AnyRent

Project Report Submitted by

DIYAMOL JACOB

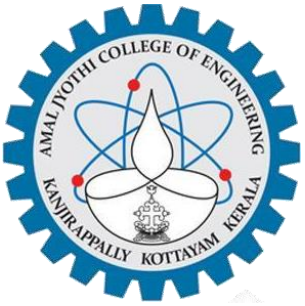
Reg. No.: AJC21MCA-2054

In Partial fulfillment for the Award of the Degree Of

MASTER OF COMPUTER APPLICATIONS

(MCA TWO YEAR)

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]

2022-2023

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**ANYRENT"** is the bonafide work of **DIYAMOL JACOB (Regno:AJC21MCA-2054)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2022-23.

Ms. Merin Chacko

Internal Guide

Ms. Meera Rose Mathew

Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department

DECLARATION

I hereby declare that the project report **"ANYRENT"** is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfillment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2022-2023.

Date: **DIYAMOL JACOB**

KANJIRAPPALLY **Reg: AJC21MCA-2054**

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 express sincere gratitude to my guide **Ms. Merin Chacko** for her 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.

DIYAMOL JACOB

ABSTRACT

AnyRent is a rental portal platform . We mainly focus on people who move to another place for a short period of time and if they need products for a period, we provide them essential needs as rent through this system. It provides services Such as Hiring cars, Bikes, Furniture, Service Apartments or Home. Online shopping websites makes our life simpler. We can buy new things without stepping out of our home. This application is providing some additional features for vendors to edit or delete their products. Customers need to register and place orders in our site and pay his amount. A vendor directly registers into this system and can advertise his own rental product. We are responsible for communication between users and maintain the database.

Some of the core functionalities offered are:

- User Registration

  Each user should have an account to access all the functionalities of website. User can login using login page and email verification and logout using the logout page. All the user sessions will be saved in the database. Users can update their profile. If user forgot their password, we can reset it. Users can post or add their Products for rent as well as they can buy any products as rent. They can manage their profile and can delete posted products. They can communicate through this website.

- Admin Functionality

  Admin has the whole power to manage the system. He has the power to add various categories of products and view the users.

- Search by location

  Users can find products in corresponding location.

CONTENT

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

AnyRent is a rental portal platform. We mainly focus on people who move to another place for a short period of time and if they need products for a period, we provide them essential needs as Rent through this system. It provides services Such as Hiring cars, Bikes, Furniture, Home and Apartments.

1.2 PROJECT SPECIFICATION

AnyRent is a website that helps the users to view, purchase from their current locations or without visiting a store. It helps in providing best services to the users. AnyRent is a rental portal platform. We mainly focus on people who move to another place for a short period of time and if they need products for a period, we provide them essential needs as Rent through this system. It provides services Such as Hiring cars, Bikes, Furniture, Service Apartments or Home. Online shopping websites makes our life simpler. We can buy new things without stepping out of our home. This application is providing some additional features for vendors to edit or delete their products. Users need to register into this system and can post or advertise his own rental Products for rent as well as they can buy any products as rent. We are responsible for communication between users providing a Request to Chat mechanism and maintain the database.

Users

- Admin
- User/Vendor

Functionalities

- User Registration

Each user should have an account to access all the functionalities of website. User can login using login page and email verification and logout using the logout page. All the user sessions will be saved in the database. Users can update their profile. If user forgot their password, we can reset it. Users can post or add their Products for rent as well as they can buy any products as rent. They can manage their profile and can delete posted products. They can communicate through this website.

- Admin Functionality

Admin has the whole power to manage the system. He has the power to add various categories of products and view the users.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

AnyRent is a rental portal platform. We mainly focus on people who move to another place for a short period of time and if they need products for a period, we provide them essential needs as Rent through this system. It provides services Such as Hiring cars, Bikes, Furniture, Service Apartments or Home. Online shopping websites makes our life simpler. We can buy new things without stepping out of our home. This application is providing some additional features for vendors to edit or delete their products. Users need to register and can post or advertise his own rental Products for rent as well as they can buy any products as rent. We are responsible for communication between users providing a Request to Chat mechanism and maintain the database.

2.2 EXISTING SYSTEM

Existing system includes both online and offline shops. Offline systems are not fully automated. In the case of offline method, to rent a products the customer needs to visit the place where the product is available for giving rent. The customer need to make deals with vendor for renting the products. The vendor needs to keep all records as paper works. The proposed system rectifies the drawbacks of the present system. Online market place include Olx, Quikr among several others that provides online marketplace that helps users to sell, buy, rent or discover anything across India. They are efficient in terms of its security, user experience and tracking orders.

2.2.1 NATURAL SYSTEM STUDIED

In existing natural system. the customer need to find out where the product they want for rent is available. The customer needs to make deals with vendor for renting the products. The vendor needs to keep all records as paper works. Moreover, vendor have no platform for them to make advertisement for their product. Due to lack of advertisement the business won't reach to the maximum. In the existing system the records of the products and sales are maintained manually. Due to this the data retrieved is time consuming. Due to human calculation errors occurred. Time consuming: In our current system, all the process are carried out by human so naturally it requires more time and, in that sense, it will require more time to complete transaction.

2.2.2 DESIGNED SYSTEM STUDIED

Online market place include Olx, Quikr among several others that provides online marketplace that helps users to sell, buy, rent or discover anything across India. The OLX group is one of the world's fastest-growing network of trading platforms for 300 million people around the world. As the product gained more users, it witnessed a high recall rate across all categories, including electronics,

household appliances, fashion and transportation. It takes not more than a few seconds to create an interactive advertisement in OLX.in, as it requires a few clicks only. In order to do so, we just have to visit the website, and click on the button to create ad.OLX.in has set up its user interface in such a way that even a kid would have no trouble to create an advertisement in the website, as it takes only a few clicks. The user can use camera of their device to take photos of the product, and then to upload it to the website.OLX.in is not going to disappoint their customers, because it comprises an innumerable amount of audience. One of the reasons for such a wider audience is the availability of service is different languages.

2.3 DRAWBACKS OF EXISTING SYSTEM

- The existing system was not very effective.
- There are many rental systems which are available online. But, they are not providing all products at one place.
- Many of them are not providing effective communication between user and the vendor.
- Also present rental systems restricted to only one vendor means products are supplied only from one rental show room.

