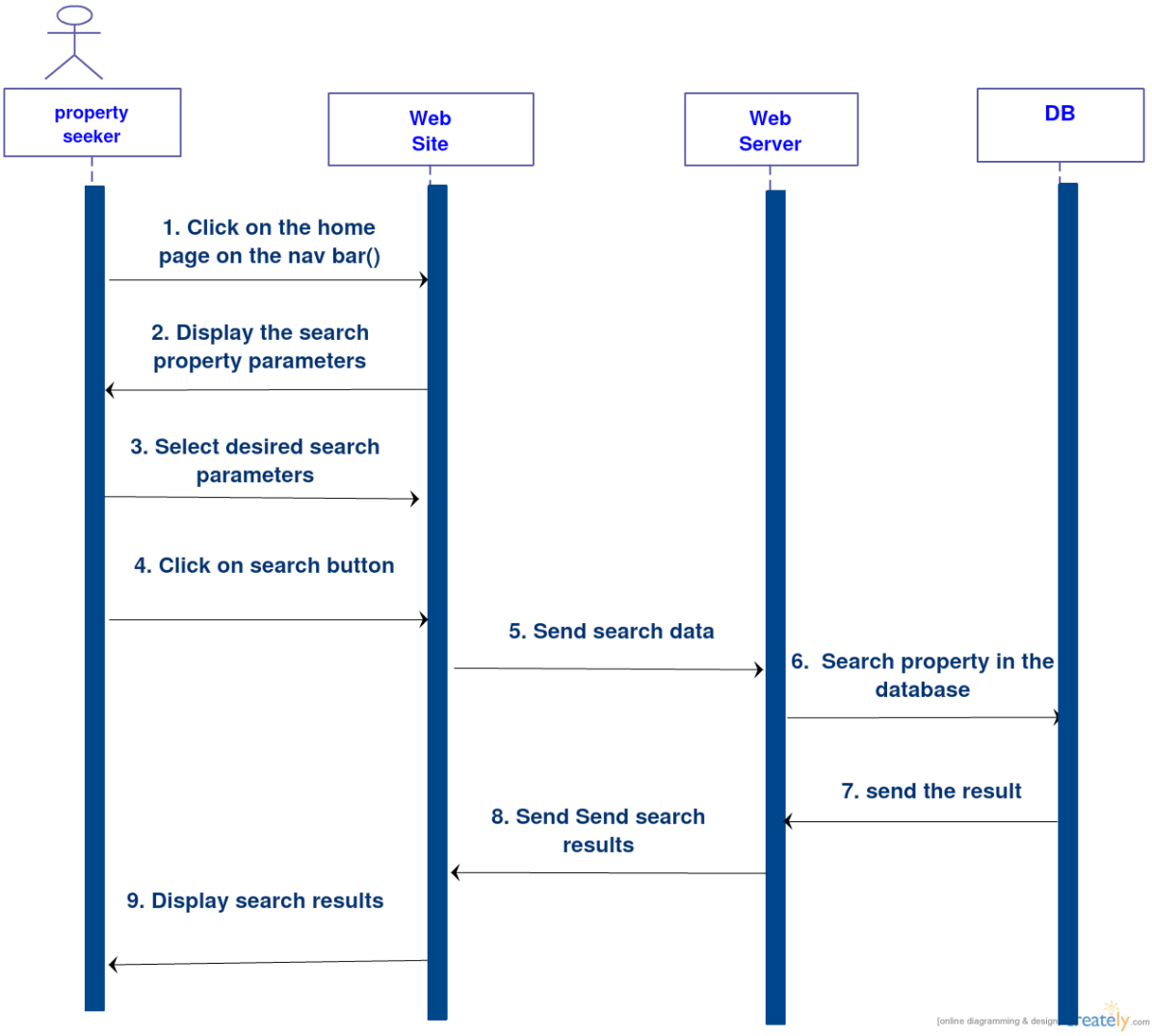

Figure 3.6: Search property sequence diagram

3.3.Architecture Pattern

The application is implemented with model view and controller (MVC) design pattern. This is a design pattern that separates the application's presentation layer that displays data on the user interface from the data that is being presented. This provides isolation of the code such that the business logic can be changed without affecting any other layer. I chose this pattern because I am concerned about maintainability of the

system hence the MVC model is appropriate in that it is easy to debug and fix the errors that occur in the code. It is also easier to test code that has been written in MVC since the classes can be tested independently.

Since the software involves numerous interaction with the database, it is designed using AJAX, which is a client side script that communicates with the server without need for page reload. This improves the performance, speed and usability of the application. Ajax makes use of asynchronous calls to the web server allowing the client browser to continue with other tasks instead of waiting for all data to arrive.

Figure 3.7 below shows the architectural pattern used (MVC)

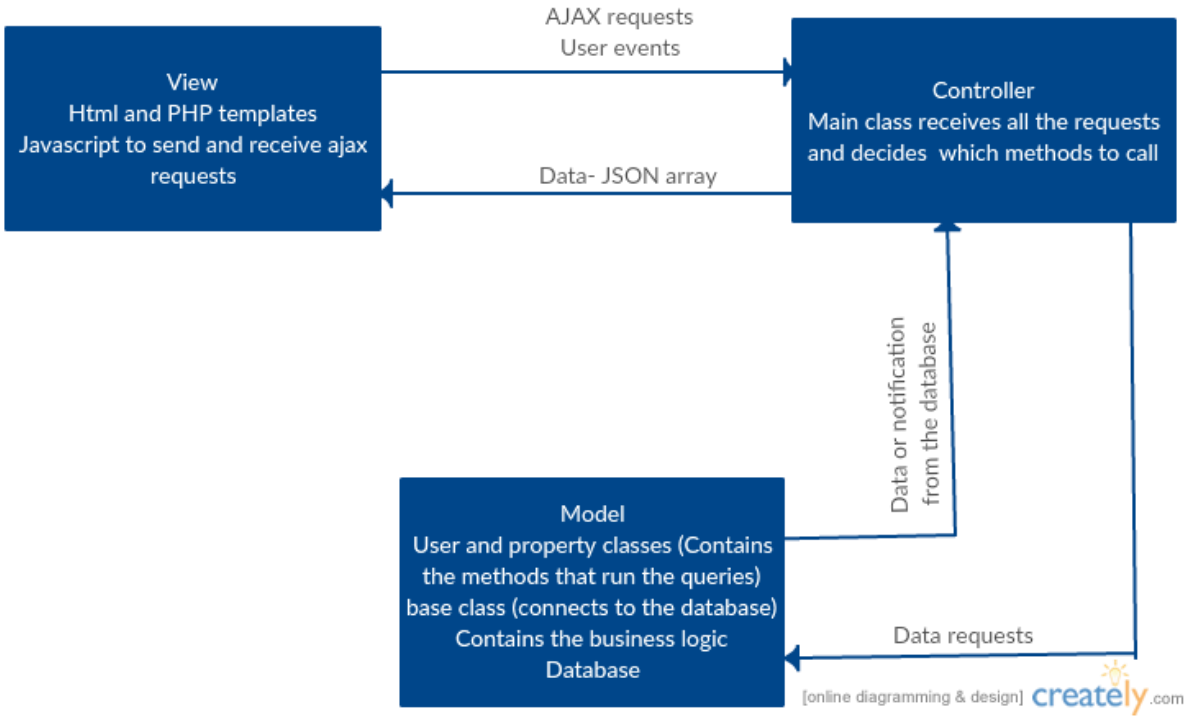

Figure 3.7: MVC architectural pattern

Chapter 4: Implementation

4.1.Technologies Used for Development

The user interface of the software is built with Material Design, which was introduced by Google in 2014 and is based on ‘Flat’ design. Material Design is flexible and tries to bring together motion and interaction design. This makes the application highly responsive hence making it suitable for different screen sizes while maintaining a good look and feel. I choose Materialize CSS framework to maintain consistency in the user interface and also due to its low learning curve that reduced the development time. HTML and Java Script have been used to increase interactivity on the site as well as to handle AJAX requests. AJAX improves the responsiveness of the software by getting data to and from the database without reloading the page.

The software uses a MySQL database to store data. The backend of the software is written in PHP. PHP is a general purpose scripting language that is mostly suitable for server side web development. I choose PHP because it is open source hence it is free and has a large support group with references and guidelines available hence reducing the development time. PHP is also cross-platform and therefore can work excellently with any operating system.

