

FINDMYNEST REAL ESTATE MANAGEMENT SYSTEM

Project Report Submitted By

NIKKY GEORGE PHILIP

Reg. No.: AJC17MCA-I044

In Partial fulfillment for the Award of the Degree Of

**INTEGRATED MASTER OF COMPUTER APPLICATIONS
(INMCA)**
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



**AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2017-2022

DEPARTMENT OF COMPUTER APPLICATIONS
AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY



CERTIFICATE

This is to certify that the Project report, "**FINDMYNEST REAL ESTATE MANAGEMENT SYSTEM**" is the bonafide work of **NIKKY GEORGE PHILIP (Reg.No:AJC17MCA-I044)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2017-22.

Mr. Jinson Devis
Internal Guide

Ms. Meera Rose Mathew
Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose
Head of the Department

External Examiner

DECLARATION

I hereby declare that the project report "**FINDMYNEST REAL ESTATE MANAGEMENT SYSTEM**" is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2017-2022.

Date: **NIKKY GEORGE PHILIP**
KANJIRAPPALLY **Reg. No: AJC17MCA-I044**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide, **Mr. Jinson Devis** for his inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

NIKKY GEORGE PHILIP

ABSTRACT

Findmynest Real Estate Management System is a web application which is meant to help the customers to buy and sell their properties easily. The customer can also reduce the time and effort for searching buyers by using this system. Customers can post ads of their properties i,e Houses ,Villas , Apartments and make use of the web application to stay connected with Agents also. The proposed system includes three users they are administrator, user and agent. Registered customers can login to the site and can either post their ads for free or they can approach an agent for selling their properties faster. The customer can also see the status of the ads posted. Customers can also chat with the agents if they are looking to buy a property. Agents can view the requests that made buy the customers. For posting their ads and they can review the details and post ad by making a payment. Agents can also see the enquiries from the customers about the ads posted by them. The administrator has the central control over the whole system. Whenever a user posts their ad it will be under review and will be activated only when the administrator approves the ad. The administrator can manage the ads , users etc. Customers can apply for home loans through our site. Customers can also check the status of the loan application. The customer can see the status of the ads posted. The website also provide a feature to instantly chat with the agents if they are looking to buy a property. Agents can view the Ad posting requests that made buy the customers. A rental system shall be added to the project so that a client can easily rent a house, Villa or Flat etc. User can book an appointment with the agent to discuss about the property.

CONTENT

Sl. No	Topic	Page No
1	INTRODUCTION	1
1.1	PROJECT OVERVIEW	2
1.2	PROJECT SPECIFICATION	2
2	SYSTEM STUDY	4
2.1	INTRODUCTION	5
2.2	EXISTING SYSTEM	6
2.3	DRAWBACKS OF EXISTING SYSTEM	6
2.4	PROPOSED SYSTEM	6
2.5	ADVANTAGES OF PROPOSED SYSTEM	7
3	REQUIREMENT ANALYSIS	8
3.1	FEASIBILITY STUDY	9
3.1.1	ECONOMICAL FEASIBILITY	9
3.1.2	TECHNICAL FEASIBILITY	10
3.1.3	BEHAVIORAL FEASIBILITY	10
3.2	SYSTEM SPECIFICATION	11
3.2.1	HARDWARE SPECIFICATION	11
3.2.2	SOFTWARE SPECIFICATION	11
3.3	SOFTWARE DESCRIPTION	11
3.3.1	PHP	11
3.3.2	MYSQL	12
4	SYSTEM DESIGN	14
4.1	INTRODUCTION	15
4.2	UML DIAGRAM	15
4.2.1	USE CASE DIAGRAM	16
4.2.2	SEQUENCE DIAGRAM	19
4.2.3	STATE DIAGRAM	21
4.2.4	COMPONENT DIAGRAM	22
4.2.5	DEPLOYMENT DIAGRAM	23
4.2.6	COLLABORATION DIAGRAM	24
4.2.7	ACTIVITY DIAGRAM	25
4.2.8	OBJECT DIAGRAM	26

4.2.9	CLASS DIAGRAM	27
4.3	USER INTERFACE DESIGN USING FIGMA	28
4.4	DATA BASE DESIGN	35
5	SYSTEM TESTING	47
5.1	INTRODUCTION	48
5.2	TEST PLAN	49
5.2.1	UNIT TESTING	49
5.2.2	INTEGRATION TESTING	55
5.2.3	VALIDATION TESTING	55
5.2.4	USER ACCEPTANCE TESTING	57
6	IMPLEMENTATION	58
6.1	INTRODUCTION	59
6.2	IMPLEMENTATION PROCEDURE	59
6.2.1	USER TRAINING	60
6.2.2	TRAINING ON APPLICATION SOFTWARE	60
6.2.3	SYSTEM MAINTENANCE	60
7	CONCLUSION & FUTURE SCOPE	61
7.1	CONCLUSION	62
7.2	FUTURE SCOPE	62
8	BIBLIOGRAPHY	63
9	APPENDIX	65
9.1	SAMPLE CODE	66
9.2	SCREEN SHOTS	82

List of Abbreviation

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

“FINDMYNEST REAL ESTATE MANAGEMENT SYSTEM” is a web application which is meant to help the customers to buy and sell their properties easily. The customer can also reduce the time and effort for searching buyers by using this system. Customers can post ads of their properties i,e Houses ,Villas , Apartments and make use of the web application to stay connected with Agents also. The proposed system includes three users they are administrator, user and agent. Registered customers can login to the site and can either post their ads for free or they can approach an agent for selling their properties faster. The customer can also see the status of the ads posted. Customers can also chat with the agents if they are looking to buy a property. Agents can view the requests that made buy the customers. For posting their ads and they can review the details and post ad by making a payment. Agents can also see the enquiries from the customers about the ads posted by them. The administrator has the central control over the whole system. Whenever a user posts their ad it will be under review and will be activated only when the administrator approves the ad. The administrator can manage the ads , users etc. Customers can apply for home loans through our site. Customers can also check the status of the loan application. The customer can see the status of the ads posted. The website also provide a feature to instantly chat with the agents if they are looking to buy a property. Agents can view the Ad posting requests that made buy the customers. A rental system shall be added to the project so that a client can easily rent a house, Villa or Flat etc. User can book an appointment with the agent to discuss about the property.

1.2 PROJECT SPECIFICATION

The proposed system is made to help the customers to buy and sell their properties easily and faster. This system will also act as a bridge between the customer and agent. We will also provide facility for users to post their review about the agent. They can also view details of their ads and the ads they have added to their favorite list.

The system includes 3 modules. They are:

1. Admin Module

Admin must have a login into this system. He has the overall control of the system. Admin

can view and update the details of the ads posted on the website. Admin can also view and manage the user details, favorite ads, locations.

2. Customer Module

Customer can register and either post their ads for free or they can approach an agent for selling their properties faster. The customer can also see the status of the ads posted. Customers can also chat with the agents if they are looking to buy a property. They can also search for a particular agent nearest to their location.

3. Agent Module

Agents can view the requests that made by the customers. For posting their ads and they can review the details and post ad by making a payment. Agents can also see the enquiries from the customers about the ads posted by them.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 EXISTING SYSTEM

Existing system is not a fully automated system. Customer can register and they can book for service. Each customer can create their own profile .The proposed system rectify the drawbacks of the present system.

It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure. Using the new system customers can post their ads for free or they can approach an agent for selling their properties faster. The customer can also see the status of the ads posted. They can also search for a particular agent nearest to their location. Customers can also chat with the agents if they are looking to buy a property.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Less convenient in posting the ad details including finding an agent nearest to their location.
- Human effort is needed.
- It is difficult to maintain important information in books.
- More manual hours need to generate required reports.

2.4 PROPOSED SYSTEM

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive for growth of service center; on such consideration the system is proposed. In our proposed system there is admin who can view all the customers. It allows customers to post their ads or approach an agent to get their ad posted and featured which make it easy to be sold faster .Users of this proposed system are admin , customer and agent. The software application which avoids more manual hours that need to spend in record keeping and generating reports. This application keeps the data in a centralized way which is available to all the users simultaneously. It is very easy to manage historical data in database.

2.5 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources, and the system will work in almost all configurations. It has got following features:

➤ **Better security:** -

For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining the documents.

➤ **Ensure data accuracy:** -

The proposed system eliminates the manual errors while entering the details of the users during the registration.

➤ **Better service:** -

The product will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

The cost of project, Findmynest Real Estate Management System was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.

3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 4GB and, Hard disk 1TB

3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - Intel core i3

RAM - 4 GB

Hard disk - 1 TB

3.2.2 Software Specification

Front End - HTML, CSS

Backend - MYSQL

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page ,it now stands for PHP:HypertextPreprocessor, a recursive acronym.PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page.PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You setup rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL92” refers to the standard released in 1992,

“SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

- **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After some standardization, UML has

become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- Statechart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles are played by the actors within and around

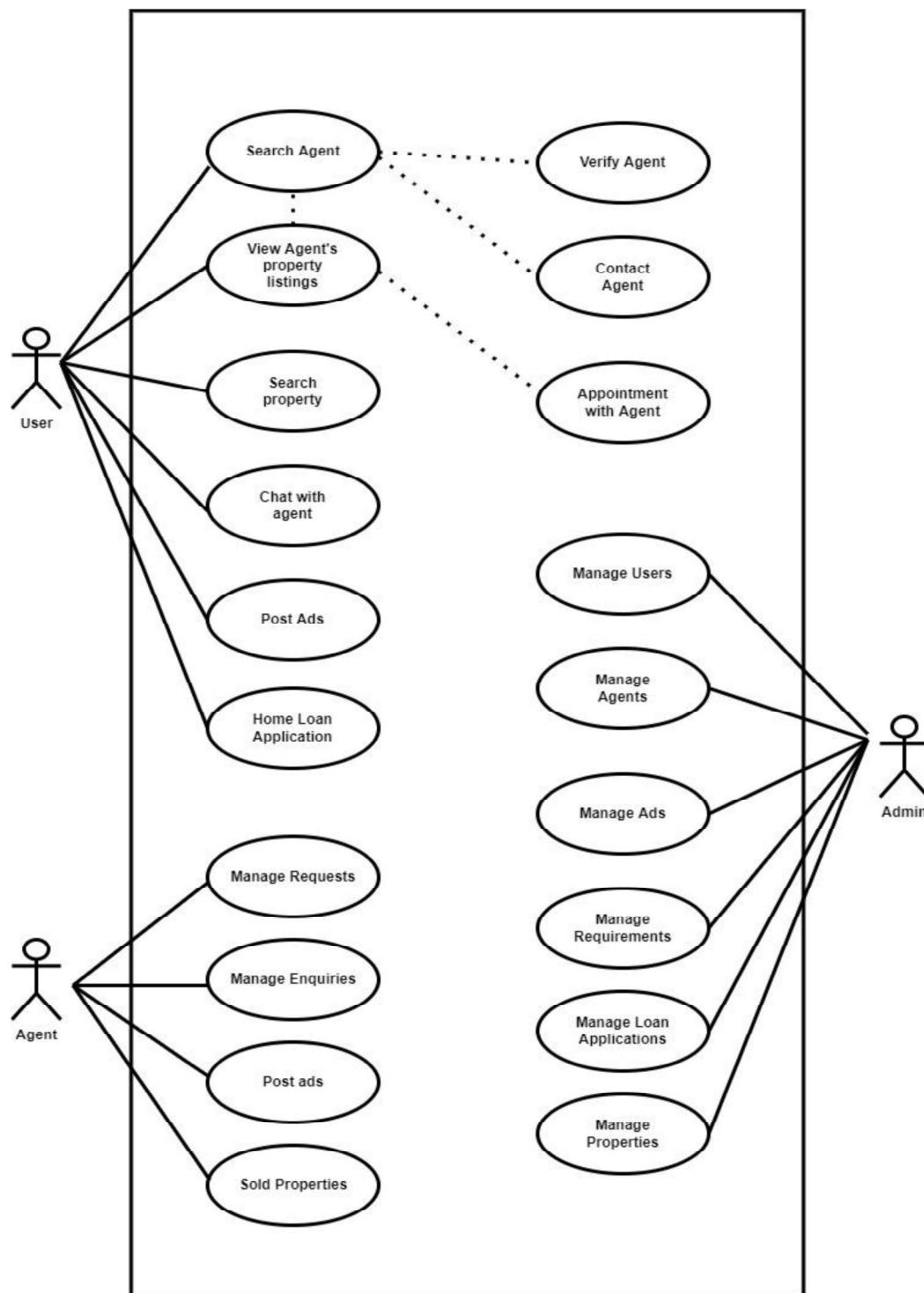
the system.

- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Usecase Diagram



4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

- i. **Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.
- ii. **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram
- iii. **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.
Messages can be broadly classified into the following categories:
 - Synchronous messages
 - Asynchronous Messages
 - Create message
 - Delete Message
 - Self-Message

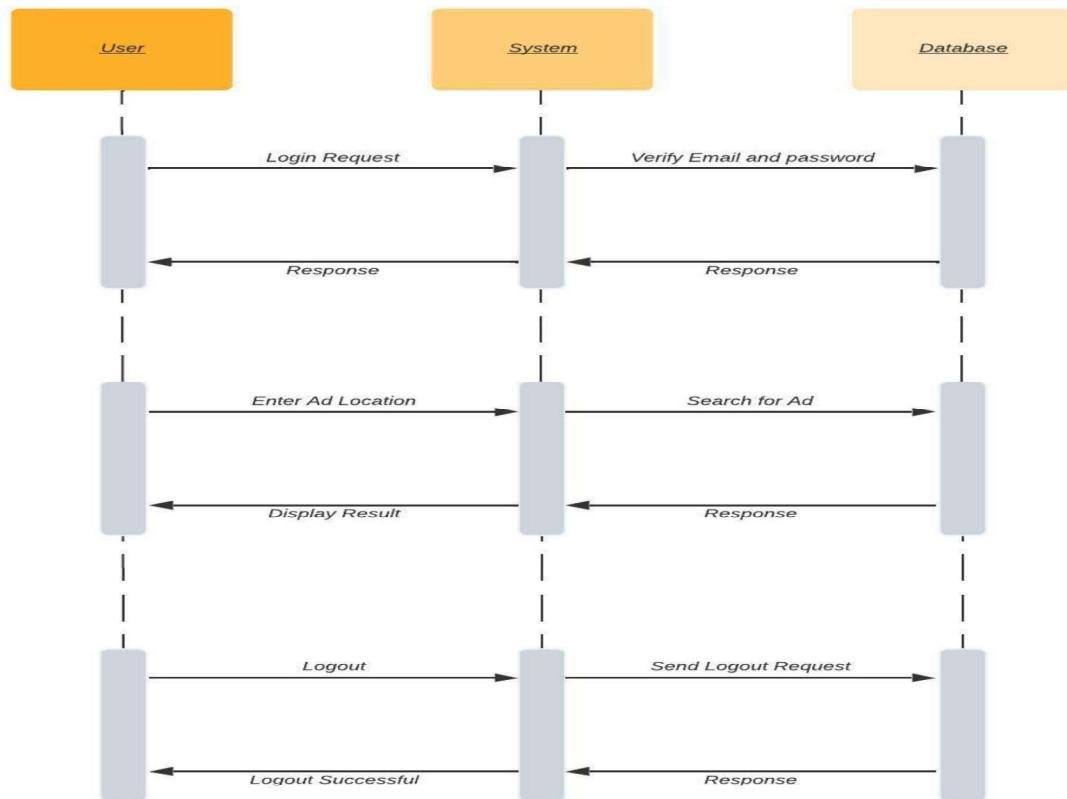
- Reply Message
- Found Message
- Lost Message

iv. Guards – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.

Sequence Diagram



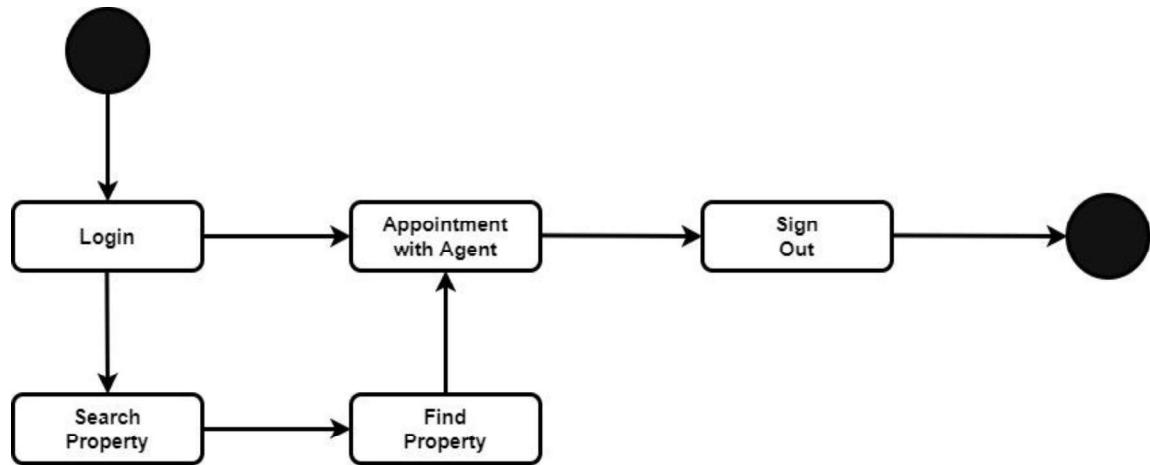
4.2.3 STATE DIAGRAM

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioral diagram and it represents the behavior using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams. These terms are often used interchangeably. So simply, a state diagram is used to model the dynamic behavior of a class in response to time and changing external stimuli. We can say that each and every class has a state but we don't model every class using State diagrams. We prefer to model the states with three or more states.

Uses of statechart diagram –

- We use it to state the events responsible for change in state (we do not show what processes cause those events).
- We use it to model the dynamic behavior of the system .
- To understand the reaction of objects/classes to internal or external stimuli.

State Diagram

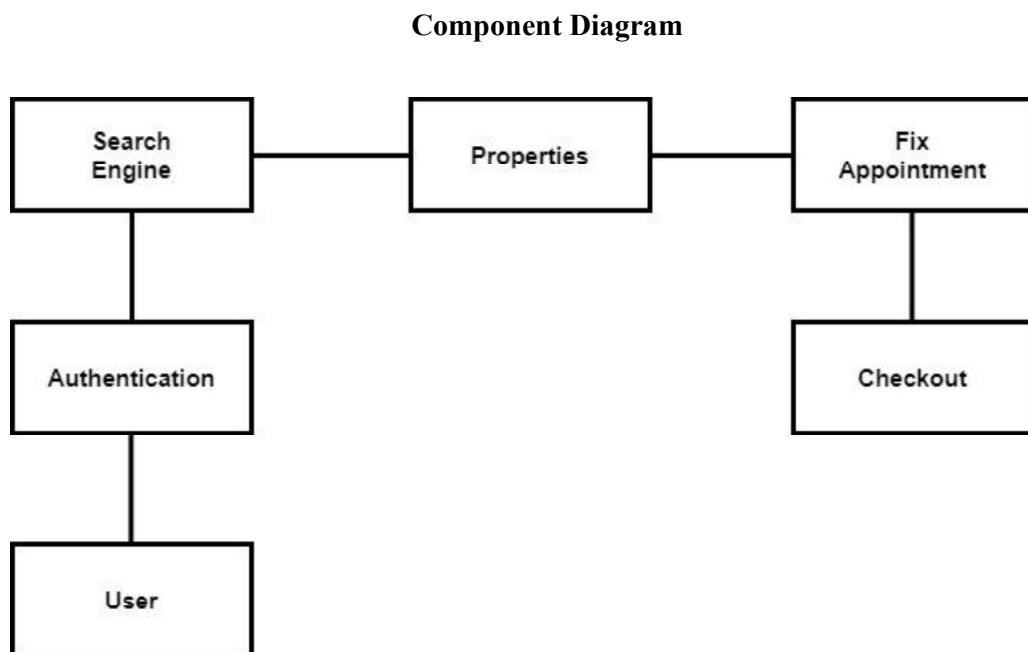


4.2.4 COMPONENT DIAGRAM

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities. Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc. Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

The purpose of the component diagram can be summarized as –

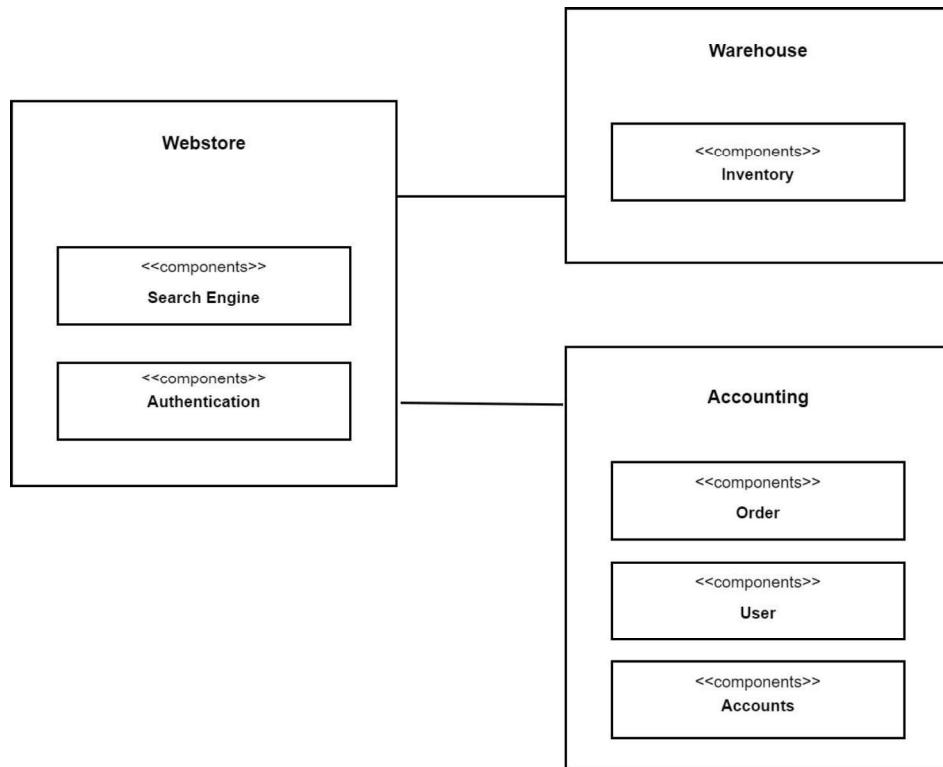
- Visualize the components of a system.
- Construct executables by using forward and reverse engineering.
- Describe the organization and relationships of the components.



4.2.5 DEPLOYMENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. UML is mainly designed to focus on the software artifacts of a system. However, these two diagrams are special diagrams used to focus on software and hardware components. Most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on the hardware topology of a system. Deployment diagrams are used by the system engineers.

Deployment Diagram



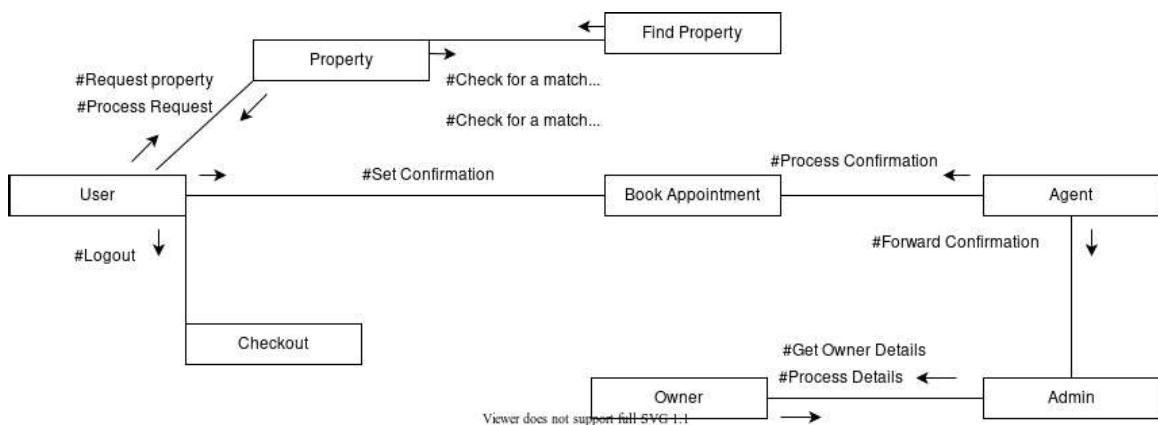
4.2.6 COLLABORATION DIAGRAM

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

Following are the components of a component diagram that are enlisted below:

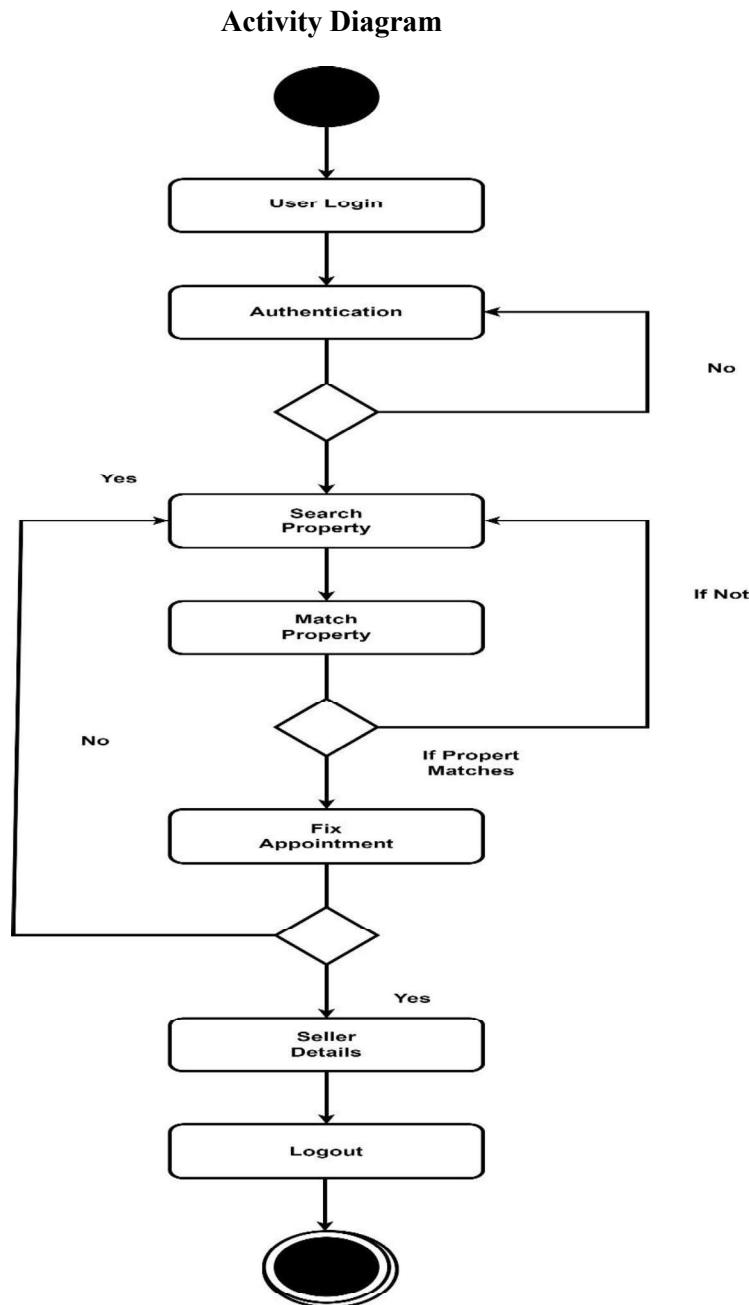
1. **Objects:** The representation of an object is done by an object symbol with its name and class underlined, separated by a colon.
2. **Actors:** In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective role and name. In this, one actor initiates the use case.
3. **Links:** The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line.
4. **Messages:** It is a communication between objects which carries information and includes a sequence number, so that the activity may take place.

Collaboration Diagram



4.2.7 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

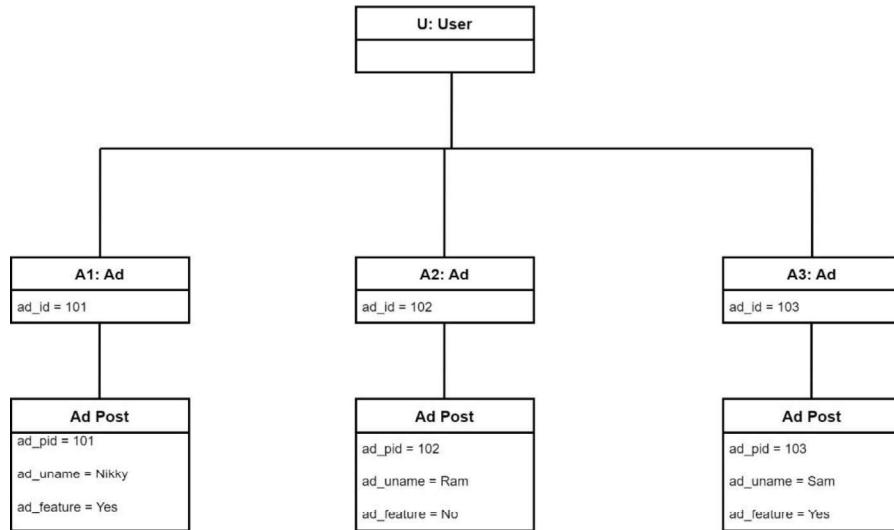


4.2.8 OBJECT DIAGRAM

Object Diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment. Object diagrams are used to render a set of objects and their relationships as an instance.

The purpose of a diagram should be understood clearly to implement it practically. The purposes of object diagrams are similar to class diagrams. The difference is that a class diagram represents an abstract model consisting of classes and their relationships. However, an object diagram represents an instance at a particular moment, which is concrete in nature. It means the object diagram is closer to the actual system behavior. The purpose is to capture the static view of a system at a particular moment. The purpose of the object diagram can be summarized as –

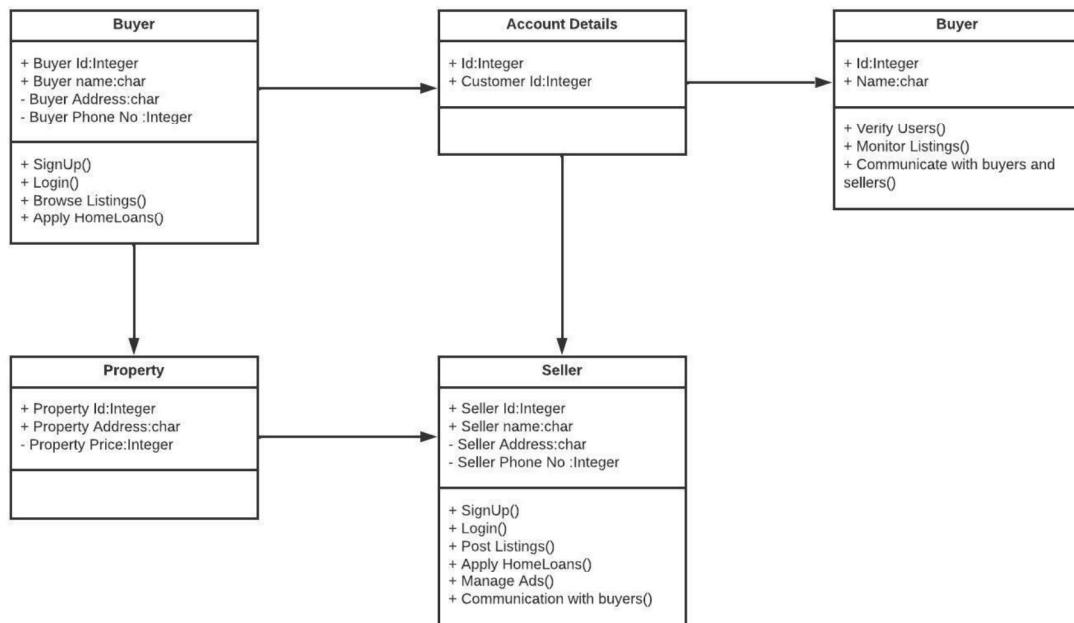
- Forward and reverse engineering.
- Object relationships of a system Static view of an interaction.
- Understand object behavior and their relationship from practical perspective



4.2.9 CLASS DIAGRAM

Class Diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

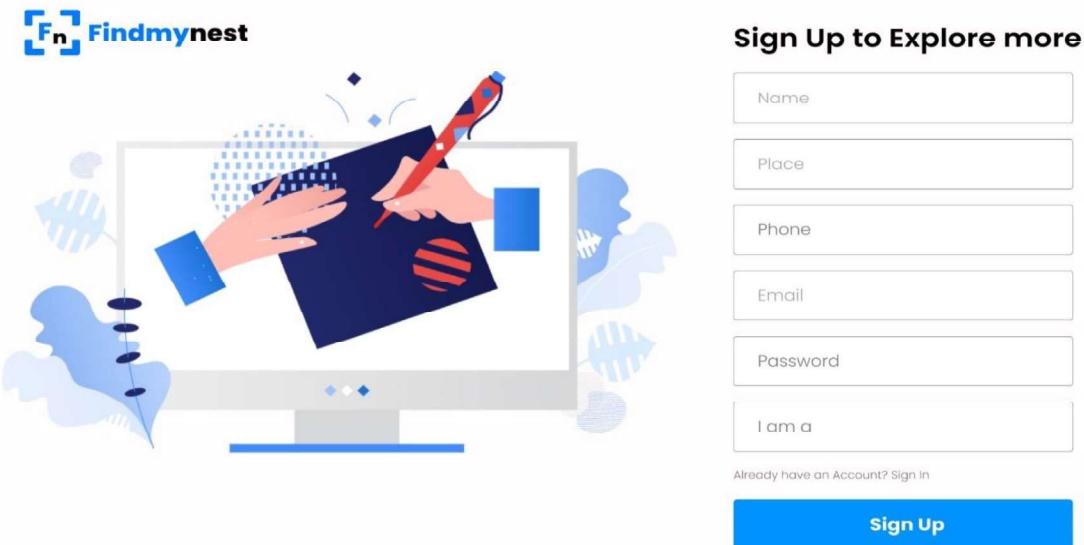
The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction. UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application; however class diagram is a bit different. It is the most popular UML diagram in the coder community. The purpose of the class diagram can be summarized as –



4.3 USER INTERFACE DESIGN USING FIGMA

4.3.1-INPUT DESIGN

Form Name : User Registration



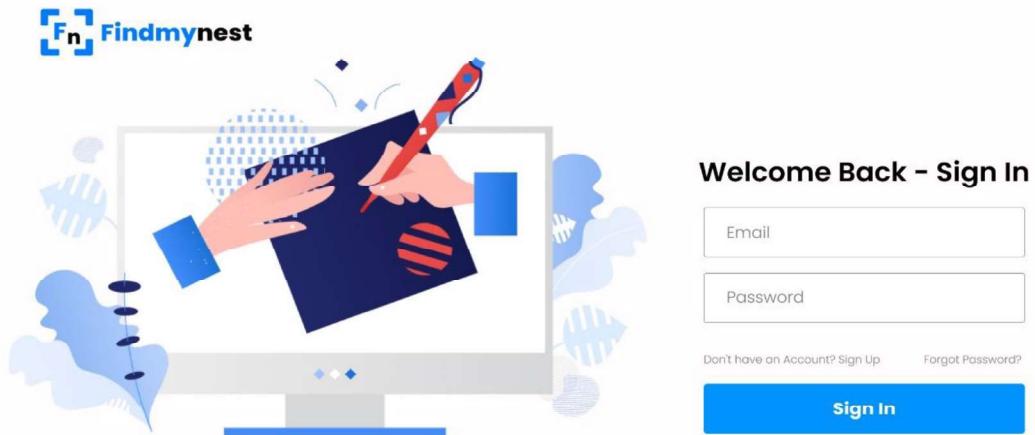
The form features a header with the Findmynest logo and a stylized illustration of hands interacting with a computer screen displaying a globe. The main section is titled "Sign Up to Explore more" and contains six input fields: Name, Place, Phone, Email, Password, and a dropdown for "I am a". A "Sign Up" button is at the bottom.

Name
Place
Phone
Email
Password
I am a

Already have an Account? Sign In

Sign Up

Form Name : User Login



The form features a header with the Findmynest logo and a stylized illustration of hands interacting with a computer screen displaying a globe. The main section is titled "Welcome Back - Sign In" and contains two input fields: Email and Password. Below the fields are links for "Don't have an Account? Sign Up" and "Forgot Password?". A "Sign In" button is at the bottom.

Email
Password

Don't have an Account? Sign Up Forgot Password?