2.4 PROPOSED SYSTEM

The Designed system is a web-based system which can be accessed by user/customer from anywhere around the city. The system can offer a greater number of products from User/vendor in different locations. A user/vendor directly registers into this system using this system user interface without any manual approach. The proposed system can accept any type of product for rental, especially house or apartments, cars, bikes and furniture. The system interface support to the vendors to upload their product image into the system. A customer directly interacts with this product image and gets necessary information regarding the rental products and can search products through location and category. Administration play vital role here. Administrator can able to add any categories of products and view the users.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- Simplicity of Creating Advertisements
- Cost free
- Wider Audience
- Intuitive User Interface
- Accept any type of products for rental

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

A feasibility study is a detailed analysis that considers all of the critical aspects of a proposed project in order to determine the likelihood of it success. A feasibility study is an assessment of the practicality of a proposed plan or project. A feasibility study analyzes the viability of a project to determine whether the project or venture is likely to succeed. The study is also designed to identify potential issues and problems that could arise while pursuing the project.

The document provides the feasibility of the proposed system that is being developed and three aspects of the feasibility of a project are considered such as technical, economical, and behavioral during the feasibility study.

3.1.1 Economical Feasibility

Economic feasibility analysis is the most commonly used method for determining the efficiency of a new project. It is also known as cost analysis. It helps in identifying profit against investment expected from a project. Cost and time are the most essential factors involved in this field of study. The economic feasibility study for projects shows all the costs necessary for the project, which are the costs of establishing the project, as well as the size of the investment, estimating the size of the project's profits, and knowing the net profit during a specific period of time. and tools needed for the project.

The proposed system is developed as part of project work, thus there is no manual cost spent for the proposed system. All the required resources were already available, which shows that the system is economically feasible for development. The project was developed at a low cost as it is completely developed using open-source software.

3.1.2 Technical Feasibility

The technical feasibility study can be defined as the study related to all the technical aspects of the project. It includes defining the technical specifications of the product and the size of the project, and preparing the necessary labor schedules for production.

In technical feasibility the following issues are taken into consideration.

- Whether the required technology is available or not
- Whether the required resources are available

3.1.3 Behavioral Feasibility

Behavioral feasibility answers the following questions:

- Whether the proposed system will cause any harm to its users?
- It is proposed system sufficient to support the users?

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.

3.1.4 Feasibility study questionnaire

Questionnaire

A questionnaire or survey is a set of predetermined questions designed to elicit Information about a particular topic. It's a useful method for gaining insights quickly from a large group of people. Surveys enable you to collect data from people wherever they are and are also inexpensive. Because of this, it is a widely used technique for gathering requirements. However, surveys might not be the best tool for analyzing complex issues.

Details of person that you have contacted for data collection?

Name : Mr.Joy Joseph

Phone Number : +91 7025916034

Interview Q&A With Joy Joseph 06/09/2022 through telephone

1. How you are giving proper advertisement about your shop?

   We provide ads in social media and newspaper.

2. How you are delivering the rented vehicle to the customer?

   The customer basically come to our hub. Only for premium customers we provide delivery to their places.

3. What are the documents you collect from customer?

   Photocopy of the aadhar card and driving license and we collect extra money from them as deposit and in the time of return, we give their money.

4. If your customer damages the vehicle, how would you respond?

   I will take compensation from them.

5. How much is the base rent for the vehicle?

   Actually it is based on the vehicle type and the duration of the time they requests.

6. What methods you opt for payment?

We provide both online and offline modes.

7. How do you ensure quality?

We provide better services and maintenance for our vehicles and we provide proper services for our customers.

8. What about fuel?

We only provide 2 litre petrol for free.

9.What are the offers you provide?

We provide a cashback of Rs. 500 for the first ride.

10.Do you face any issues regarding data privacy?

Yes. we cannot sort the things easily.

3.1 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - Intel Core i5

RAM - 4 GB

Hard disk - 1 T B

3.2.2 Software Specification

Front End - HTML, CSS

Back End - Django, SQLite

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, Django, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 DJANGO

Django is an open-source web framework written in the Python programming language. Named after the jazz guitarist Django Reinhardt, it is used by some of the largest websites in the world including Instagram, Mozilla, and NASA, but also lightweight enough to be a popular choice for weekend side projects and startups. Its "batteries-included" approach means a powerful website can be generated quickly in the hands of a skilled developer

Django adopts a "batteries-included" approach similar to Python and comes with a number of built-in features including an extensible authentication system, robust admin app, lightweight testing web server, and support for multiple databases including PostgreSQL, MySQL, MariaDB, Oracle, and SQLite. It is known for its leading security best practices and comes with comprehensive documentation, available either online or as a PDF/ePUB for offline consumption.

As a mature project, Django rarely makes breaking changes and has a clear deprecation schedule for any updates. A major new version is released every nine months or so with monthly patch releases for security and bug fixes. There is also a vibrant ecosystem of third-party applications visible on the Django Packages site--which provide additional functionality. Over time, the most popular packages are often rolled into Django itself.

3.3.2 SQLite

SQLite is a self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine. It is the most used database engine in the world. It is an in-process library and its code is publicly available. It is free for use for any purpose, commercial or private. It is basically an embedded SQL database engine. Ordinary disk files can be easily read and write by SQLite because it does not have any separate server like SQL. The SQLite database file format is cross-platform so that anyone can easily copy a database between 32-bit and 64-bit systems. Due to all these features, it is a popular choice as an Application File Format.

Applications of SQLite

- Due to its small code print and efficient usage of memory, it is the popular choice for the database engine in cellphones, PDAs, MP3 players, set-top boxes, and other electronic gadgets.
- It is used as an alternative for open to writing XML, JSON, CSV or some proprietary format into disk files used by the application.
- As it has no complication for configuration and easily stores file in an ordinary disk file, so it can be used as a database for small to medium sized websites.
- It is faster and accessible through a wide variety of third-party tools, so it has great application in different software platforms.

CHAPTER 4
SYSTEM DESIGN

4.1 INTRODUCTION

System Design is the process of designing the architecture, components, and interfaces for a system so that it meets the end-user requirements. system design ranges from discussing the system requirements to product development. System development creates or alters the system so that the processes, practices, and methodologies are changed to develop the system. Therefore, a systematic approach is needed to manage the system requirements and design methodology. It can be classified as logical design and physical design. The logical design represents the abstract data-flow, while the physical design represents the system's input and output processes. It specializes in developing great artwork by saving time and effort. This helps in creating plans for information systems. It is used to solve internal problems, boost efficiency, and broadcast opportunities. It also is the foundation of any business. It contributes a lot to successfully achieving the required results and makes working easier and simpler.