By taking advantage of the MVC model, the code is isolated and one tier can be changed without affecting the other tiers. The integrated development environments (IDEs) that were used to program the software are Sublime Text and Brackets.

4.2.Classes and Methods

Table 4.1 shows methods that are in the three main PHP backend classes.

Table 4.1: Backend class methods

Method Name	**Function**
User Class	
login()	Validates username and password provided by the user to allow login
signUp()	Saves user details to the database during registration
editUserType()	Changes the user type from admin to regular or vice vasa
getUserDetail()	Fetches user details for display on their profile
sendPasswordResetCode()	Generates a random four digit code and sends it to the user as well as save it to the database
changeUserPassword()	Change user password with a given verification code
deleteResetCode()	Deletes the reset code after password reset
Post Class	
addPopertyBasics ()	Inserts generic property features to the database
addPopertyFeatures()	Save property features for houses
addLandFeatures()	Adds land features to the database
addPropertyPictures()	Adds property pictures to the database
getLastProrpertyId ()	Returns the last property Id of a property that has just been added
getMyProperty()	Fetches all properties that have been added by a specific user
fetchHomePageProperty()	Fetches featured posts for display on the home page
fetchHouses()	Fetches all houses from the database

fetchLands()	Fetches all lands from the database
fetchAllSearchProperty()	Fetches properties for display on the search page
propertyCountPerUser()	Gets count of properties that have been added by a user for display on the profile
deletePost()	Deletes a property given its id
deletePictures ()	Delete pictures of a property with a given id
deleteLand()	Delete land details of a property with a given id
deletePropertyFeatures()	Delete features of a property with a given id
refinedHouseSearch()	Searches for houses with given parameters in the database
refinedLandSearch()	Searches for land with given parameters in the database
refinedPropertySearch()	Searches for properties in the database with given search parameters
refinedCountySearch()	Searches for all the properties in a county in the database
refinedSaleRentSearch()	Searches for all properties for rent or in the database
refinedCountyRentSearch()	Searches for all properties for rent or sale in a given county in the database
refinedAllCountySearch()	Searches for properties from a given county in the database
refinedCountyAllSubCountySearch()	Searches for houses in the database
Alert Class	
setAnAlertEmail ()	Subscribes for an email alerts with given parameters
setAnAlertPhone ()	Subscribes for message alerts with given parameters
alertSearch ()	Fetches all the alert that have been added by a user
unsubscribeAlert ()	Deactivates an alert with a given alert id

alertCountPerUser()	Gets the count of alerts that have been made by a user for display on the profile
Base Class	
connect()	Establishes connection to the database
log_error()	Gets the details of the errors that occur
fetch()	Fetches the results of the query
query()	Executes the query
get_num_rows()	Returns the number of row in a table

4.3.Entity Relational Diagram for Database

Figure 4.2 below shows the ER diagram for system users and alerts tables.

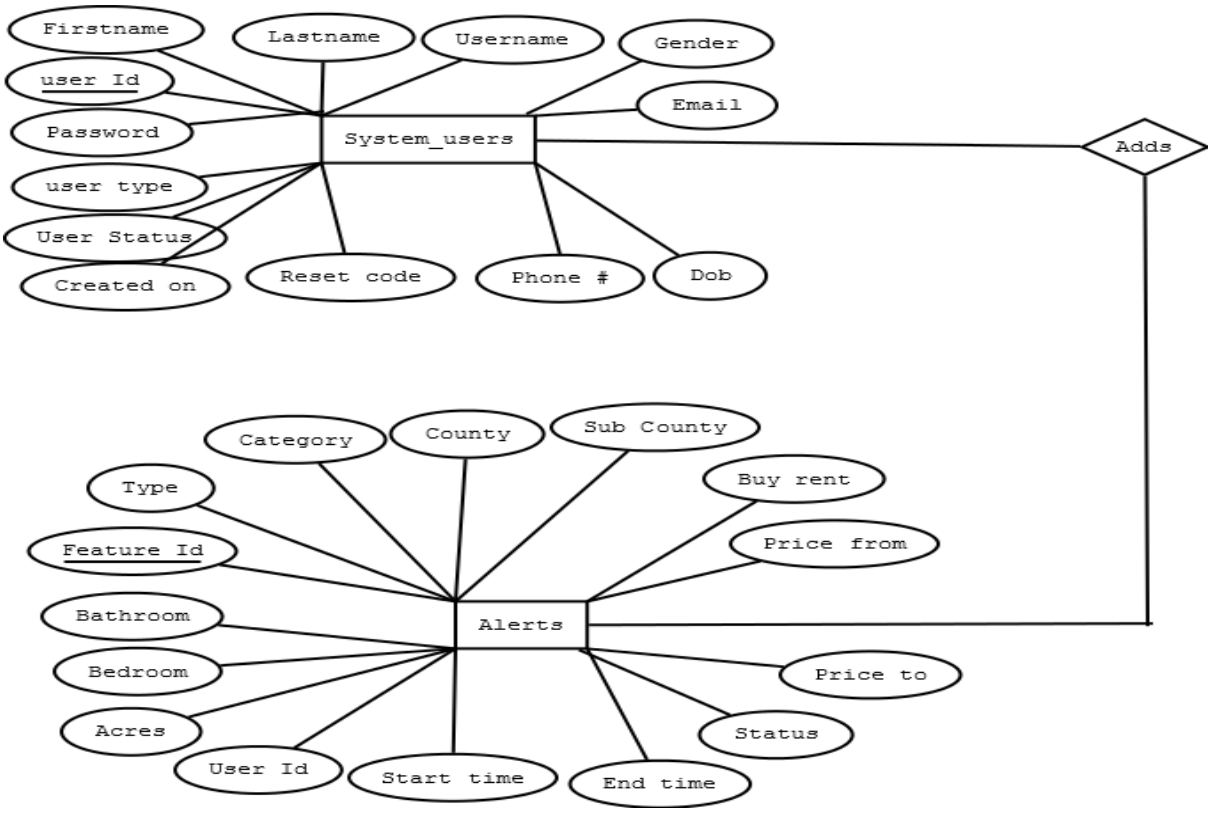

Figure 4.2: ER diagram 1 (system users, alerts)

Figure 4.3 below shows the ER diagram for the system users, property, property features, land, and pictures tables.