Sign In

Form Name : Post Ad

Ad Details

Title

Description

Describe about your Ad

Category

Property type

Price

Location

Area (Sq.ft)

Rooms

Bathrooms

Bed Rooms

Image 1

Choose file No file chosen

Image 2

Choose file No file chosen

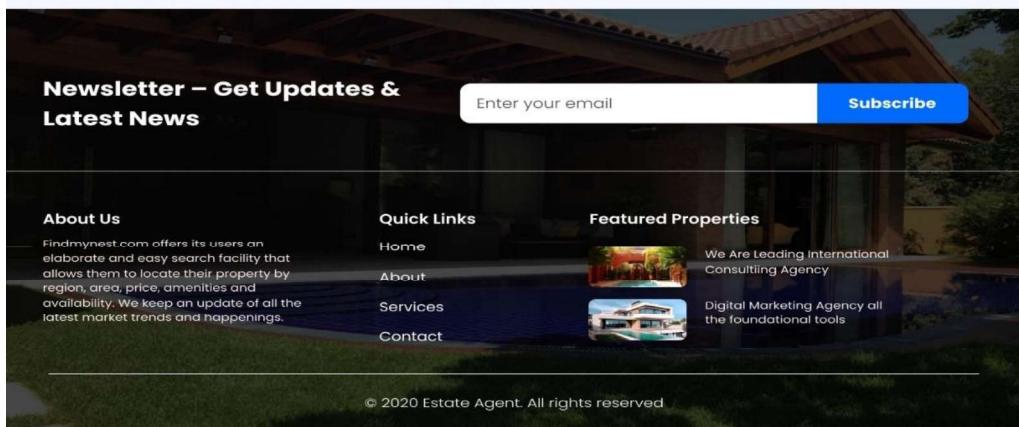
Image 3

Choose file No file chosen

Do you want to Feature your Ad ?(Rates apply)

Yes No

Post Your Ad



Form Name : Post Requirements

Findmynest
[Home](#)
[Buy](#)
[Sell](#)
[Rent](#)
[Contact](#)
[About Us](#)

Tell us about your Requirements

Property Type

Select One

Car Parking Spaces

Select One

Furnished Status

Select One

Budget Amount

Select One

Prefered Location

Select One

Area (Sq.ft)

Select One

Rooms

Select One

Bathrooms

Select One

Bedrooms

Select One

Post Requirements

Newsletter – Get Updates & Latest News

Subscribe

About Us

Findmynest.com offers its users an elaborate and easy search facility that allows them to locate their property by region, area, price, amenities and availability. We keep an update of all the latest market trends and happenings.

Quick Links

- [Home](#)
- [About](#)
- [Services](#)
- [Contact](#)

Featured Properties



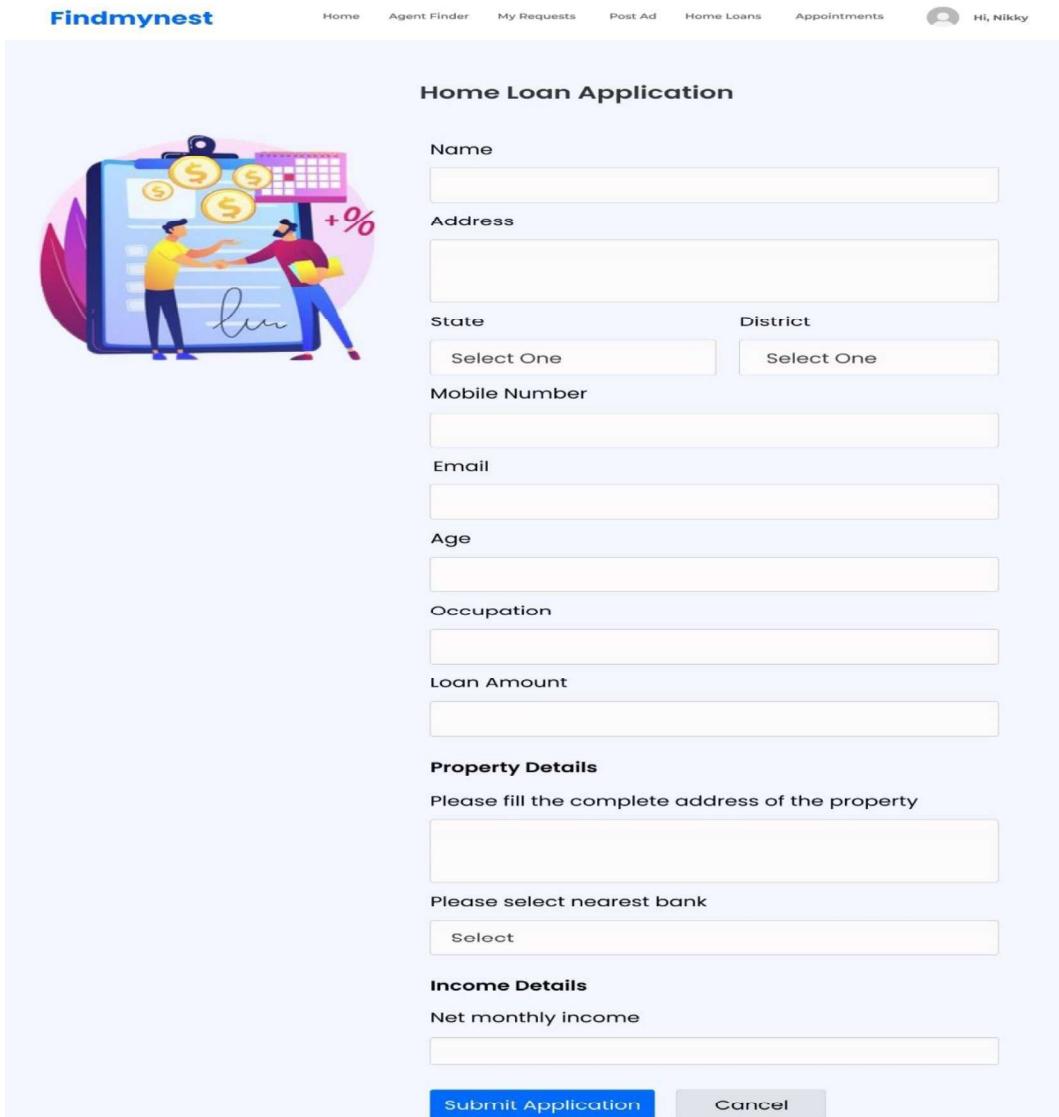
We Are Leading International Consulting Agency



Digital Marketing Agency all the foundational tools

© 2020 Estate Agent. All rights reserved.

Form Name : Home Loan



The screenshot shows the 'Home Loan Application' form on the Findmynest website. At the top, there's a navigation bar with links for Home, Agent Finder, My Requests, Post Ad, Home Loans, Appointments, and a user profile for 'Hi, Nikky'. The main form area has a title 'Home Loan Application' and features a background illustration of two people shaking hands over a clipboard with coins and a percentage sign. The form fields include:

- Name: Input field
- Address: Input field
- State: Select dropdown with 'Select One' option
- District: Select dropdown with 'Select One' option
- Mobile Number: Input field
- Email: Input field
- Age: Input field
- Occupation: Input field
- Loan Amount: Input field

Property Details

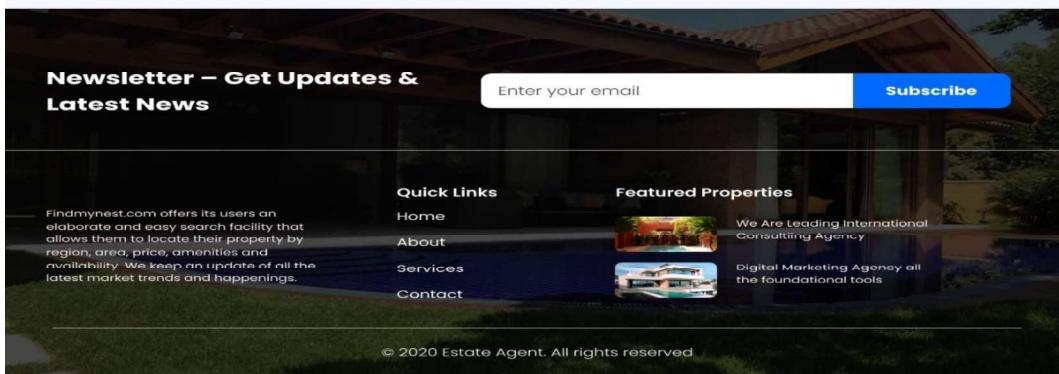
Please fill the complete address of the property: Input field

Please select nearest bank: Select dropdown with 'Select' option

Income Details

Net monthly income: Input field

At the bottom are two buttons: 'Submit Application' (blue) and 'Cancel'.



Form Name : Ad view

Findmynest
Home Agent Finder My Requests Post Ad Home Loans Appointments
Hi, Nikky



House For Sale
Rs. 9500000

For Sale **House**

Description
Good Furnished House For Sale

Add to Favorites **Enquire now**

Book an Appointment

Seller Details



Nikky
 Email ID: nikky@gmail.com
 Location: Kottayam

Ad Details



Ad ID: 62
 Location: Kottayam
 Posted On: 15/12/2021

Newsletter – Get Updates & Latest News

Subscribe

About Us

Findmynest.com offers its users an elaborate and easy search facility that allows them to locate their property by region, area, price, amenities and availability. We keep an update of all the latest market trends and happenings.

Quick Links

- [Home](#)
- [About](#)
- [Services](#)
- [Contact](#)

Featured Properties



We Are Leading International Consulting Agency



Digital Marketing Agency all the foundational tools

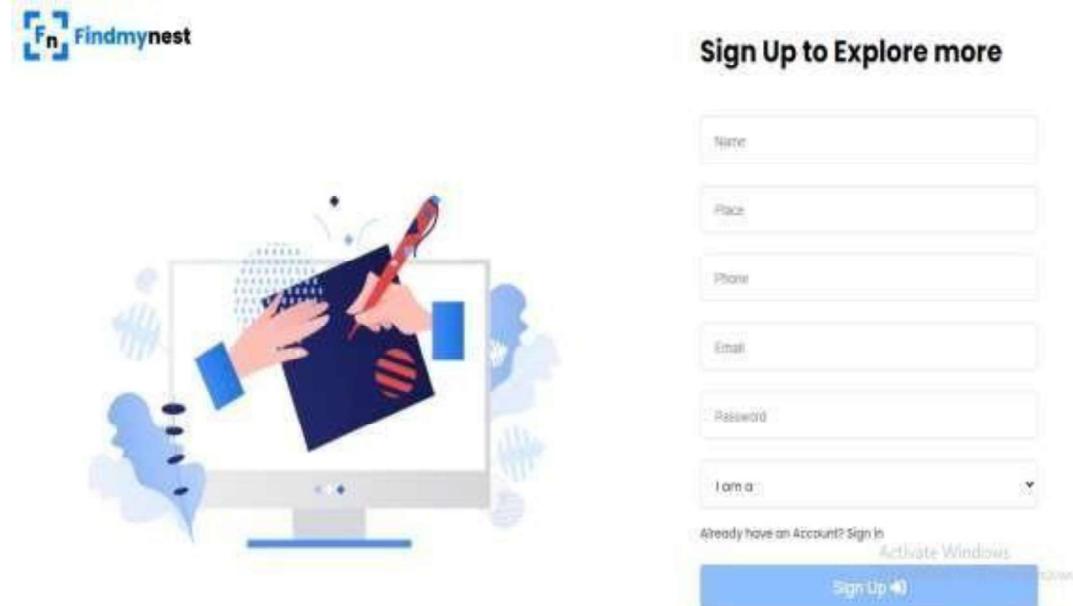
© 2020 Estate Agent. All rights reserved

Amal Jyothi College of Engineering, Kanjirappally

Department of Computer Applications

4.3.2 OUTPUT DESIGN

User Registration



The registration form features a header with the Findmynest logo and a call-to-action button 'Sign Up to Explore more'. Below the header is a central illustration of a hand writing on a notepad. To the right of the illustration are input fields for Name, Place, Phone, Email, and Password. A dropdown menu for 'I am a:' is also present. At the bottom, there are links for 'Already have an Account? Sign In' and 'Activate Windows', followed by a large blue 'Sign Up' button.

Name

Place

Phone

Email

Password

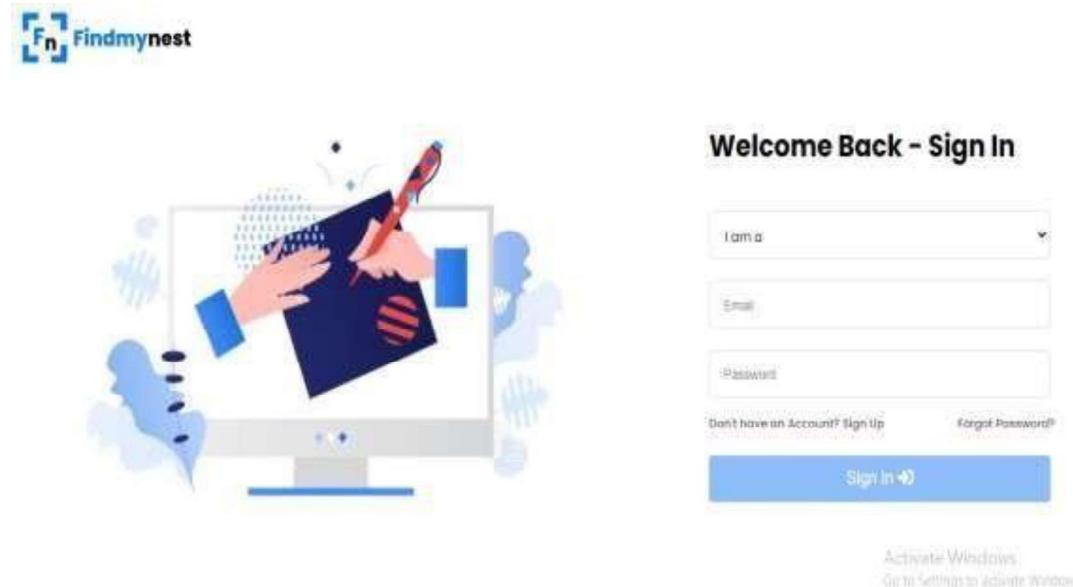
I am a:

Already have an Account? [Sign In](#)

[Activate Windows](#)

[Sign Up](#)

User Login



The login form has a header with the Findmynest logo and a 'Welcome Back - Sign In' message. It includes input fields for 'I am a', 'Email', and 'Password'. Below these fields are links for 'Don't have an Account? [Sign Up](#)' and 'Forgot Password?'. A large blue 'Sign In' button is at the bottom. At the very bottom, there are links for 'Activate Windows' and 'Go to Settings to activate Windows'.