4.2 UML DIAGRAM

The components of the principles of object-oriented programming are represented by the language known as the Unified Modeling Language (UML), which is utilized in the industry of software engineering. It serves as the standard definition of the entire software architecture or structure. Complex algorithms are solved and interacted with in Object-Oriented Programmingby treating them as objects or entities. Anything can be one of these things. It could either be a bank manager or the bank itself. The thing can be a machine, an animal, a vehicle, etc. The issue is how we connect with and control them, even though they are capable of and ought to execute duties. Interacting with other objects, sending data from one object to another, manipulating other objects, etc., are examples of tasks. There could be hundreds or even thousands of objectsin a single piece of software

UML diagram includes the following diagrams:

- Use case Diagram
- Class diagram
- Object Diagram
- Sequence Diagram
- Activity Diagram
- State chart Diagram
- Deployment Diagram
- Component Diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a visual representation of the interactions between system components. An approach for identifying, outlining, and organizing system requirements is called a use case. The word "system" in this context refers to a project or business that is under development or operation, such a mail-order goods sales and service web page. The Unified Modelling Language (UML) makes use of use case diagrams. a common notation for simulating systems and things in the actual world. Planning for overall requirements is one of the system objectives. Testing and debugging a software product, and verifying a hardware design Performing a consumer service, developing, writing an online help guide, or focused task Use cases in a product sales context, for instance, would include ordering of goods, catalogue revision, transaction processing, and client.

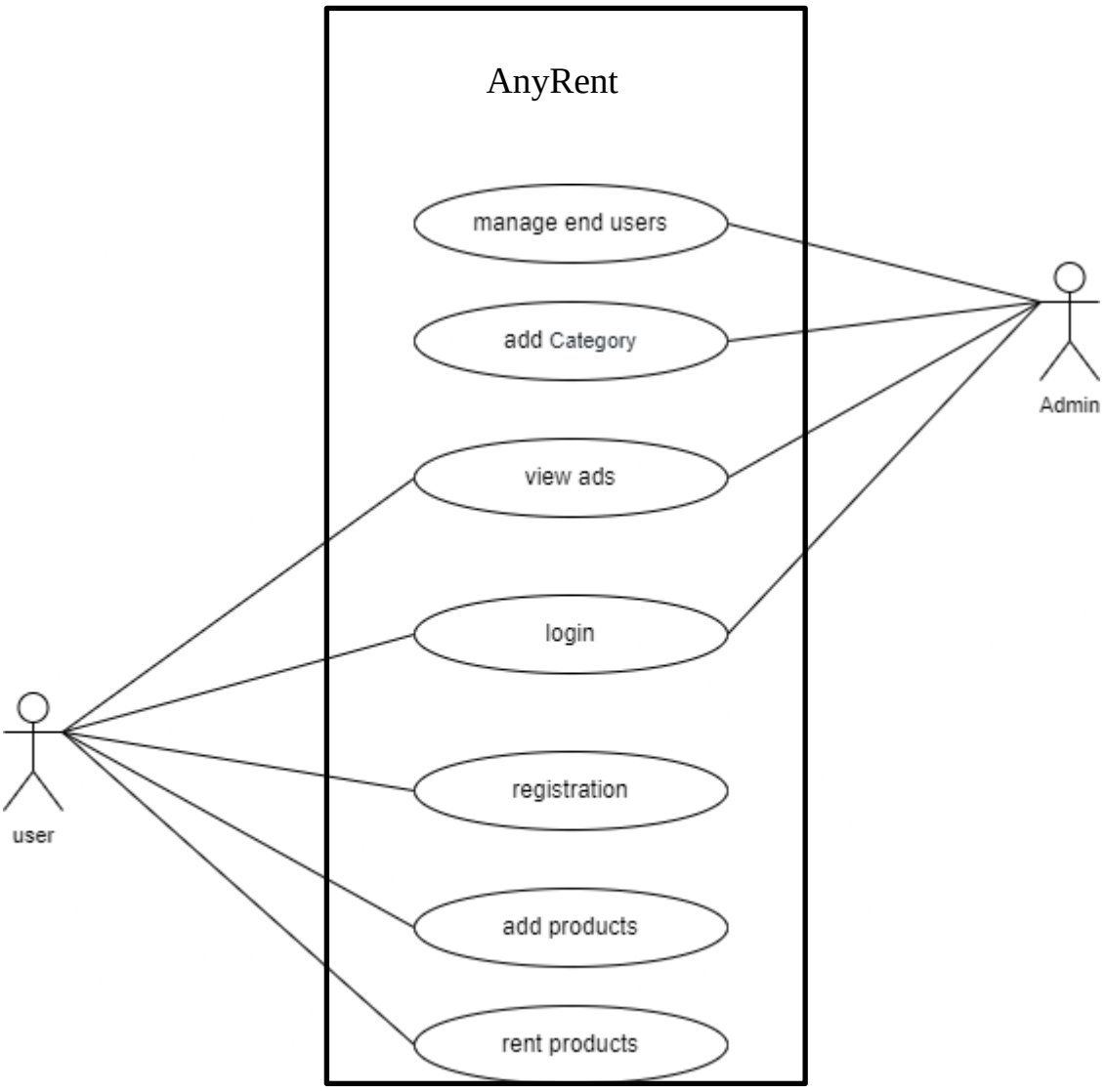


Figure:4.2.1 Use case Diagram

4.2.1 SEQUENCE DIAGRAM

A sequence diagram essentially shows how things interact with one another sequentially, or the order in which these interactions occur. A sequence diagram can also be referred to as event diagrams or event scenarios. Sequence maps define the actions that the system's components takeand in what order. These schematics are Businesspeople and software developers frequently employ documentation and understanding specifications for both current and future systems.

Sequence Diagram Notations –

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.

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

Messages – Communication between objects is depicted using messages The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelinesand messages form the core of a sequence diagram.

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.