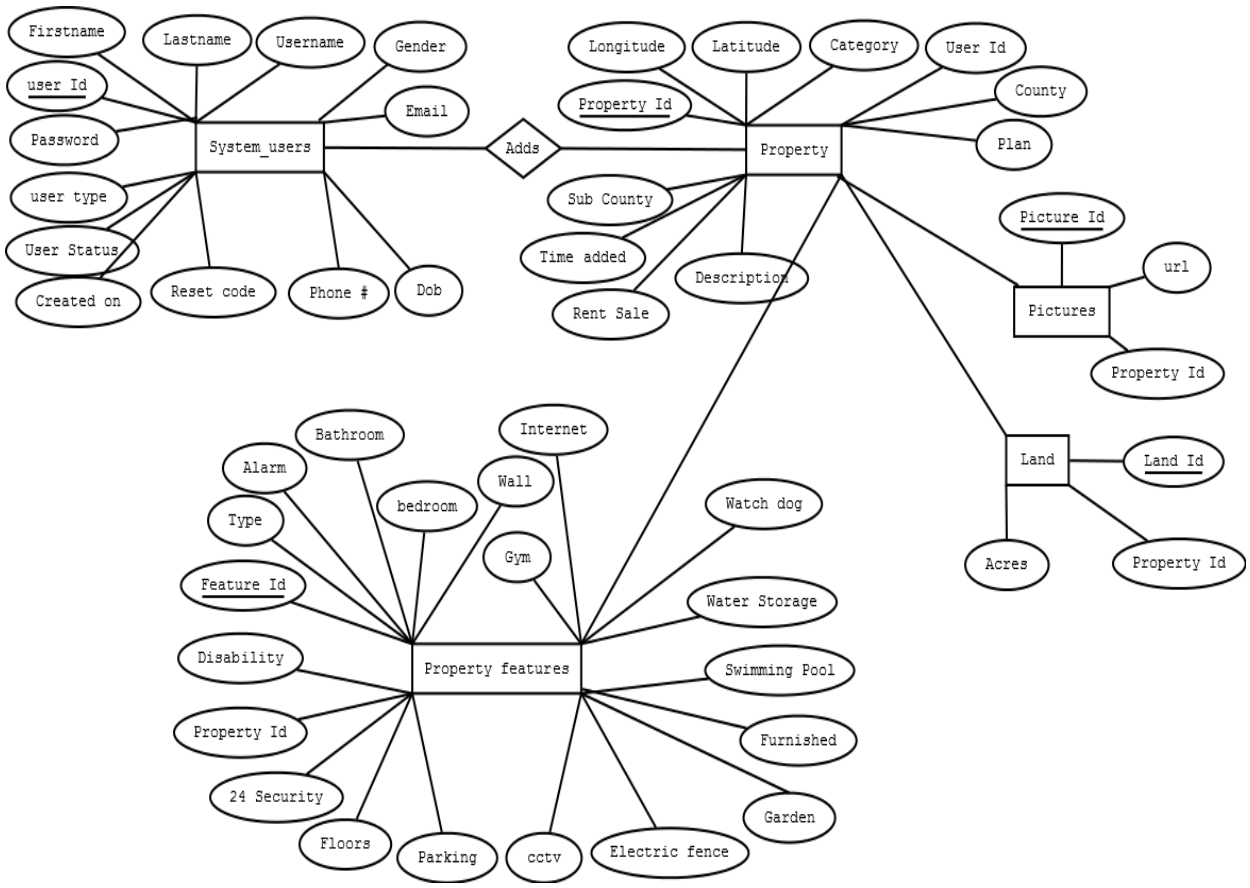

Figure 4.3: ER diagram 2 (users, property, property features, land, pictures)

The software is implemented with MySQL database and consists of six tables i.e. property, property features, lands, pictures, alerts and users tables.

The property table consist of generic details of the various properties while the users table store user details and credentials. The property features table store the property features of houses while the land table store the details of the land. The pictures table store the pictures of the added properties. The alert table stores all the subscribed alerts.

Below is explanation of steps to follow when performing some of the major tasks on the application.

i. **Add property:** The user clicks on add property on the navigation bar, a page loads with the fields that are required to add a property. Figure 4.4 below shows the property location details that need to be provided when adding a property on the website.

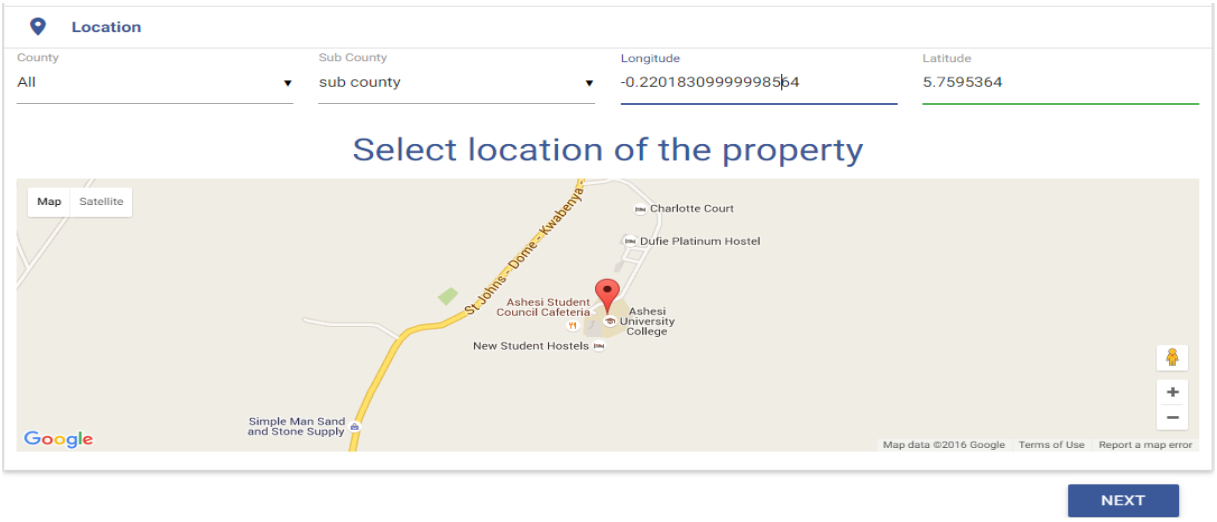

Figure 4.4: Add property (location)

Figure 4.5 below shows the property details that need to be provided when adding all properties on the website.

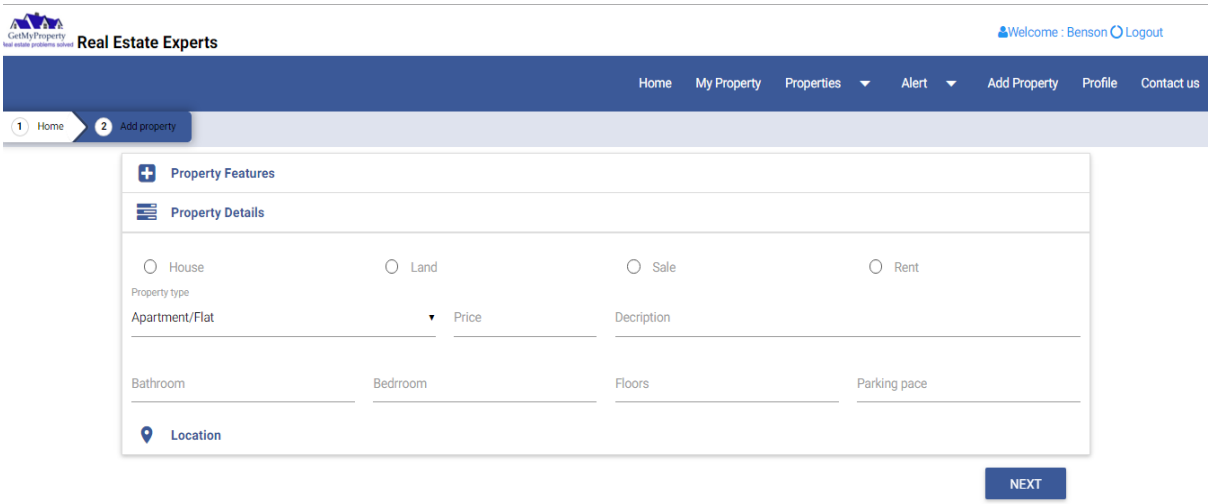

Figure 4.5: Add property (details)

Figure 4.6 below shows additional property features

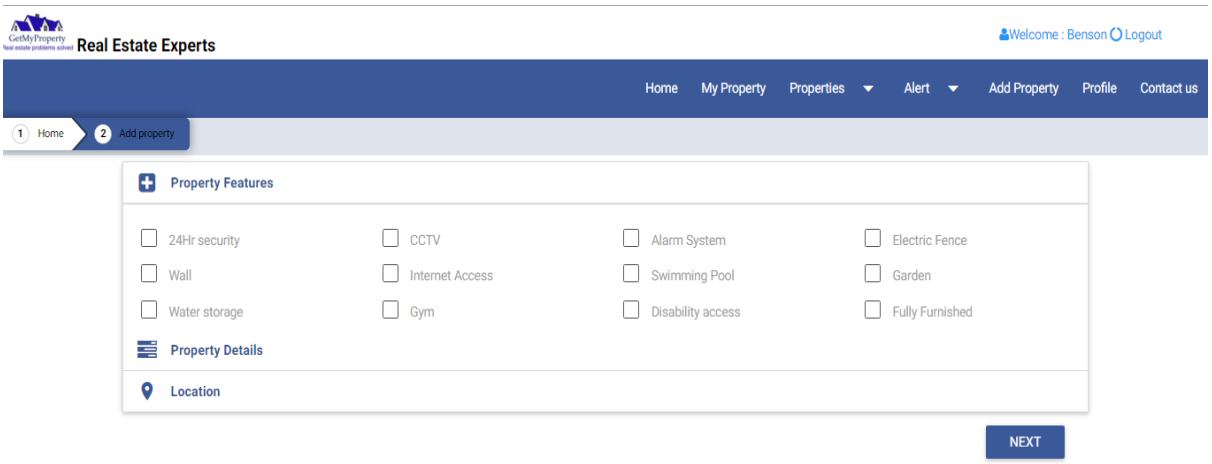

Figure 4.6: Add property (features)

If the user has not logged in, they are redirected to the login page for login before proceeding with the task as shown in figure 4.7 below.