Welcome Back - Sign In

I am a

Email

Password

Don't have an Account? [Sign Up](#)

[Forgot Password?](#)

[Sign In](#)

Activate Windows

Go to Settings to activate Windows

Post Ad

Tell us about your Ad



Free ADS:-

- No extra charges.
- 3 months validity.
- 24x7 Support.
- Free Maintenance.

Featured ADS:-

- Cost Applicable.
- Lifetime validity.
- 24x7 Support.
- Free Maintenance.

Title		
Description	Describe about your ad	
Category	Property Type	
Select Category	Select Type	
Price	Location	Area Sq.ft
	Select One	
Rooms	Bathrooms	Bed Rooms
Select One	Select One	Select One
Image 1	Image 2	Image 3
<input type="button" value="Choose file"/> No file chosen	<input type="button" value="Choose file"/> No file chosen	<input type="button" value="Choose file"/> No file chosen
		
<input type="button" value="Post Your Ad"/> <input type="button" value="Cancel"/>		

Active
Go to Se

Post Requirements

Tell us about your Requirements



Description

Describe about your ad

Property Type	Budget Amount	
Select Type		
Location	Area Sq.ft	
Rooms	Bathrooms	Bed Rooms
Select One	Select One	Select One
<input type="button" value="Post Your Requirements"/>		

Free ADS:-

- No extra charges.
- 3 months validity.
- 24x7 Support.
- Free Maintenance.

Featured ADS:-

- Cost Applicable.
- Lifetime validity.
- 24x7 Support.
- Free Maintenance.

Active
Go to Se

4.4 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

4.4.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

4.4.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be done by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminates repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

TABLE DESIGN**Table No. 01****Table Name : tbl_usercredentials****Primary Key : user_id****Foreign Key :****Table Description : To store Login information**

Field Name	Data Type	Size	Description
User_id	Int	20	Primary key of login table
Email	Varchar	100	Unique username to login
pass	Varchar	100	Password to login
createdDate	Varchar	50	Account creation date
role	Varchar	20	Role
Status	Int	10	Status of the user

Table No. 02**Table Name : tbl_userdetails****Primary Key : user_id****Foreign Key :****Table Description : To store Users details**

Field Name	Data Type	Size	Description
User_id	Int	50	Primary key of user details table
fname	Varchar	50	User's Name
address	Varchar	200	Address of user
phone	Varchar	10	Phone number of User
place	Varchar	100	Place of the user
createdDate	Varchar	200	Account creation date
Status	Int	5	Status of the User

Table No. 03**Table Name : tbl_agents****Primary Key : agent_id****Foreign Key :****Table Description : To store Agent details**

Field Name	Data Type	Size	Description
agent_id	Int	10	Primary key of agent table
name	Varchar	30	Agent's Name
email	Varchar	40	Email of agent
password	Varchar	20	Password to login
address	Varchar	100	Address of Agent
phone	Varchar	12	Phone number of agent
place	Varchar	30	Place of the agent
createdDate	Varchar	30	Account creation date
commission	Varchar	10	Commission of agent
role	Varchar	20	To show the role
avg_rating	varchar	10	Average rating of agent
status	Int	10	Status of the Agent

Table No. 04**Table Name : tbl_ads****Primary Key : ad_id****Foreign Key :****Table Description : To store Ad details**

Field Name	Data Type	Size	Description
ad_id	Int	10	Primary key of Ads table
Ad_title	Varchar	30	Title of the ad
Ad_descriptio n	Varchar	1500	Description about ad

Ad_type	Varchar	20	Type of ad
Ad_loc	Varchar	20	Location of ad
Ad_price	Varchar	15	Price of ad
Ad_cat	Varchar	15	Category of ad
Lat	Varchar	30	Latitude of location
Lng	Varchar	30	Longitude of location
createdDate	Varchar	30	Ad posted date
room	Int	10	Total number of rooms
Bathroom	Int	10	Total bathrooms
Bedroom	Int	10	Total bedrooms
Ad_feature	Varchar	15	Featured or not
Area	Varchar	10	Total area of property
Image1	Varchar	100	Image 1 of property
Image2	Varchar	100	Image 2 of property
Image3	Varchar	100	Image 3 of property
Feature1	Varchar	30	Feature 1 of property
Feature2	Varchar	30	Feature 2 of property
Feature3	Varchar	30	Feature 3 of property
Feature4	Varchar	30	Feature 4 of property
Feature5	Varchar	30	Feature 5 of property
Feature6	Varchar	30	Feature 6 of property
Feature7	Varchar	30	Feature 7 of property
Username	Varchar	100	Username of the ad owner
Seller_name	Varchar	30	Name of the seller who posted ad
Seller_email	Varchar	40	Email of the seller who posted ad
status	Int	10	Status of the ad

Table No. 05

Table Name : tbl_requirements

Primary Key : req_id

Foreign Key :

Table Description : To store Requirement details

Field Name	Data Type	Size	Description
req_id	Int	20	Primary key of Requirements table
req_descriptio n	Varchar	100	Description about requirement
req_price	Varchar	30	Budget
req_type	Varchar	30	Type of property
Req_loc	Varchar	50	Location of property
Furnish_status	Varchar	30	Furnishing Status
Parkspaces	Integer	20	Parking Spaces
createdDate	Varchar	100	Requirements posted date
room	Int	10	Total no.of rooms
Bathroom	Int	10	Total no.of bathrooms
Bedroom	Int	10	Total no.of bedrooms
area	Int	20	Total area
User_mail	Varchar	50	User mail id
status	Int	11	Status

Table No. 06

Table Name : tbl_enquiry

Primary Key : en_id

Foreign Key :

Table Description : To store Enquiry details

Field Name	Data Type	Size	Description
En_id	Int	10	Primary key of Enquiry table
Ad_id	Int	10	Id of the ad

M_from	Varchar	40	Sender
Date_from	Varchar	30	Date of message
M_to	Varchar	40	Receiver
Date_to	Varchar	30	Date of reply
Message	Varchar	50	Message
Reply	Varchar	50	Reply
Status	Integer	20	Status

Table No. : 07

Table Name : tblFavorites

Primary Key : fav_id

Foreign Key :

Table Description : To store Favorite ads details

Field Name	Data Type	Size	Description
Fav_id	Int	20	Primary key of Favorites table
Ad_id	Int	30	Id of the ad
Ad_title	Varchar	100	Title of ad
Ad_descriptio n	Varchar	1500	Description about ad
Ad_price	Varchar	30	Price of property
Ad_type	Varchar	20	Type of property
Ad_cat	Varchar	20	Category of property
Image1	Varchar	100	Image of property
Ad_loc	Varchar	30	Location of property
Username	Varchar	50	Username of the property owner
status	Int	10	Status

Table No. 08**Table Name : tbl_places****Primary Key : id****Foreign Key :****Table Description : To store location details**

Field Name	Data Type	Size	Description
Id	Int	10	Primary key of place table
name	Varchar	30	Name of the location
district	Varchar	40	District
lat	Double		Latitude of location
long	Double		Longitude of location
status	Int	10	status

Table No. 09**Table Name : tbl_queries****Primary Key : id****Foreign Key :****Table Description : To store customer queries**

Field Name	Data Type	Size	Description
Id	Int	10	Primary key of query table
Name	Varchar	30	Name of the user
Email	Varchar	30	Email of user
Phone	Varchar	20	Phone no. of user
Message	Varchar	200	query
username	Varchar	30	Username
Reply	Varchar	100	Reply to user
status	Int	10	status

Table No. 10

Table Name : tbl_reviews

Primary Key : r_id

Foreign Key :

Table Description : To store customer reviews

Field Name	Data Type	Size	Description
R_id	Int	10	Primary key of review table
Name	Varchar	30	Name of the user
Email	Varchar	30	Email of user
C_date	Varchar	20	Date of review
Review	Varchar	200	Review
Rating	Varchar	20	Rating
Avg_rating	Varchar	20	Average Rating
Agent_id	Int	30	Id of the agent
status	Int	10	status

Table No. 11

Table Name : tbl_appointment

Primary Key : id

Foreign Key :

Table Description : To store Appointment details

Field Name	Data Type	Size	Description
Id	Int	10	Primary key of review table
Ad_id	Varchar	30	Id of Ad
User	Varchar	30	Email of user
Date	Varchar	20	Date
Time	Varchar	200	Time
Message	Varchar	20	Message
Seller_email	Varchar	20	Seller Email
Appointment_status	Int	30	Appointment status

Status	Int	10	status
---------------	-----	----	--------

Table No. : 12**Table Name : tbl_loanapplication****Primary Key : id****Foreign Key :****Table Description : To store Home loan application details**

Field Name	Data Type	Size	Description
Id	Int	10	Primary key of review table
Name	Varchar	30	Id of Ad
Address	Varchar	30	Email of user
State	Varchar	20	Date
District	Varchar	200	Time
Mobile	Varchar	20	Message
email	Varchar	20	Seller Email
age	Int	30	Appointment status
occupation	Int	10	status
Loan_amt	Varchar	30	Loan Amount
Prop_des	Varchar	30	Property Description
Bank	Varchar	30	Bank
Income	Varchar	30	Monthly Income
Date	Varchar	20	Date
User_id	Varchar	30	User id
Loan_status	Varchar	20	Loan Status
status	Int	20	Status

Table No. : 13**Table Name : tbl_payments****Primary Key : id****Foreign Key :****Table Description : To store Payment details**

Field Name	Data Type	Size	Description
Id	Int	10	Primary key of review table
Name	Varchar	30	Name of user
Amount	Varchar	30	Amount paid
Payment status	Varchar	20	Payment status
Added_on	Varchar	200	Payment date

Table No. : 14

Table Name : tbl_reviews

Primary Key : r_id

Foreign Key :

Table Description : To store Agent Review details

Field Name	Data Type	Size	Description
R_id	Int	10	Review Id
Name	Varchar	30	Name of user
email	Varchar	30	Email of user
C_date	Varchar	20	Date of review
review	Varchar	200	Review
rating	Varchar	30	Rating
Agent_id	Int	20	Agent id
status	Int	20	status

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- ❖ Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code were removed and ensured that all modules are working, and gives the expected result.