User

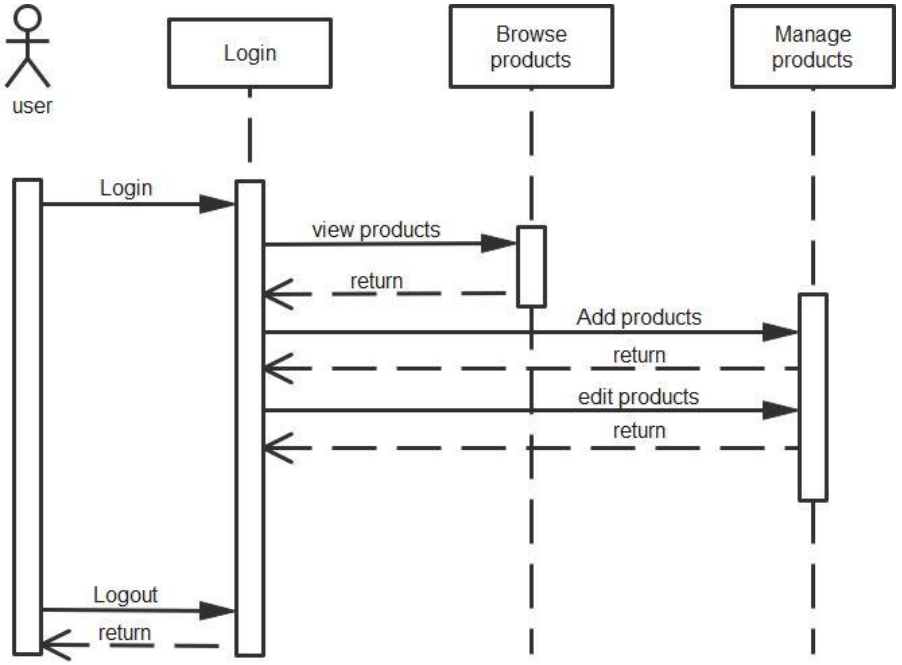


Figure:4.2.2 Sequence Diagram

Admin

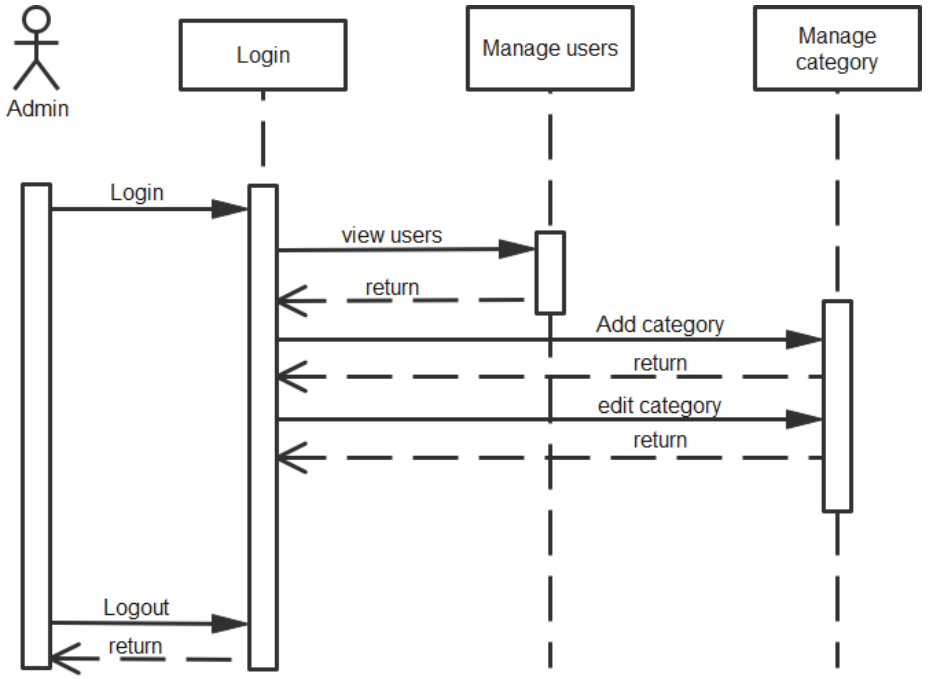


Figure:4.2.2 Sequence Diagram

4.2.2 State Chart Diagram

A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics. The State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model reactive systems. Reactive systems can be defined as a system that responds to external or internal events. State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of the State chart diagram is to model the lifetime of an object from creation to termination. State chart diagrams are also used for the forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

The main purposes of using State chart diagrams are:

- To model the dynamic aspect of a system.
- To model the lifetime of a reactive system.
- To describe different states of an object during its lifetime.
- Define a state machine to model the states of an object.

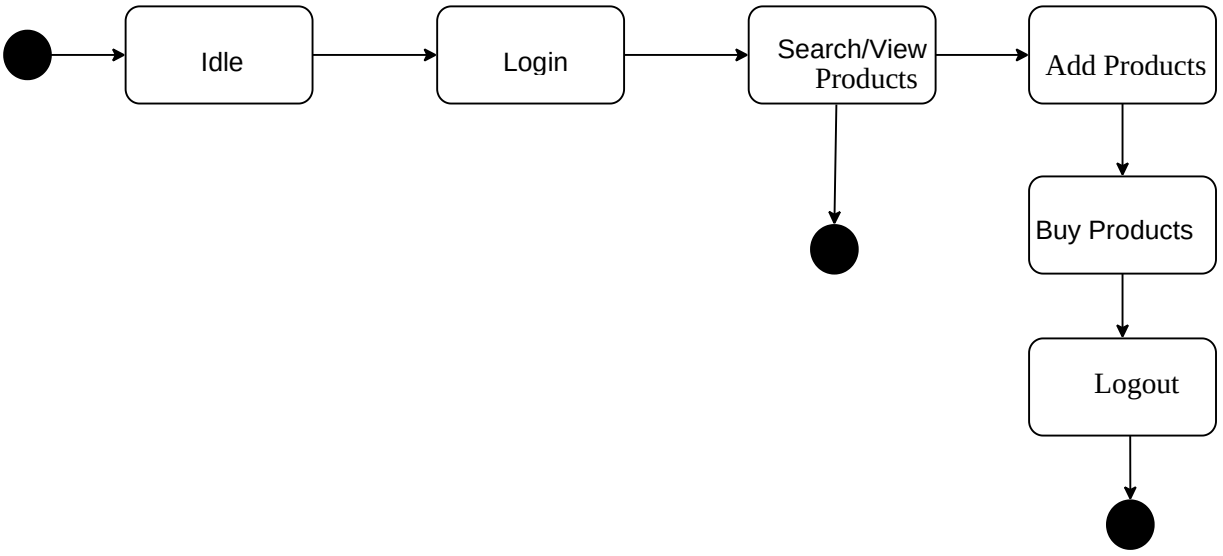


Figure:4.2.3 State Chart Diagram

4.2.2 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores. Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types are:

- Ellipses represent actions
- Diamonds represent decisions.
- Bars represent the start (split) or end (join) of concurrent activities.
- Black circle represents the start (initial node) of the workflow.
- Encircled black circle represents the end (final node).