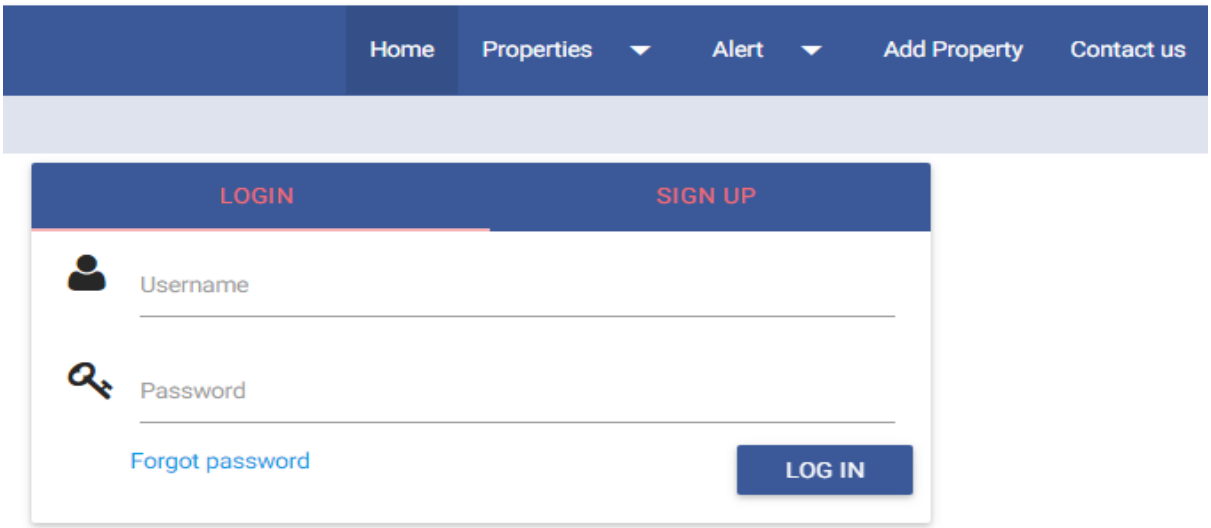

Figure 4.7: User login

The next button on the page will load a page that prompts the user to choose if the ad is featured or free as shown in figure 4.8 below.

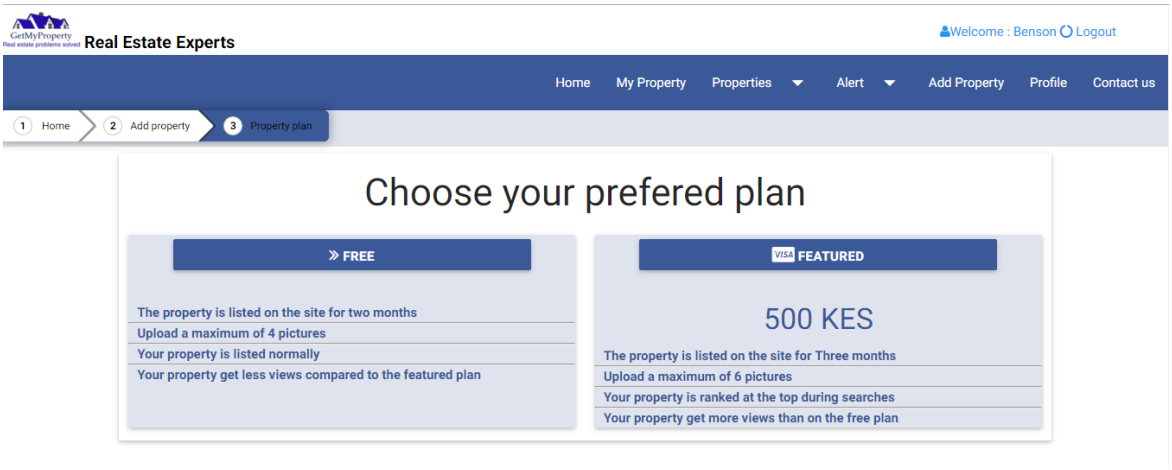

Figure 4.8: Preferred property plan

The user will then upload pictures depending on the plan they choose as shown in figure 4.9 below.

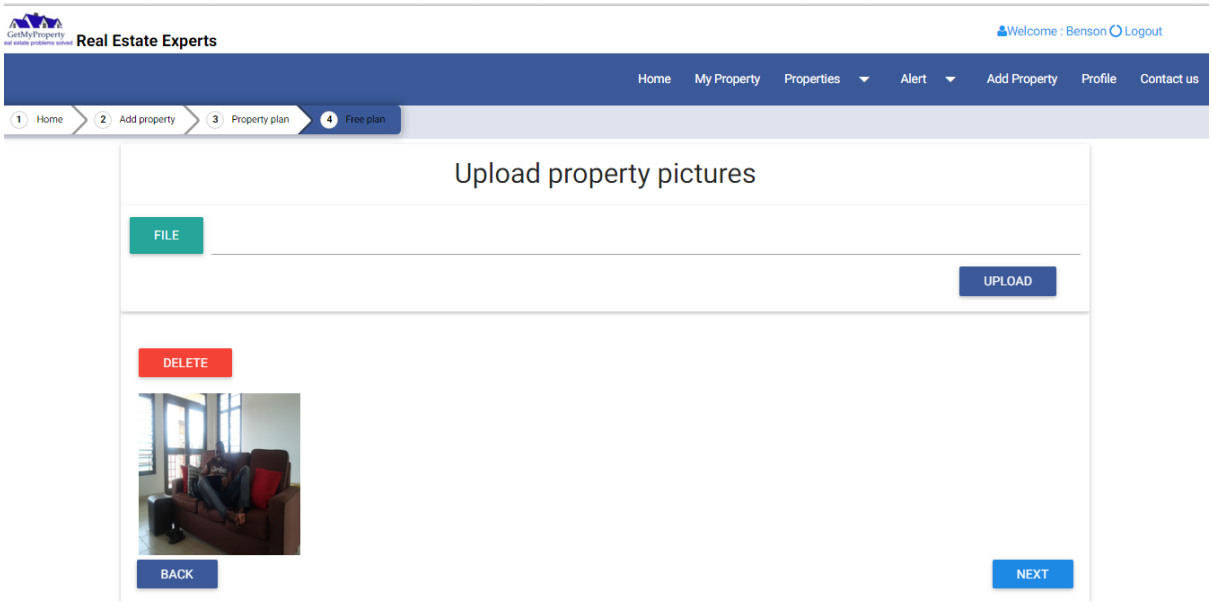

Figure 4.9: Upload pictures

When the next button is clicked, a page loads showing a preview of the property features together with the uploaded pictures as shown in figure 4.10 below.

Figure 4.10: Review post

The user can click on the back button to go to the previous pages to edit the property details. The details of the property provided on all the pages are saved on session variables on the server. The add button when clicked fetches all the values from the session variables and then triggers the add property function which executes SQL query to insert the provided data into the database. When the insert query is successful, the session variables are reset and the user is redirected to the page that displays the property they have added. Table 4.11 below shows a summary of the fields that need to be provided while adding a property.

Table 4.11: Property fields

1	Generic property features	• Property Id (Auto increment number) • County • Sub county • Longitude • Latitude • Property category • For sale or rent • Price • Description

		• Time added (Timestamp) • User Id (Foreign Key) • plan
2	House property features	• Property feature Id - (Auto increment number) • Property Id (Foreign Key) • Property type • Bathroom • Bedroom • Floors • Parking space • 24 HR security • CCTV • Alarm system • Watch dog • Electric Fence • Wall • Internet access • Swimming pool • Garden • Water storage • Gym • Fully furnished
3	Land features	• Land Id (Auto increment number) • Property Id (Foreign key) • Acres
4	Property pictures	• Picture Id (Auto increment number) • Property Id (Foreign key) • Picture Url

ii. **View property**: The user will be able to view and search the properties they have posted on the application. The user has to log in to get access to this page. The user can also view the properties that have been posted by other managers on the site even without login in. Figure 4.12 below shows the properties that have been added by a user.