5.2.1.1 Test Case

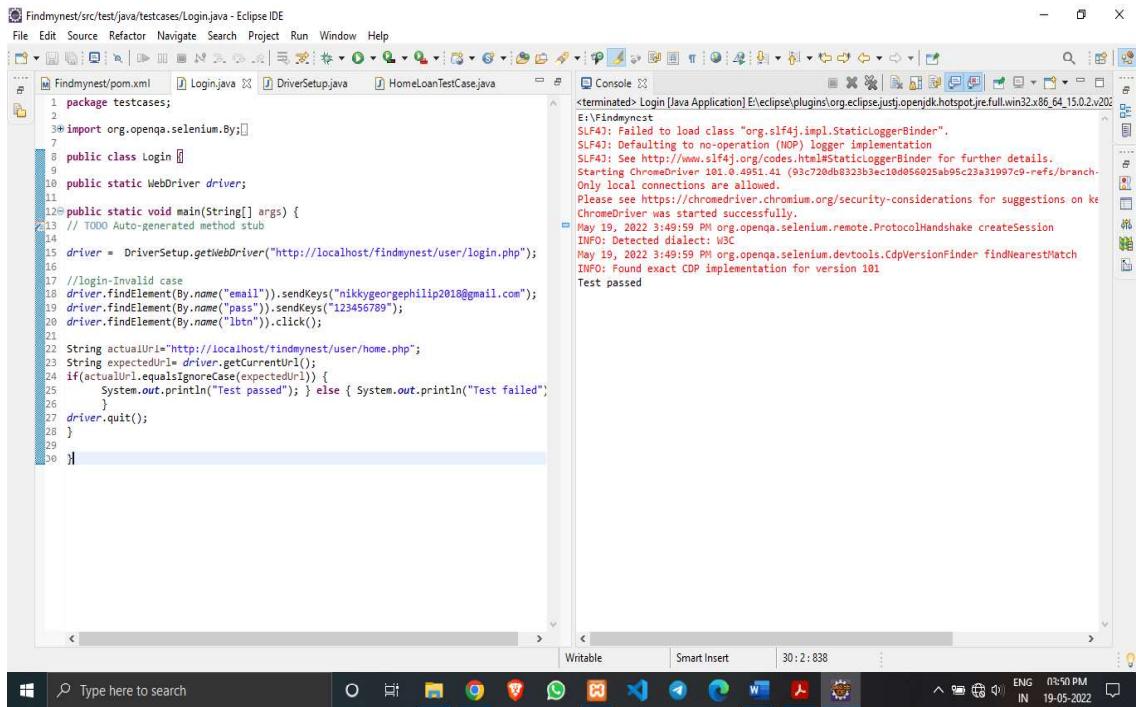
Test Case 1					
Project Name: Findmynest- A Real Estate Management System					
Login Test Case					
Test Case ID: Test 1	Test Designed By: Nikky George Philip				
Test Priority(Low/Medium/High): High	Test Designed Date: 17-05-2022				
Module Name: Login Screen	Test Executed By : Jinson Devis				
Test Title : Verify login with valid email and password	Test Execution Date: 18-05-2022				
Description: Test the Login Page					
Pre-Condition : User has valid email id and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigation to Login Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid Email Id	User Name: nikkygeorge philip2018 @gmail.com	User should be able to Login	User Logged in and navigated to Home Page	Pass
3	Provide Valid Password	Password: 123456789			
4	Click on Sign In button				
5	Provide Invalid Email Id or password	Email Id: user@gmail.com Password: User1234	User should not be able to Login	Message for enter valid email id or password displayed	Fail
6	Provide Null Email Id or Password	Email Id: null Password: null			
7	Click on Sign In button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

Code

```

package testcases;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import chromedriver.DriverSetup;
public class Login {
    public static WebDriver driver;
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        driver = DriverSetup.getWebDriver("http://localhost/findmynest/user/login.php");
        //login-Invalid case
        driver.findElement(By.name("email")).sendKeys("nikkygeorgephilip2018@gmail.com");
        driver.findElement(By.name("pass")).sendKeys("123456789");
        driver.findElement(By.name("lbtn")).click();
        String actualUrl="http://localhost/findmynest/user/home.php";
        String expectedUrl= driver.getCurrentUrl();
        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
            System.out.println("Test passed"); } else { System.out.println("Test failed");
        }
        driver.quit();
    }
}

```



5.2.1.2 Test Case

Test Case 2					
Project Name: Findmynest- A Real Estate Management System					
Home Loan Test Case					
Test Case ID: Test 2	Test Designed By: Nikky George Philip				
Test Priority(Low/Medium/High): High	Test Designed Date: 17-05-2022				
Module Name: Home Loan Screen	Test Executed By : Jinson Devis				
Test Title : Apply for Home Loan	Test Execution Date: 18-05-2022				
Description: Test the Home Loan Page					
Pre-Condition : Enter correct details					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigation to Home Loan Page		Loan Application must be submitted	Home Loan Application submitted	Pass
2	Provide Name	Name: Nikky George Philip	Loan Application must be submitted	Loan Application must be submitted Successfully	Pass
3	Provide Address	Address: Thenakara			
4	Provide State	State: Kerala			
5	Provide District	District: Kottayam			
6	Provide Mobile	Mobile: 7594959975			
7	Provide Email	Email: nikkygeorge philip2018 @gmail.co m			
8	Provide Age	Age: 25			
9	Provide Occupation	Occupation: Business			
10	Provide Loan Amount	Loan Amount: 2500000			
11	Provide Property Address	Property Address: Thenakara			
12	Provide Nearest bank	Nearest Bank:			

		Axis Bank			
13	Provide Monthly income	Monthly Income: 30000			
14	Click on Submit Application button				
Post-Condition: Loan application is submitted successfully. The Loan Application is saved in the database.					

Code

```

package testcases;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;

import chromedriver.DriverSetup;

public class HomeLoanTestCase {

    public static WebDriver driver;

    public static void main(String[] args) throws InterruptedException {
        // TODO Auto-generated method stub

        driver = DriverSetup.getWebDriver("http://localhost/findmynest/user/login.php");

        driver.findElement(By.name("email")).sendKeys("nikkygeorgephilip2018@gmail.com");
        driver.findElement(By.name("pass")).sendKeys("123456789");
        driver.findElement(By.name("lbtn")).click();

        driver.get("http://localhost/findmynest/user/homeloan.php");
        //login-Invalid case
        driver.findElement(By.name("name")).sendKeys("Nikky George Philip");
        driver.findElement(By.name("address")).sendKeys("Thenakara");
        driver.findElement(By.name("state")).sendKeys("Kerala");
        driver.findElement(By.name("district")).sendKeys("Kottayam");
        driver.findElement(By.name("mobile")).sendKeys("7594959975");
        driver.findElement(By.name("email")).sendKeys("nikkygeorgephilip2018@gmail.com");
        driver.findElement(By.name("age")).sendKeys("25");
        driver.findElement(By.name("occu")).sendKeys("Business");
        driver.findElement(By.name("loan_amt")).sendKeys("2500000");
        driver.findElement(By.name("prop_des")).sendKeys("Thenakara");
        driver.findElement(By.name("bank")).sendKeys("Axis Bank");
        driver.findElement(By.name("income")).sendKeys("30000");
        Thread.sleep(5000);
        driver.findElement(By.name("loan_btn")).click();

        String actualUrl="http://localhost/findmynest/user/homeloan.php";
        String expectedUrl= driver.getCurrentUrl();
        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
    
```

```

        System.out.println("Test Passed"); } else { System.out.println("Test Failed");

    }
driver.quit();
}

}

```

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows files like Findmynest/pom.xml, Login.java, DriverSetup.java, and HomeLoanTestCase.java.
- Code Editor:** Displays the Java code for HomeLoanTestCase.java.
- Console Tab:** Shows the execution logs and the test result.

```

Findmynest/src/test/java/testcases/HomeLoanTestCase.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Findmynest/pom.xml Login.java DriverSetup.java HomeLoanTestCase.java
8 public class HomeLoanTestCase {
9     public static WebDriver driver;
10    public static void main(String[] args) throws InterruptedException {
11        driver = DriverSetup.getWebDriver("http://localhost/findmynest/user/login.php");
12        driver.findElement(By.name("email")).sendKeys("nikkygeorgephilip2018@gmail.com");
13        driver.findElement(By.name("pass")).sendKeys("123456789");
14        driver.findElement(By.name("btn")).click();
15        driver.get("http://localhost/findmynest/user/homeloan.php");
16        //log in
17        driver.findElement(By.name("name")).sendKeys("Nikky George Philip");
18        driver.findElement(By.name("address")).sendKeys("Thenakara");
19        driver.findElement(By.name("state")).sendKeys("Kerala");
20        driver.findElement(By.name("district")).sendKeys("Kottayam");
21        driver.findElement(By.name("mobile")).sendKeys("7594959975");
22        driver.findElement(By.name("email")).sendKeys("nikkygeorgephilip2018@gmail.com");
23        driver.findElement(By.name("age")).sendKeys("25");
24        driver.findElement(By.name("occu")).sendKeys("Business");
25        driver.findElement(By.name("loan_amt")).sendKeys("2500000");
26        driver.findElement(By.name("prop_des")).sendKeys("Thenakara");
27        driver.findElement(By.name("bank")).sendKeys("Axis Bank");
28        driver.findElement(By.name("income")).sendKeys("30000");
29        Thread.sleep(5000);
30        driver.findElement(By.name("loan_btn")).click();
31        String actualUrl="http://localhost/findmynest/user/homeloan.php";
32        String expectedUrl= driver.getCurrentUrl();
33        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
34            System.out.println("Test failed"); } else { System.out.println("Test Passed");
35        }
36    }
37    driver.quit();
38 }
39 }
40 }
41 }
42 }
43 }
44 }
45 }
46 }

```

The console output shows the test passed:

```

<terminated> HomeLoanTestCase [Java Application] E:\eclipse\plugins\org.eclipse.jdt\openjdk.hotspot.jre.full.win32.x86,
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 101.0.4951.41 (03c720db9323b3a10d056025ab95c23a31097c9-rev/branch-Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on ke
May 19, 2022 4:25:59 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 19, 2022 4:25:59 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
Test Passed

```

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to

ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. The proposed system introduces facility for customer to approach agents to sell their property. Provides lots of advantages like Finding an agent nearest to their location, post ads, post requirements, enquire about ad, adding ads to favorites lists and may more.

7.2 FUTURE SCOPE

- The proposed system is designed in such a way that the payment should be done in online mode.
- Customers can able to do advanced search options
- Customers can able to add complaints and feedbacks etc.
- Data security can be enhanced.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.
- Roger S Pressman, “*Software Engineering*”, 1994.
- Pankaj Jalote, “*Software engineering: a precise approach*”, 2006.
- James lee and Brent ware Addison, “Open source web development with LAMP”, 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES:

- www.w3schools.com
- www.jquery.com
- <http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf>
- www.agilemodeling.com/artifacts/useCaseDiagram.html

CHAPTER 9

APPENDIX

9.1 Sample Code

Customer

Postads.php

```
<?php
session_start();
error_reporting(0);
date_default_timezone_set("Asia/Kolkata");
$c_date = date("d/m/Y");
include 'header1.php';
include 'connect.php';
$query = "SELECT * FROM tbl_places";
$r = mysqli_query($conn,$query);
?>
<section class="py-5 my-5">
    <div class="container">
        <div class="bg-white shadow rounded-lg d-block d-sm-flex">
            <div class="profile-tab-nav1">
                <h3 class="mb-4 ml-5 mt-5">Tell us about your Ad</h3>
                <center></center>
                <div class="col-md-12 ml-2 mt-2 mr-1">
                    <h4>Free ADS:-</h4>
                    <h6><i class="fa fa-check-square"></i>&ampnbsp&ampnbspNo extra charges.</h6>
                    <h6><i class="fa fa-check-square"></i>&ampnbsp&ampnbsp3 months validity.</h6>
                    <h6><i class="fa fa-minus-square"></i>&ampnbsp&ampnbsp24x7 Support.</h6>
                    <h6><i class="fa fa-minus-square"></i>&ampnbsp&ampnbspFree Maintenance.</h6><br>
                    <h4>Featured ADS:-</h4>
                    <h6><i class="fa fa-check-square"></i>&ampnbsp&ampnbspCost Applicable.</h6>
                    <h6><i class="fa fa-check-square"></i>&ampnbsp&ampnbspLifetime validity.</h6>
                    <h6><i class="fa fa-check-square"></i>&ampnbsp&ampnbsp24x7 Support.</h6>
                    <h6><i class="fa fa-check-square"></i>&ampnbsp&ampnbspFree Maintenance.</h6><br>
                </div>
            </div>
            <div class="tab-content p-4 p-md-5" id="v-pills-tabContent">
                <div class="tab-pane fade show active" id="account" role="tabpanel" aria-labelledby="account-tab">
                    <form action="#" method="POST" class="row g-3" enctype="multipart/form-data" autocomplete="off">
                        <div class="row">
                            <div class="col-md-12">
```

```

<div class="form-group">
    <label>Title</label>
    <input type="text" id="fname" class="form-control" name="title" >
</div>
<span id="name" class="er"><font size="2">Please enter a valid title,
Numbers not allowed!!</font></span>

</div>
<div class="col-md-12">
    <div class="form-group">
        <label>Description</label>
        <textarea class="form-control" id="des" name="des"
placeholder="Describe about your ad" rows="4"></textarea>
    </div>
    <span id="name1" class="er"><font size="2">Please enter a valid
Description, Numbers not allowed!!</font></span>
    </div>
    <div class="col-md-6">
        <div class="form-group">
            <label for="inputCity" class="form-label">Category</label>
            <select class="form-control" name="category" required>
                <option value="">Select Category </option>
                <option value="Rent">Rent</option>
                <option value="Sale">Sale</option>
            </select>
        </div>
        </div>
        <div class="col-md-6">
            <div class="form-group">
                <label for="inputState" class="form-label">Property Type</label>
                <select class="form-control" name="type" required>
                    <option value="">Select Type </option>
                    <option value="House">House</option>
                    <option value="Villa">Villa</option>
                    <option value="Apartment">Apartment</option>
                </select>
            </div>
        </div>
        <div class="col-md-4">
            <div class="form-group">
                <label for="inputCity" class="form-label">Price</label>
                <input type="text" class="form-control" id="price" name="price"
required="required">
            </div>
            <span id="eprice" class="er"><font size="2">Letters not
allowed!!</font></span>
        </div>
        <div class="col-4">