Arrows run from the start towards the end and represent the order in which activities happen. Activity diagrams can be regarded as a form of a structured flowchart combined with a traditional data flow diagram. Typical flowchart techniques lack constructs for expressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops.

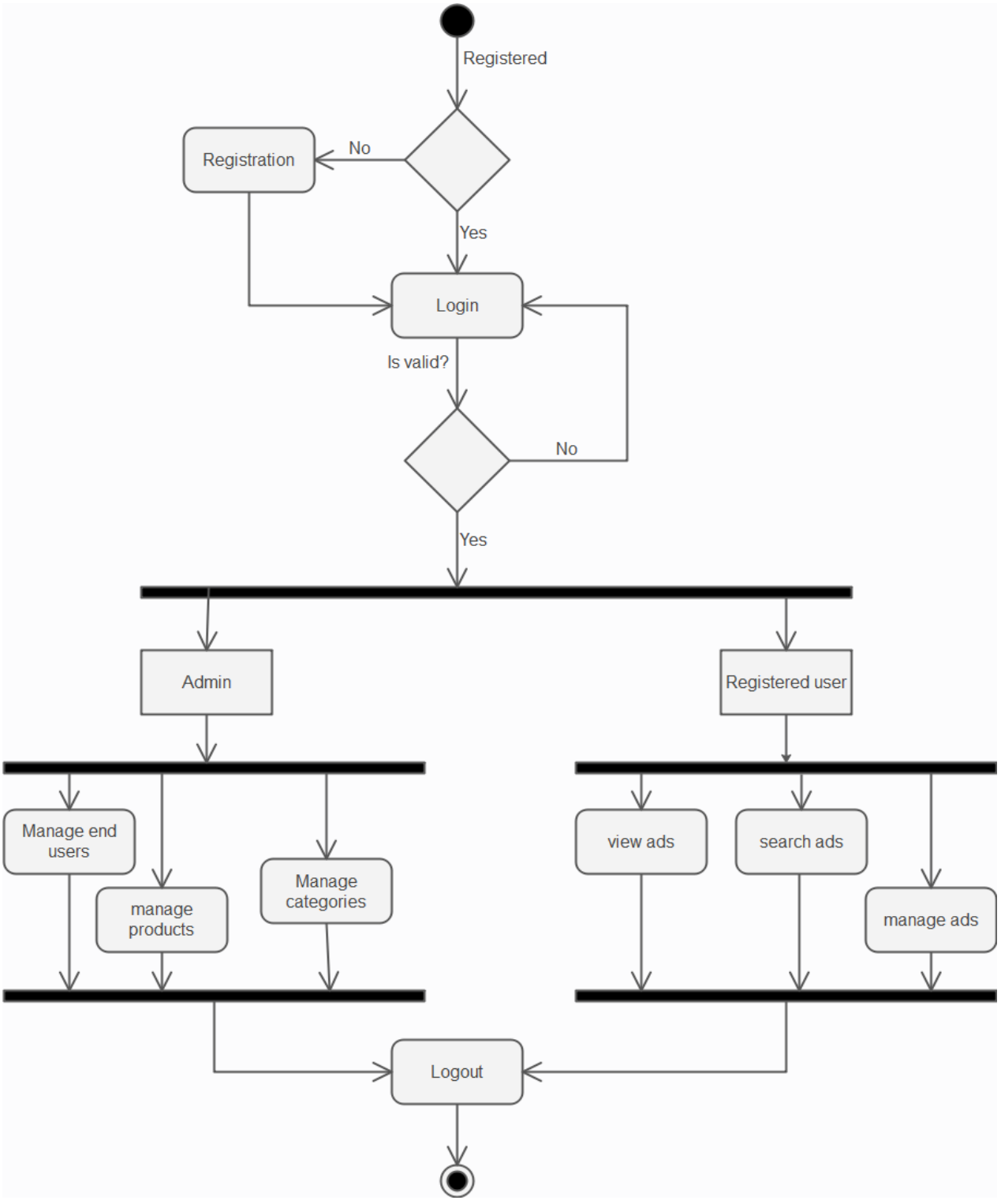


Figure:4.2.4 Activity Diagram

4.2.3 Class Diagram

Class diagrams is a Static diagram. It represents the application's static view. Class diagrams are used to create executable code for software applications as well as for visualizing, explaining, and documenting various elements of systems. The characteristics and functions of a class are described in a class diagram, along with the restrictions placed on the system. Because they are the only UML diagrams that can be directly transferred to object-oriented languages, class diagrams are frequently employed in the modelling of object-oriented systems. A collection of classes, interfaces, affiliations, collaborations, and constraints are displayed in a class diagram.