Figure 4.12: User properties

iii. **Edit property**: The user searches for the property to edit or update from the list of properties they have added. When the property to edit is selected, all the necessary fields about the property will load on the forms for editing. The update button saves the changes made on the forms to the database. A user has to log in to edit the property and can only edit the properties that they have posted.

iv. **Delete property**: The delete post feature is used to delete an existing property in the database. Besides the property, there is a delete icon that when clicked triggers the delete SQL query to remove the record from the database. The query deletes the property from the property table first and then deletes the corresponding records from the picture, property features or land tables. A user can only delete the properties they have posted. Figure 4.13 below shows the delete property feature.

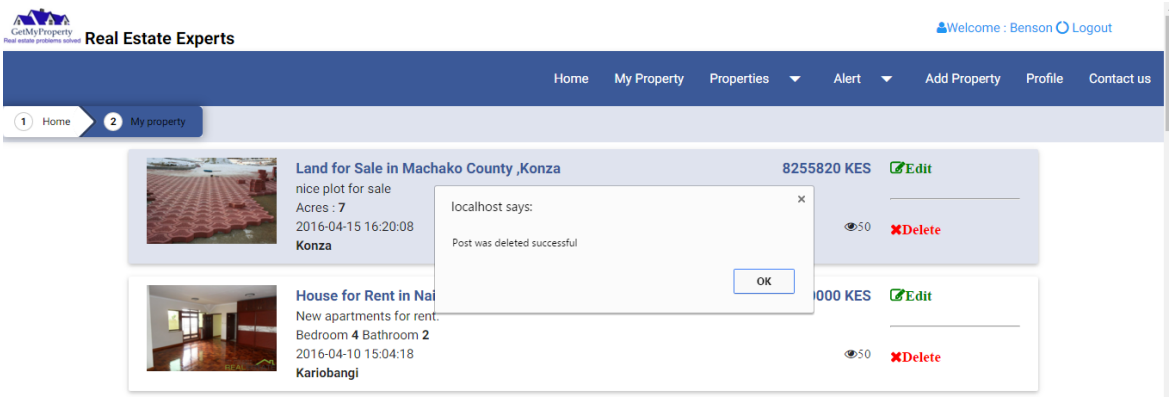

Figure 4.13: Delete property

v. **Search property:** This feature looks for existing records in the database depending on the search parameters provided. The user can combine multiple parameters to personalise the search and give results that match user interest. The user can also change the search parameters and click on the search button. The search query is dynamic and appends parts of the query depending on the search parameters provided. The number of properties and pages returned is displayed at the top left of the page. The pagination at the bottom of the page helps the user browse through the properties that are fetched during the search. Figure 4.14 below shows the search property page.

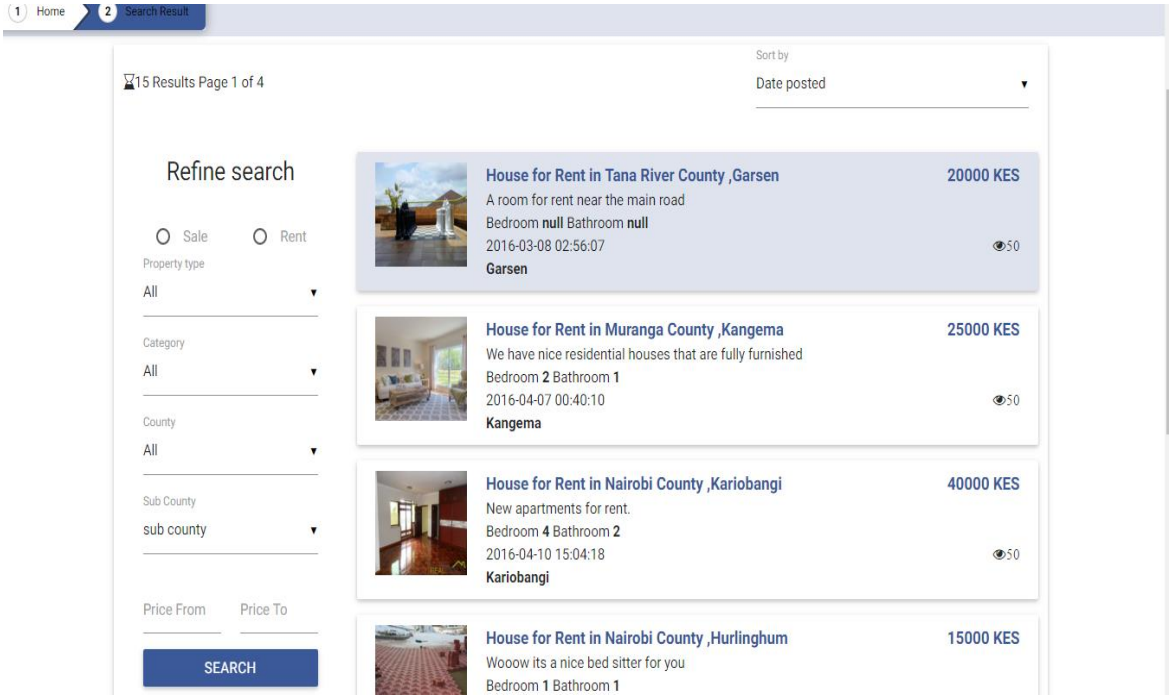

Figure 4.14: Search property

vi. **Property Details:** The user clicks on the heading of the property they want to view more details. A page with the property details and the property owner or manager information is loaded. The user can see the contact information of the property owner and also send them an email. The user can also browse the property pictures on the slideshow by clicking on the pictures. The general details, security details and the amenities are presented below property pictures. The user can also report the ad to the administrator on the basis that it is not appropriate. Figure 4.15 below shows the property details page.

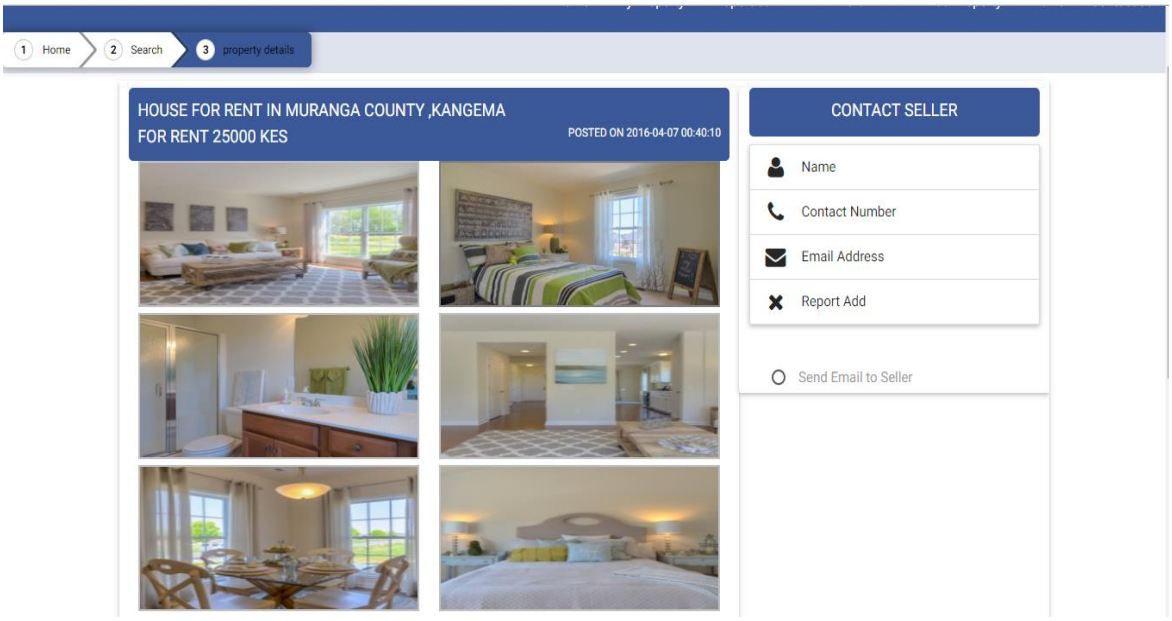

Figure 4.15: Property details page

Below the property details a maximum of three similar properties are listed. This is to improve the user experience by giving them more options of what they are searching for. Figure 4.16 below shows the similar properties listed below property details.