```

```

<div class="form-group">
<label for="inputAddress2" class="form-label">Location</label>
<select class="form-control" name="location" required>
    <option value="">Select One </option>
    <?php
        while($row = mysqli_fetch_array($r))
        {
            echo('<option value='.$row['name'].'>'.$row['name'].'</option>');
        }
    ?>
</select>
</div>
</div>
<div class="col-md-4">
    <div class="form-group">
        <label for="inputCity" class="form-label">Area Sq.ft</label>
        <input type="text" class="form-control" id="area" name="area" required="required">
    </div>
    <span id="earea" class="er"><font size="2">Letters not allowed!!</font></span>
</div>
<div class="col-4">
    <div class="form-group">
        <label for="inputAddress2" class="form-label">Rooms</label>
        <select class="form-control" name="rooms" required>
            <option value="">Select One </option>
            <option value="1">1</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
            <option value="5">5</option>
            <option value="6">6</option>
            <option value="7">7</option>
        </select>
    </div>
</div>
<div class="col-4">
    <div class="form-group">
        <label for="inputAddress2" class="form-label">Bathrooms</label>
        <select class="form-control" name="bathrooms" required>
            <option value="">Select One </option>

            <option value="1">1</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
            <option value="5">5</option>
            <option value="6">6</option>
            <option value="7">7</option>
        </select>
    </div>
</div>

```

```

        </select>
    </div>
</div>
<div class="col-4">
    <div class="form-group">
        <label for="inputAddress2" class="form-label">Bed Rooms</label>
        <select class="form-control" name="bedrooms" required>
            <option value="">Select One </option>
            <option value="1">1</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
            <option value="5">5</option>
            <option value="6">6</option>
            <option value="7">7</option>
        </select>
    </div>
</div>
<div class="col-md-4">
    <div class="form-group">
        <label for="inputCity" class="form-label">Image 1</label>
        <input type="file" accept="image/*" onchange="loadFile1(event)">
    </div>
</div>
<div class="col-md-4">
    <div class="form-group">
        <label for="inputCity" class="form-label">Image 2</label>
        <input type="file" accept="image/*" onchange="loadFile2(event)">
    </div>
</div>
<div class="col-md-4">
    <div class="form-group">
        <label for="inputCity" class="form-label">Image 3</label>
        <input type="file" accept="image/*" onchange="loadFile3(event)">
    </div>
</div>
<div class="col-md-4">
    <img id="output1" width="120px" height="100px" />
</div>
<div class="col-md-4">
    <img id="output2" width="120px" height="100px" />
</div>
<div class="col-md-4">
    <img id="output3" width="120px" height="100px" />
</div>

```

```

        </div>
        <input type="text" name="feature" value="no" hidden>
        <input type="text" name="createdDate" value="<?php echo $c_date; ?>" hidden>
    <div class="col-12 mt-4">
        <button type="submit" name="post_ad_btn" id="lbtn" class="btn-primary">Post Your Ad</button>
        <button class="btn btn-light" onclick="goBack()">Cancel</button>
    </div>
    </form>
</div>
</div>
</div>
</div>
</section>
<script>
var loadFile1 = function(event) {
    var reader = new FileReader();
    reader.onload = function(){
        var output = document.getElementById('output1');
        output.src = reader.result;
    };
    reader.readAsDataURL(event.target.files[0]);
};

var loadFile2 = function(event) {
    var reader = new FileReader();
    reader.onload = function(){
        var output = document.getElementById('output2');
        output.src = reader.result;
    };
    reader.readAsDataURL(event.target.files[0]);
};

var loadFile3 = function(event) {
    var reader = new FileReader();
    reader.onload = function(){
        var output = document.getElementById('output3');
        output.src = reader.result;
    };
    reader.readAsDataURL(event.target.files[0]);
};
</script>
<script type="text/javascript">
function goBack()
{
    window.location.href = "ad_requests.php";
}
</script>

```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/jquery-validation@1.19.3/dist/jquery.validate.js"></script>
<script src="../assets/js/jquery_validator.js"></script>
<?php
include 'useradController.php';
include 'footer.php';
?>
```

Agent_finder.php

```
<?php
session_start();
error_reporting(0);
$email = $_SESSION['email'];
include "../config.php";
include "header1.php";
$query2 = "SELECT * FROM tbl_agents WHERE status = 1";
$res2 = mysqli_query($conn,$query2);
$query1 = "SELECT * FROM tbl_places WHERE status = 1";
$res1 = mysqli_query($conn,$query1);
$row1 = mysqli_fetch_array($res1);
?>
<form action="#" method="post">
<div class="container bg-white shadow round px-3 py-3 mt-5">
<h3><font color="#006aff">Find Agents Nearest to you</font></h3>
<div class="row">
<div class="col-sm">
<h6>Location</h6>
</div>
<div class="col-sm">
</div>
</div>
<div class="row">
<div class="col-sm">
<select class="form-control" name="loc" required>
<?php
while($row1 = mysqli_fetch_array($res1))
{
echo('<option value="'.$row1['name'].'">'.$row1['name'].'</option>');
}
?>
</select>
</div>
<div class="col-sm">
<input type="submit" class="btn btn-primary px-5" name="fsearch" value="Search">
</div>
</div>
</div>
</form>
```

```

<section class="grids-4" id="properties" tabindex="0" style="outline:none;">
<div id="grids4-block" class="py-5">
<div class="container py-md-3">
<div class="row col-12">
<?php
if(isset($_POST['fsearch']))
{
    $loc = $_POST['loc'];
    $rating = $_POST['rating'];
    $query = "SELECT * FROM tbl_agents WHERE place = '$loc' and status = 1";
    $res = mysqli_query($conn,$query);
    while($row = mysqli_fetch_array($res))
    {
        $query3 = "SELECT * FROM tbl_reviews WHERE agent_id = ".$row['agent_id']."";
        $res3 = mysqli_query($conn,$query3);
        $r = mysqli_num_rows($res3);
        echo(
            <div class="col-4 mt-3">
                <div class="card horizontal shadow round border-1-red">
                    <div class="row">
                        <div class="col-w-25">
                            <div class="card-image pl-4 mt-2"></div>
                        </div>
                        <div class="col p-3">
                            <div class="card-stacked">
                                <div class="card-content">
                                    <h5>$row['name'].</h5>
                                    <p class="flow-text">Email ID: '$row['email'].</p>
                                    <p class="flow-text">Location: '$row['place'].</p>
                                    <p class="flow-text"><font color="#006aff">('.$r.') Reviews</font></p>
                                </div>
                                <div class="card-action float-right mr-2"><a href="agent_profile.php?id='".$row['agent_id']."' class="btn btn-primary mx-1 p-1">View Profile</a></div>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            );
        }
    }
else
{
    while ($row2 = mysqli_fetch_array($res2)) {
        $query3 = "SELECT * FROM tbl_reviews WHERE agent_id =
        ".$row2['agent_id']."";
}

```

```

$res3 = mysqli_query($conn,$query3);
$r = mysqli_num_rows($res3);
echo(
    <div class="col-4 mt-3">
        <div class="card horizontal shadow round border-1-red">
            <div class="row">
                <div class="col-w-25">
                    <div class="card-image pl-4 mt-2"></div>
                </div>
                <div class="col p-3">
                    <div class="card-stacked">
                        <div class="card-content">
                            <h5>' . $row2['name'] . '</h5>
                            <p class="flow-text">Email ID: ' . $row2['email'] . '</p>
                            <p class="flow-text">Location: ' . $row2['place'] . '</p>
                            <p class="flow-text"><font color="#006aff" style="font-size: 1.2em;">(' . $r . ') Reviews

```

Agent

Approve_Ad.php

```

<?php
include 'header.php';
$user = $_SESSION['email'];
$query = "SELECT * FROM tbl_ads WHERE seller_email = '$user' and status = 2";

```

```

$res = mysqli_query($conn,$query);
?>
<div class="heading text-center mx-auto mt-5">
    <h3 class="head">Approved Ads</h3>
</div>
<section class="py-5 my-5 mt-n3">
    <div class="container">
        <div class="bg-white shadow rounded-lg">
            <?php
                if(mysqli_num_rows($res) > 0)
                {
                    while($row = mysqli_fetch_array($res))
                    {
                        echo(
                            <div class="container-fluid">
                                <div class="row">
                                    <div class="col-11 ml-5 mt-3">
                                        <div class="card">
                                            <div class="card-horizontal">
                                                <div class="img-square-wrapper">
                                                    
                                                </div>
                                            <div class="card-body">
                                                <h4 class="card-title mt-n3">' . $row['ad_title'] . '</h4>
                                                <h4 class="card-title1">' . $row['ad_price'] . '</h4>
                                                <p class="card-text mb-3">' . $row['ad_description'] . '</p>
                                                <p class="card-text mb-3"><span class="fa fa-
bed"></span>&nbsp;&nbsp;' . $row['bedroom'] . ' Beds
&nbsp;&nbsp;<span class="fa fa-bath"></span>&nbsp;&nbsp;
' . $row['bathroom'] . ' Baths
&nbsp;&nbsp;<span class="fa fa-share-square-o"></span>&nbsp;&nbsp;
' . $row['area'] . ' sq ft</p>
                                            <a class="btn1 btn-primary1 text-light">For
' . $row['ad_cat'] . '</a>&nbsp;&nbsp;&nbsp;
                                            <a class="btn1 btn-warning text-light">' . $row['ad_type'] . '</a>
                                        </div>
                                    </div>
                                </div>
                            </div>
                        );
                    }
                }
            echo('<center><font color="red" size="3">No Ads Approved!!</font></center>');
        
```

```

    }
?>

<br><br>
</div>
</div>
</section>
<br>

<br> <br>
<br> <br>
<br>
<?php
include 'footer.php';
?>
```

HomeLoan.php

```

<?php
session_start();
error_reporting(0);
date_default_timezone_set("Asia/Kolkata");
$c_date = date("d/m/Y");
include 'header1.php';
?>
<div class="container1">
<div id="hide_div">
<section class="py-5 my-3">

<center><h3 class="mb-3 mt-n4">Home Loan Application</h3></center>
<div class="container">
    <div class="bg-white shadow rounded-lg d-block d-sm-flex">
        <div class="profile-tab-nav1">
            <h4 class="mb-4 ml-5 mt-5">Applicant Details for applying loan</h4>
            <center></center>
        </div>
        <div class="tab-content p-4 p-md-5" id="v-pills-tabContent">
            <div class="tab-pane fade show active" id="account" role="tabpanel" aria-labelledby="account-tab">
                <form action="#" method="POST" class="row g-3" enctype="multipart/form-data" autocomplete="off">
                    <div class="row">
                        <div class="col-md-9">
                            <div class="form-group">
                                <label>Name</label>
                                <input type="text" id="fname" class="form-control" name="name" >
                            </div>
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>
</section>
</div>
</div>
```

```

<span id="name" class="er"><font size="2">Please enter a valid name, Numbers
not allowed!!</font></span>

</div>
<div class="col-md-9">
    <div class="form-group">
        <label>Address</label>
        <textarea class="form-control" id="des" name="address"
rows="4"></textarea>
    </div>
</div>
<div class="col-md-5">
    <div class="form-group">
        <label for="inputCity" class="form-label">State</label>
        <select class="form-control" name="state" required>
            <option value="">Select</option>
            <option value="Kerala">Kerala</option>
        </select>
    </div>
</div>
<div class="col-md-4">
    <div class="form-group">
        <label for="inputState" class="form-label">District</label>
        <select class="form-control" name="district" required>
            <option value="">Select</option>
            <option value="Alappuzha">Alappuzha</option>
            <option value="Ernakulam">Ernakulam</option>
            <option value="Idukki">Idukki</option>
            <option value="Kannur">Kannur</option>
            <option value="Kasargod">Kasargod</option>
            <option value="Kollam">Kollam</option>
            <option value="Kottayam">Kottayam</option>
            <option value="Kozhikode">Kozhikode</option>
            <option value="Malappuram">Malappuram</option>
            <option value="Palakkad">Palakkad</option>
            <option value="Pathanamthitta">Pathanamthitta</option>
            <option value="Trivandrum">Trivandrum</option>
            <option value="Thrissur">Thrissur</option>
            <option value="Wayanad">Wayanad</option>
        </select>
    </div>
</div>
<div class="col-md-9">
    <div class="form-group">
        <label for="inputCity" class="form-label">Mobile Number</label>
        <input type="text" class="form-control" id="phone" maxlength="10"
name="mobile" required="required">
    </div>
    <span id="ephone" class="er"><font size="2">Please enter a valid phone

```