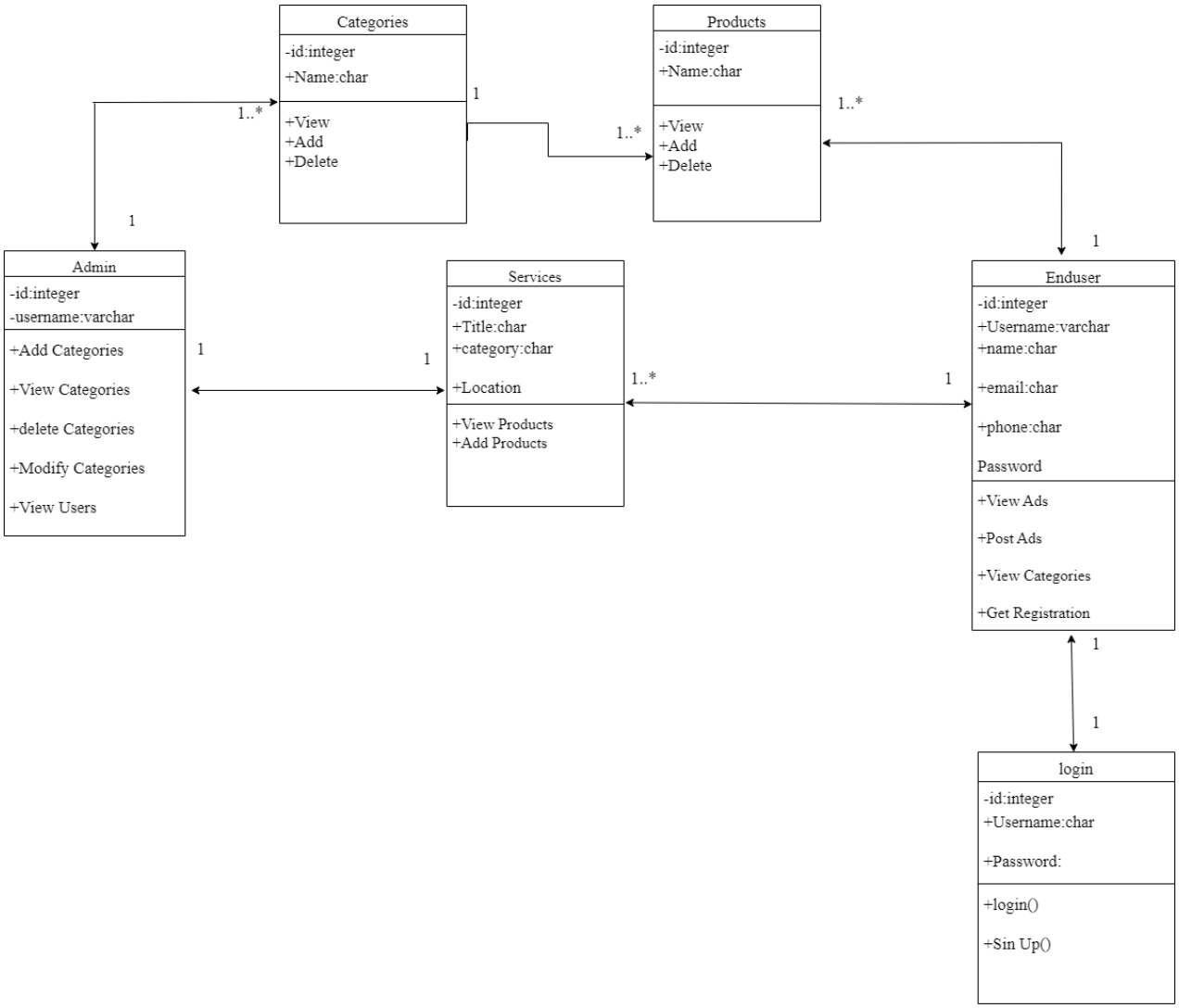


Figure:4.2.5 Class Diagram

4.2.3 Object Diagram

Class diagrams are a requirement for object diagrams because they are the source of class diagrams. An object diagram illustrates a specific instance of a class diagram. The basic concepts of class diagrams and object diagrams are the same. Object diagrams are also used to describe a system's static view, which is a snapshot of the system taken at a particular point in time. You can see a group of things and their relationships by using object diagrams.