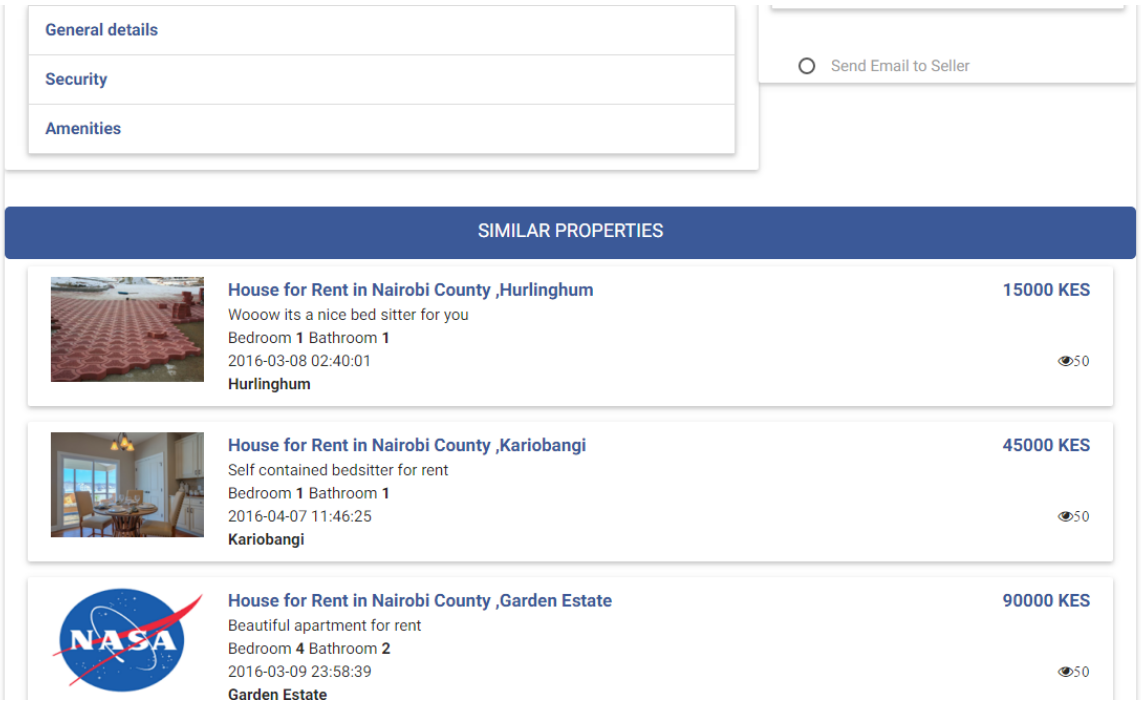

Figure 4.16: Similar properties

vii. **User Registration:** The user clicks on the login link at the top right corner of the page to load the registration page. Click on the sign up tab to display the registration form. The user provides the following information. Figure 4.17 below shows the registration form.

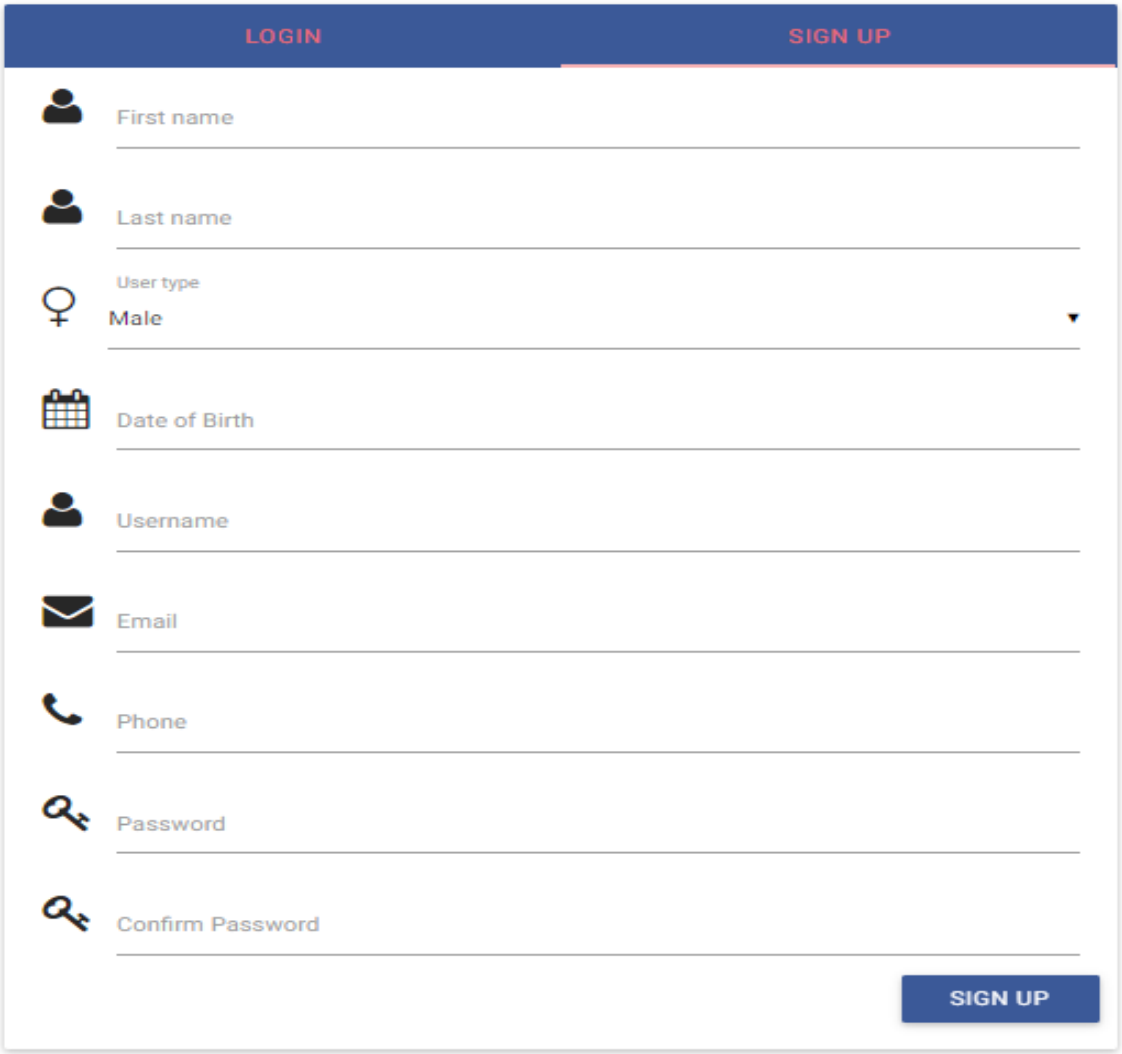

Figure 4.17: User registration

Clicking on the sign up button saves the user details in the database. The email field is optional while all the other fields are mandatory.

viii. **Login:** The user clicks on the login link at the top right corner of the page to load the login page. The user provides the username and password that were provided

during user registration. When user authentication is successful, the user is redirected to their respective pages depending on the user type. The admin has a different view from a regular user. Figure 4.18 below shows the user login form.

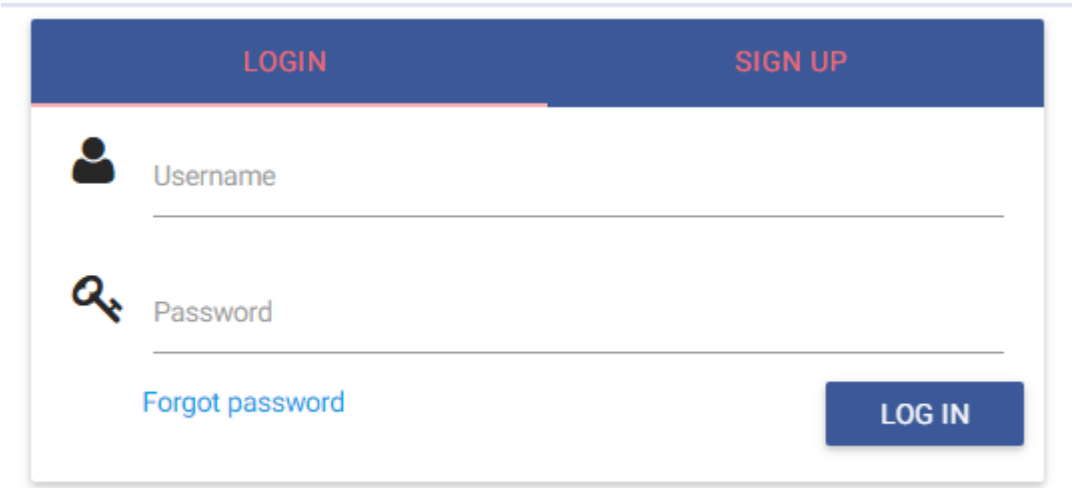

Figure 4.18: User Login

4.4.Implementation Challenges

The biggest challenge during the implementation was storing on session variables all the data provided by the user while adding a property. I achieved this by collecting all the data using JavaScript and sending it to the PHP page using AJAX I then stored all the variables in a session array that was encoded to a json object and sent back to the view page for display.