```

number!!</font></span>
    </div>
    <div class="col-9">
        <div class="form-group">
            <label for="inputAddress2" class="form-label">Email</label>
            <input type="text" class="form-control" id="email" name="email"
required="required">
        </div>
        <span id="error" class="er"><font size="2">Please enter a valid
email!!</font></span>
    </div>
    <div class="col-md-9">
        <div class="form-group">
            <label for="inputCity" class="form-label">Age</label>
            <input type="text" class="form-control" id="area" name="age"
required="required">
        </div>
        <span id="earea" class="er"><font size="2">Letters not
allowed!!</font></span>
    </div>
    <div class="col-9">
        <div class="form-group">
            <label for="inputAddress2" class="form-label">Occupation</label>
            <input type="text" class="form-control" id="fname" name="occu"
required="required">
        </div>
        <span id="name" class="er"><font size="2">Numbers not
allowed!!</font></span>
    </div>
    <div class="col-9">
        <div class="form-group">
            <label for="inputAddress2" class="form-label">Loan Amount</label>
            <input type="text" class="form-control" id="area" name="loan_amt"
required="required">
        </div>
        <span id="earea" class="er"><font size="2">Letters not
allowed!!</font></span>
    </div>
    <div class="col-9 mt-3 mb-4">
        <h4><b>Property Details</b></h4>
    </div>
    <div class="col-md-9">
        <div class="form-group">
            <label for="inputCity" class="form-label">Please fill the complete address of
the property</label>
            <textarea class="form-control" id="des" name="prop_des"
rows="4"></textarea>
        </div>
    </div>
    <div class="col-md-9">

```

```

<div class="form-group">
    <label for="inputCity" class="form-label">Please select nearest bank</label>
    <select class="form-control" name="bank" required>
        <option value="">Select</option>
        <option value="Axis Bank">Axis Bank</option>
        <option value="Bank Of India">Bank Of India</option>
        <option value="Canara Bank">Canara Bank</option>
        <option value="Catholic Syrian Bank Ltd">Catholic Syrian Bank
        Ltd</option>
        <option value="Central Bank Of India">Central Bank Of India</option>
        <option value="Citibank">Citibank</option>
        <option value="City Union Bank Ltd">City Union Bank Ltd</option>
        <option value="Hdfc Bank Ltd">Hdfc Bank Ltd</option>
        <option value="Icici Bank Ltd">Icici Bank Ltd</option>
        <option value="Kotak Mahindra Bank">Kotak Mahindra Bank</option>
        <option value="South Indian Bank">South Indian Bank</option>
        <option value="State Bank Of India">State Bank Of India</option>
        <option value="The Federal Bank Ltd">The Federal Bank Ltd</option>
        <option value="Union Bank Of India">Union Bank Of India</option>
    </select>
</div>
</div>
<div class="col-9 mt-3 mb-4">
    <h4><b>Income Details</b></h4></b>
</div>
<div class="col-md-9">
    <div class="form-group">
        <label for="inputCity" class="form-label">Net monthly income</label>
        <input type="text" class="form-control" id="area" name="income"
required="required">
    </div>

    <span id="earea" class="er"><font size="2">Letters not
allowed!!</font></span>
    </div>
    <input type="text" name="feature" value="no" hidden>
    <input type="text" name="createdDate" value="<?php echo $c_date; ?>">
hidden>
    <div class="col-12 mt-4">
        <button type="submit" name="loan_btn" id="lbtn" class="btn-
primary">Submit Application</button>
        <button class="btn btn-light" onclick="goBack()">Cancel</button>
    </div>
    </form>
</div>
</div>
</div>
</div>
</section>
</div>

```

```

<div id="show_div" style="display:none">
<div class="container">
  <div class="alert alert-success mx-5 w-75 p-4 my-3 mb-6 mt-5" role="alert">
    <h4 class="alert-heading">Success!</h4>
    <p>Your Loan Application has been submitted successfully</p>
  </div>
</div>
</div>
<div id="error_div" style="display:none">
<div class="container">
  <div class="alert alert-danger mx-5 w-75 p-4 my-3 mb-6 mt-5" role="alert">
    <h4 class="alert-heading">Alert!</h4>
    <p>You have already submitted a loan application, wait till the application been processed.</p>
  </div>
</div>
</div>
</div>
<script>
  var loadFile1 = function(event) {
    var reader = new FileReader();
    reader.onload = function(){
      var output = document.getElementById('output1');
      output.src = reader.result;
    };
    reader.readAsDataURL(event.target.files[0]);
  };
  var loadFile2 = function(event) {
    var reader = new FileReader();
    reader.onload = function(){
      var output = document.getElementById('output2');
      output.src = reader.result;
    };
    reader.readAsDataURL(event.target.files[0]);
  };
  var loadFile3 = function(event) {
    var reader = new FileReader();
    reader.onload = function(){
      var output = document.getElementById('output3');
      output.src = reader.result;
    };
    reader.readAsDataURL(event.target.files[0]);
  };
</script>
<script type="text/javascript">
  function goBack()
  {
    window.location.href = "home.php";
  }
</script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

```

```
<script src="https://cdn.jsdelivr.net/npm/jquery-validation@1.19.3/dist/jquery.validate.js"></script>
<script src="../assets/js/jquery_validator.js"></script>
<?php
include 'userloanController.php';
include 'footer.php';

?>
```

Agent Finder.php

```
<?php
session_start();
error_reporting(0);
$email = $_SESSION['email'];
include "../config.php";
include "header1.php";
$query2 = "SELECT * FROM tbl_agents WHERE status = 1";
$res2 = mysqli_query($conn,$query2);
$query1 = "SELECT * FROM tbl_places WHERE status = 1";
$res1 = mysqli_query($conn,$query1);
$row1 = mysqli_fetch_array($res1);
?>
<form action="#" method="post">
<div class="container bg-white shadow round px-3 py-3 mt-5">
<h3><font color="#006aff"><b>Find Agents Nearest to you</b></font></h3>
<div class="row">
<div class="col-sm">
<h6>Location</h6>
</div>

<div class="col-sm">
<h6>Rating</h6>
</div>
</div>
<div class="row">
<div class="col-sm">
<select class="form-control" name="loc">
<option value="">Select One</option>
<?php
while($row1 = mysqli_fetch_array($res1))
{
    echo('<option value="'.$row1['name'].'">'.$row1['name'].'</option>');
}
?>
</select>
</div>
<div class="col-sm">
```

```

<select class="form-control" name="rate">
    <option value="">Select one</option>
    <option value="4.5">> 4.5</option>
    <option value="4">> 4.0</option>
    <option value="3.5">> 3.5</option>
    <option value="3">> 3.0</option>
    <option value="2.5">> 2.5</option>
    <option value="2">> 2.0</option>
    <option value="1.5">> 1.5</option>
    <option value="1">> 1.0</option>
</select>
</div>
<div class="col-sm">
<input type="submit" class="btn btn-primary px-5" name="fsearch" value="Search">
</div>
</div>
</div>
</form>
<section class="grid-4" id="properties" tabindex="0" style="outline:none;">
<div id="grid4-block" class="py-3">
<div class="container py-md-3">
    <div class="row col-12">
<?php
    if(isset($_POST['fsearch']))
    {
        $loc = $_POST['loc'];
        $rating = $_POST['rate'];
        $query = "SELECT * FROM tbl_agents WHERE place = '$loc' AND avg_rating > '$rating'
AND status = 1";
        $res = mysqli_query($conn,$query);

        if(mysqli_num_rows($res)>0)
        {
            while($row = mysqli_fetch_array($res))
            {
                $query3 = "SELECT * FROM tbl_reviews WHERE agent_id = ".$row['agent_id'].'";
                $query4 = "SELECT avg(rating) as avg_rating FROM tbl_reviews WHERE agent_id =
".$row['agent_id']."'";

                <h5>' . $row['name'] . '</h5>
                <p class="flow-text">Email ID: ' . $row['email'] . '</p>
                <p class="flow-text">Location: ' . $row['place'] . '</p>
                <p class="flow-text"><font color="#006aff">' . $row['avg_rating'] . ' Reviews</font></p>
            }
        }
    }
</div>
</div>
</section>
<?php
    include 'footer.php';
?>

```

9.2 Screen Shots

CUSTOMER PAGES

Home page

The screenshot shows the Findmynest Home page with four house listings for sale:

- Rs. 8500000**: House For Sale In Mundakkayam, Kottayam. Features: 4 Beds, 4 Baths, 3600 sq ft.
- Rs. 25500000**: House For Sale, Featured, Mundakkayam. Features: 4 Beds, 4 Baths, 3600 sq ft.
- Rs. 9500000**: House For Sale, Featured, Kottayam. Features: 4 Beds, 4 Baths, 3600 sq ft.
- Rs. 9000000**: Good looking House For Sale, Featured, Mundakkayam. Features: 4 Beds, 4 Baths, 3600 sq ft.

Agentfinder Page

The screenshot shows the Findmynest Agentfinder Page with a list of agents:

Agent Profile	Email ID	Location	Rating	Action
Nikky	nikky@gmail.com	Kottayam	1 Reviews ★★★★★	View Profile
Athul	athuls@gmail.com	Erumely	2 Reviews ★★★★★	View Profile
Ligin	ligin@gmail.com	Kottayam	1 Reviews ★★★★★	View Profile
Ajith	ajith@gmail.com	Erumely	0 Reviews ★★★★★	View Profile
Sam	sam@gmail.com	Kottayam	0 Reviews ★★★★★	View Profile
Ananthu	ananthu@gmail.com	Alappuzha	0 Reviews ★★★★★	View Profile

Post Ad Page

Tell us about your Ad

Free ADS:-

- No extra charges.
- 3 months validity.
- 24x7 Support.
- Free Maintenance.

Post Requirement

Tell us about your Requirements

Property Type	Car Parking spaces	
House	2	
Furnished status	Budget Amount	
Semi-furnished	250000	
Preferred Location	Area Sq.ft	
Erumely	2000	
Rooms	Bathrooms	Bed Rooms
4	4	3

Manage Agents Page

The screenshot shows the 'Agent Details' section of the Findmynest Admin interface. On the left, a sidebar menu includes 'Dashboard', 'Manage Agents' (which is selected), 'Manage Ads', 'Manage Requirements', and 'Manage Favourites'. The main content area has a title 'Agent Details' and a breadcrumb 'Home / Agent Details'. Below this is a table titled 'Agent Details' with columns: #, Name, Email, Place, Phone, Address, Status, and Action. Two entries are listed: 'Nikky' (nikky@gmail.com) from Mundakkayam and 'Athul' (athul@gmail.com) from Erumely, both with phone numbers 7594959975 and address Parathodu, status 'Active', and green 'Action' buttons.

#	Name	Email	Place	Phone	Address	Status	Action
1	Nikky	nikky@gmail.com	Mundakkayam	7594959975	Parathodu	Active	
2	Athul	athul@gmail.com	Erumely	7594959975	Parathodu	Active	

Showing 1 to 2 of 2 entries.

Manage Ads Page

The screenshot shows the 'Ad Details' section of the Findmynest Admin interface. The sidebar menu is identical to the previous page. The main content area has a title 'Ad Details' and a breadcrumb 'Home / Ad Details'. Below this is a table titled 'Ad Details' with columns: Title, Price, Location, Area, Username, Status, and Action. Six ads are listed: 'House For Sale' (Price 5000000, Location Kanjirappally, Area 3500, Username nikkygeorge@gmail.com), 'House For Sale' (Price 5000000, Location Erumely, Area 3500, Username nikkygeorge@gmail.com), 'Villa For Rent In Mundakkayam' (Price 10000, Location Mundakkayam, Area 3300, Username athul@gmail.com), 'Villa For Sale' (Price 10000000, Location Punalur, Area 4000, Username ligin@gmail.com), 'House For Sale' (Price 8000000, Location Erumely, Area 3600, Username ligin@gmail.com), and 'Villa For Rent' (Price 30000, Location Kottarakkara, Area 3500, Username ligin@gmail.com). The 'Status' column shows 'Active' for most ads and 'Pending Activation' for the last one. The 'Action' column contains green 'Action' buttons for most ads and red 'Delete' buttons for the last one.

Title	Price	Location	Area	Username	Status	Action
House For Sale	5000000	Kanjirappally	3500	nikkygeorge@gmail.com		
House For Sale	5000000	Erumely	3500	nikkygeorge@gmail.com		
Villa For Rent In Mundakkayam	10000	Mundakkayam	3300	athul@gmail.com		
Villa For Sale	10000000	Punalur	4000	ligin@gmail.com		
House For Sale	8000000	Erumely	3600	ligin@gmail.com	Active	
Villa For Rent	30000	Kottarakkara	3500	ligin@gmail.com	Pending Activation	