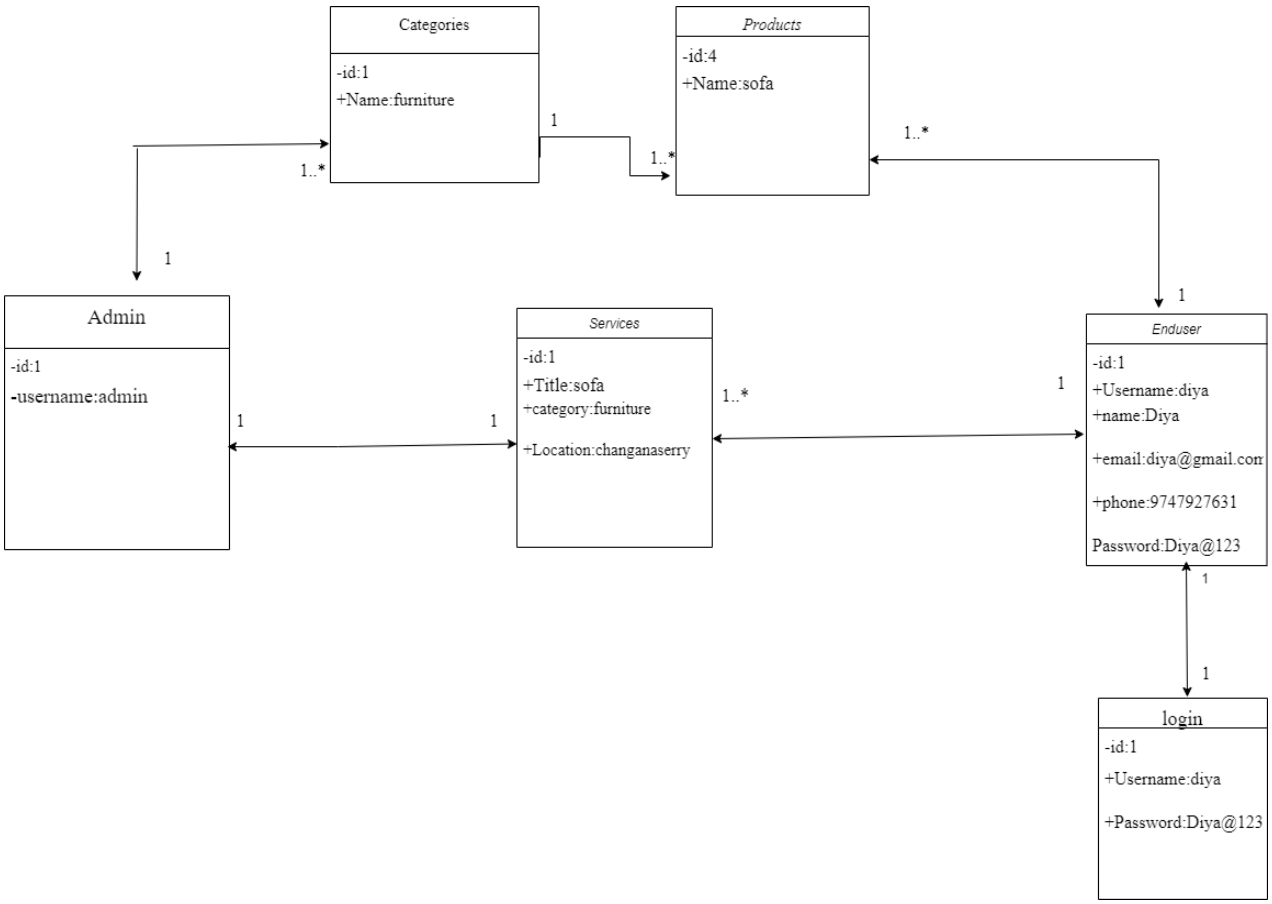


Figure:4.2.5 Object Diagram

4.2.4 Component Diagram

Component diagrams come in a variety of behavior's and personalities. The physical parts of the system are represented using component diagrams. Executables, libraries, files, documents, and other items that are physically present in a node are just a few examples. Component diagrams are used to show how the components of a system are connected and arranged. These diagrams can also be used to construct systems that can be run.

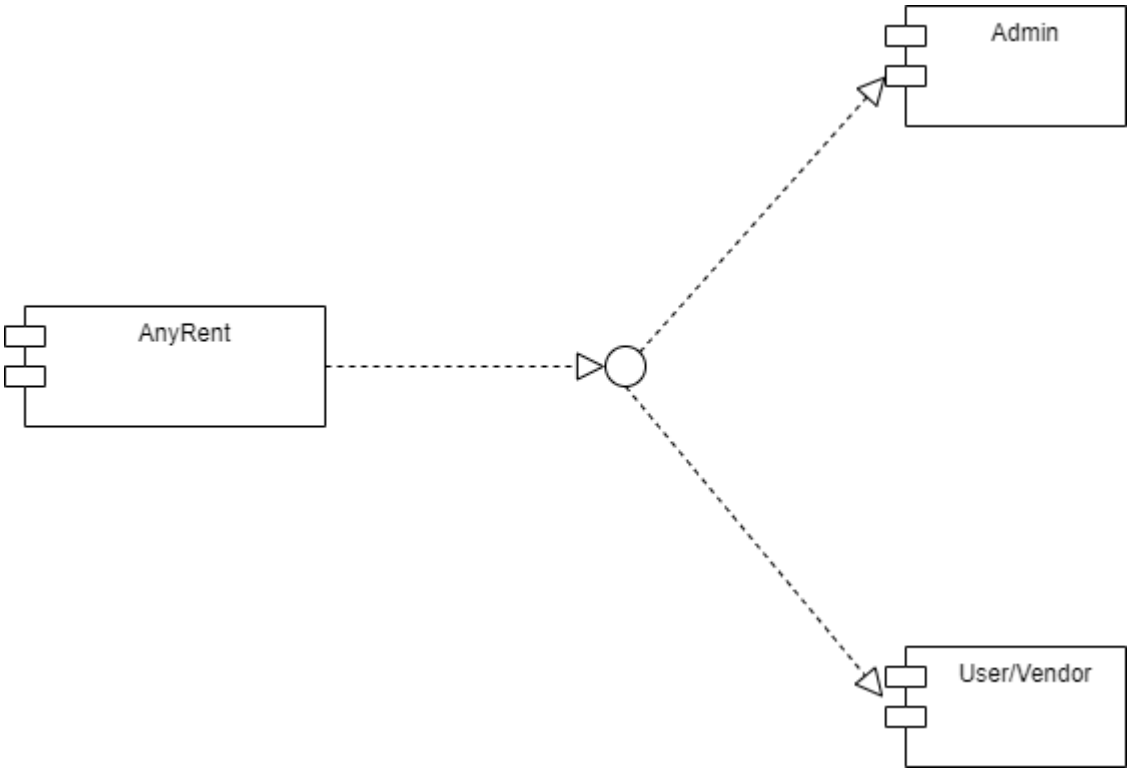


Figure:4.2.7 Component Diagram

4.2.8 Deployment Diagram

Deployment diagrams show the topology of a system's physical components, where the software components are installed. Deployment diagrams are used to describe a system's static deployment view. The key elements of deployment diagrams are nodes and connections between them.