Another challenge was displaying uploaded images before saving the URL in the database. I was able to display the image and also allow the user to delete any image before adding a property.

Chapter 5: Testing

5.1. Introduction

Testing is the process of verifying whether the developed system meets the different requirements (functional and non-functional) that were specified in the beginning of the project. This verification is done mainly to ensure that the system can perform the tasks it was developed to perform. Software testing is also done to validate if the software meets the expectation of the clients or its users (Sommerville, 2011).

I performed two major categories of testing;

i. Development testing
ii. User testing.

This chapter seeks to explain the types of testing that were conducted and the results.

5.2. Development Testing

Development testing includes all the testing activities that are carried out by the development team while creating the system (Sommerville, 2011). This testing is conducted to find bugs in the software and hence it is interweaved with the debugging process (Sommerville, 2011).

There are three levels of development testing.

i. Unit testing.
ii. Component testing.
iii. System testing.

5.2.1. Unit testing

Unit testing is the process of testing the methods and object classes that have been used to develop the software. To perform testing for my functions, I used PHPUnit.

PHPUnit is a testing framework for PHP that is used to test PHP methods and classes. I developed one testing class (testClass.php) and wrote an algorithm that tested the functions in the user, post and alerts classes. The figures below describe the steps and result of testing the getUserDetail($id) method in the user class. getUserDetail($id) method fetches the details of a user given their user id. Figure 5.1 below shows the getUserDetail method in the user class,

```php
<?php
/**
*@author Benson Wachira
*@version 1.0
*/
include "../model/base.php";

class users extends base{

    /**
     * @method boolean getUserDetails() fetches all the details of a user in the database
        * @return bool
     */
    function getUserDetail($id)
    {
        $str_query = "SELECT * FROM _system_users where xx_user_id='$id' ";
        return $this->query($str_query);
    }
```

Figure 5.1: User class testing

I developed a test method as shown in figure 5.2 below.

```php
<?php
/**
*@author Benson Wachira
*@version 1.0.0
*Test class for users,post and alerts php classes
*/

include "../classes/user.php";

class testClass extends PHPUnit_Framework_TestCase
{

    /**
     * @method  testGetDetails() tests method for displaying all the users details given their id
     * @param no parameter
     * @return bool
     **/
    public function testGetDetails()
    {
        $us = new users();
        $id="1";

        $this->AssertTrue($us->getUserDetail($id));
    }
```

Figure 5.2: Test class implementation

I then ran the testClass class on the command prompt (*phpunit testClass.php*). Figure 5.3 below shows the results of the test.

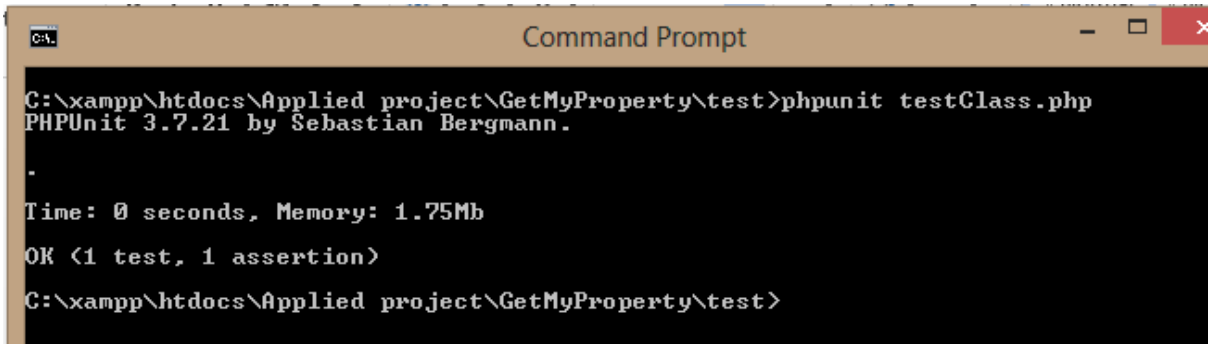

Figure 5.3: Unit testing result

I tested the major functions in the three classes to ensure that the implementation was accurate and the methods did what they were expected to do.

5.2.2. Component testing

A component is created by integrating multiple units. This form of testing focuses on testing the component interfaces (Sommerville, 2011). All the units in the three classes (user, post, alert) have been implemented independently. This means that the user class can be fully functional even if the post class has bugs. The three classes extend a base class which handles connection to the database, error handling and execution of queries. This component must be without any bug for the other classes to perform as expected.

5.2.3. System testing

This involves integrating the various components to create a system (Sommerville, 2011). The components that have been developed independently are combined to create the system. System testing is focused on testing the interactions between the various components that form the system (Sommerville, 2011).

For example, to add a property, the user has to register and use the credentials to login. I tested the various functionality of the system and ensured the components were working together as expected. I also tested the input fields by trying to add data with empty fields and providing wrong data types in the input fields. I did this to validate that the data provided by the users is clean and dependable. Figure 5.4 below shows the validation test (empty input field) when adding a property in the system.

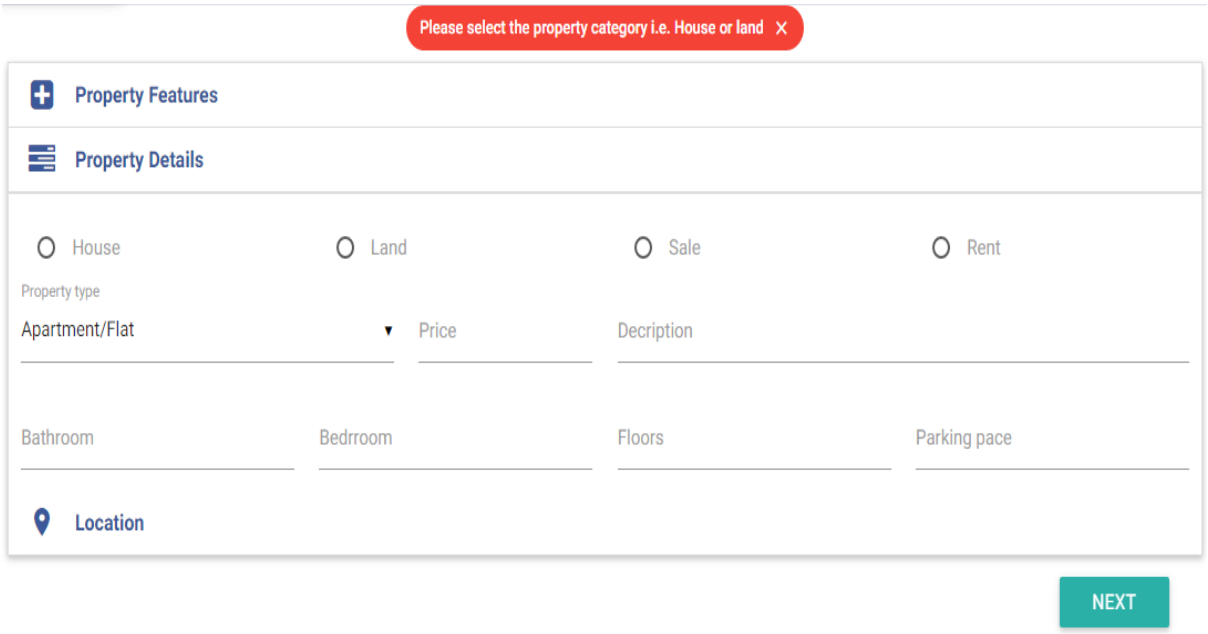

Figure 5.4: Validation (empty field)

Figure 5.5 below shows the validation test (invalid data type input) when adding a property in the system.

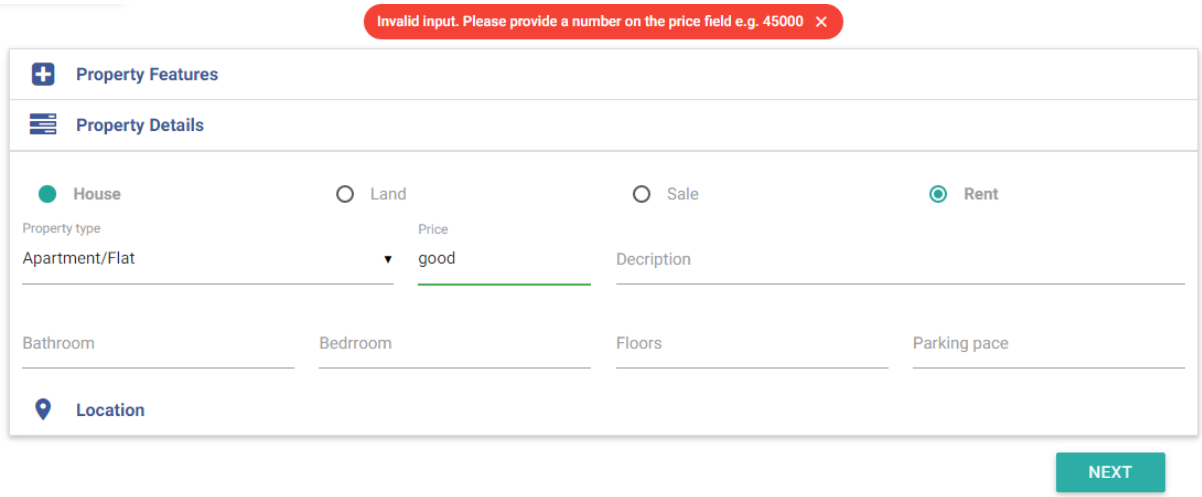

Figure 5.5: Validation (invalid data type)

5.3. User testing

User testing involves giving the system to the users for acceptance testing. The users use the system and give feedback if the system meets their expectations (Sommerville, 2011). In this testing, the users can identify flaws in the system that the development and testing team did not cater for during software development (Sommerville, 2011).

I conducted this testing by requesting five random Kenyan students at Ashesi University to test the software. My main focus was to evaluate the usability of the software while the users interacted with it. I briefed the users what the software is meant to do and gave them tasks to perform, such as create an account, subscribe for alerts, add a property, search property among others. I monitored the users during the exercise and noted the tasks that were easy and difficult to perform. I then took their feedback after the testing exercise.

5.3.1. User testing results

Below are the results of user testing.

i. The users had problem with selecting their date of birth from the date picker during registration. The date picker does not give a good range of years to select from.

ii. The phone number field lacks a hint of the format that the user is supposed to provide e.g. (233*********) and often the users entered the wrong format.

iii. When the users logged in with the new account details they did not know how to proceed since they were presented with a blank page. This is because they had not posted any property on the site.

iv. It was also difficult to locate the "add property" link on the navigation bar when I tasked the users to add a property.

v. The users did not know which fields to fill first on the first page of adding a property. This is because the fields are hidden in the collapsible. However, they easily navigated through the other pages with ease including uploading pictures.

vi. When the users made mistakes while posting an ad, they clicked on the edit icon to proceed to edit the details of the property.

vii. The users were also familiar with the subscribe alerts feature. They filled the required fields with ease.

viii. Users were also able to search for properties they need and easily adjust the search parameters to search for specific property.

ix. The users were also able to view details of the property by clicking on the heading of the property and also view seller information.

x. The similar properties below the details of the property they choose were also useful and caught their attention.

Chapter 6: Conclusion and Recommendation

6.1. Introduction

This chapter gives an overview of the milestones I was able to achieve while developing the real estate management application. It also highlights the recommendations, future work and development that can be implemented to make the application better and more efficient.

6.2. Summary

I choose this project because I am passionate about the solution it will provide. In the beginning, I did not consider how easy or complicated implementing some of the features was going to be, all I wanted was to work on a project that will help me (a property seeker) find a property more efficiently. At the end of this project, I do not regret the choice of topic since it has helped me learn how to develop an ecommerce website using different technologies. The project explored my strengths and also challenged me. I had to do research on how to implement some of the functionality in my application. The project was indeed a very good learning experience to expand my web development skills.

I learnt how to manage PHP sessions in AJAX to store the data provided by the users while adding a property. This feature made it possible for the users to go back to the previous pages and edit the details provided earlier without refilling all the fields.
I made use of SMS gateway to send text messages to the users during password reset. I also learnt how to design normalized database tables and write optimized queries. The interaction with my supervisor can be compared to interaction with a client. This made me put into practice the process of software development and accept feedback and constructive criticism.

6.3. Conclusion

I am pleased I was able to attain the objectives set in chapter one by developing a real estate management application. When this project is implemented, it will help the property owners and managers advertise their property and provide an efficient means of searching for property in Kenya. However, a business plan must be written before implementation, detailing means of generating income and the competitive advantage of the system over the existing applications.

6.4.Recommendations and Future Work

The real estate management application can be improved by including the following features.

i. Uploading and deleting multiple images at once. Currently, a user can only upload and delete one image at a time. The user should be able to select all the images he/she wants and upload them all with a single click. The user should also be able to delete multiple images both in the session variables and on the upload folder. The images should also be deleted when a property is deleted. This will conserve space by deleting the images that are not in use.

ii. Implementing the administrator dashboard. A dashboard where the administrator can monitor the users and the ads. The administrator can manage the application more efficiently and ensure that all the posts are up to date though this dashboard. It can also include features such as generating reports.

iii. Checking the number of attempts for invalid username and password. To improve the software security, the application should keep count of the number of times a user has tried to log in with invalid username or password. If the count exceeds the limit, the user is blocked and the administrator is notified. The administrator can activate the user upon complete authentication of the user.

iv. Reactivate posts that have been deactivated. Both free and featured ads are deactivated after a period of time and cannot be viewed on the site again. The user should be able to activate their ads by making a request to the administrator for activation.

v. Payment module- During implementation I included a featured plan when uploading a property where the user needs to pay a fee of 500 KES (5 USD) before proceeding to post the ad. However, this feature was not integrated to payment and billing systems. In order to make the application sustainable, the application should be integrated with payment options such as credit and debit cards, mobile money (e.g. M pesa and Airtel money). The property owners and managers should be able to make instant payments while posting featured ads.

vi. Summary of all validation. The application verifies one input field at a time when a user is filling a form. The validation should be extended to provide a summary of what the user is supposed to provide and the correct format. This will improve the usability of the application. In addition to doing client side validation, there should be server side validation to prevent cross site scripting in case the user disables JavaScript.

vii. Adding featured ads to the homepage slides. The administrator should have a dashboard for uploading images and details of featured properties for display on the homepage slides. The property owners will pay more to advertise their property on the homepage slides.

References

Martin. (2015, 21 06). *Top Programming Languages used in Web Development*. Retrieved 03 27, 2016, from cleverism.com: http://www.cleverism.com/programming-languages-web-development/

Mundi, I. (2015, 07 30). *Kenya Demographics Profile 2014*. Retrieved 09 21, 2015, from Indexmundi.com: http://www.indexmundi.com/kenya/demographics_profile.html

propertymatrix. (2003, 00 00). *Real Estate Property Management Software*. Retrieved 03 27, 2016, from propertymatrix.com: https://www.propertymatrix.com/articles/real-estate/real-estate-software/real-estate-property-management-software

reviews.com. (2015, 11 10). *Property Management Software Reviews*. Retrieved 03 27, 2016, from reviews.com: http://www.reviews.com/property-management-software/

Shiotsu, Y. (2014, 03 19). *Web Development 101: Top Web Development Languages in 2014*. Retrieved 03 27, 2015, from upwork.com: https://www.upwork.com/blog/2014/03/web-development-101-top-web-development-languages-2014/

Sommerville, I. (2011). *Software Engineering.* United States of America: Pearson.

Webfinanceinc. (2006, 00 00). *real estate*. Retrieved 03 27, 2016, from businessdictionary.com: http://www.businessdictionary.com/definition/real-estate.html