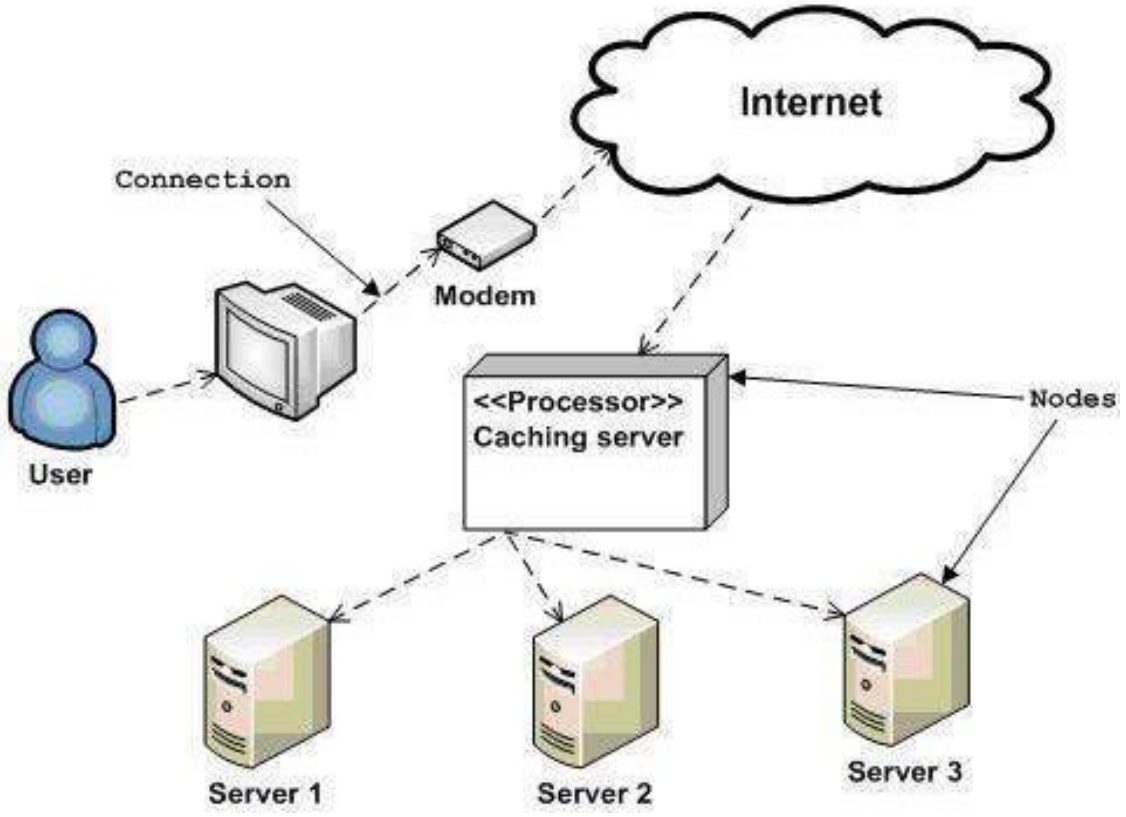


Figure:4.2.8 Deployment Diagram

4.3 USER INTERFACE DESIGN USING FIGMA

4.3.1 Form Name: User Login

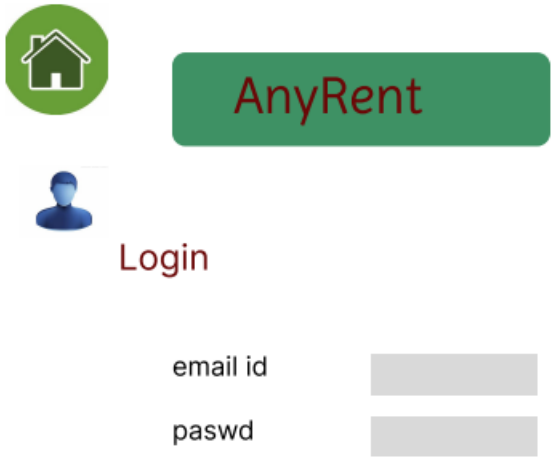


Figure 4.3.1

4.3.2 Form Name: User Registration

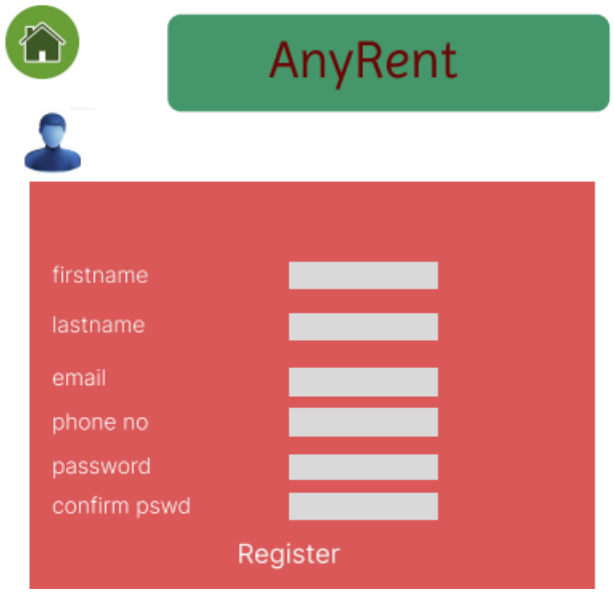


Figure 4.3.2

4.3.3 Form Name: User Home Page

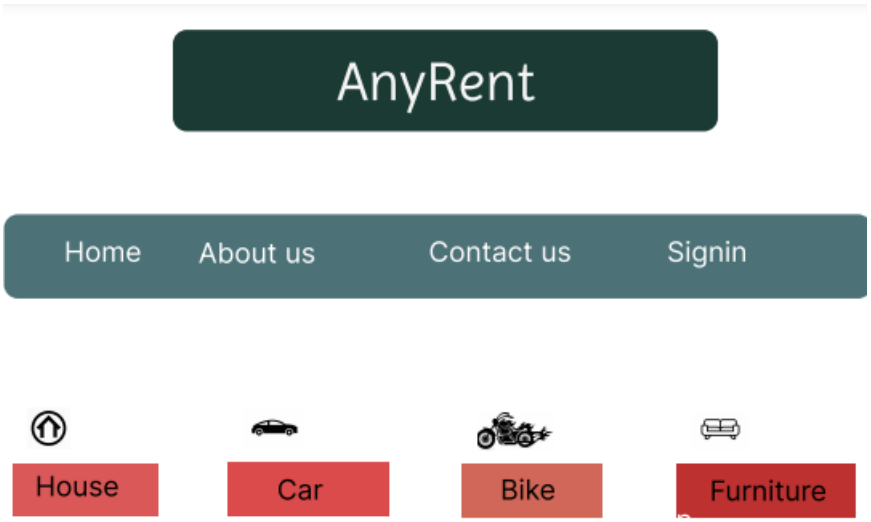


Figure 4.3.3

4.4 DATABASE DESIGN

A database is a structured system with the ability to store information and allow users to access that information quickly and effectively. Any database must be protected because its primary goal is its data. The database design process has two stages. User requirements are obtained in the first step, and a database is created to as clearly as possible meet theseobjectives. Information Level Design is the name of this stage, which is carried out independently of any specific DBMS. The second phase involves converting this information level design into a design for the DBMS that will be used to implement the system in issue. Physical Level Design is the stage where the properties of the particular DBMS are considered. Data Integrity Data independence

4.4.1 Relational Database Management System (RDBMS)

In a relational model, the database is shown as a set of relations. each connection like a file or table of records with values. formal terminology for a relational model, A column header is known as an attribute, a row is known as a tuple, and the table is known as a relation. Each table in a relational database is made up of data that is stored in rows and columns. assigned an arbitrary name. In a story, each row represents a group of associated values.

Domains, Relations, and Attributes

A relation is a table. Tuples are the units of a table's rows. An ordered group of n elements is a tuple. Attributes are referred to as columns. Every table in the database has relationships already established between them. This guarantees the integrity of both referential and entity relationships. A group of atomic values make up a domain D. Specifying the data type from which the domain's data values are derived is a standard way to define a domain. To make it easier to understand the values of the domain, it is also helpful to give it a name. Each value ina relation is atomic and cannot be broken down.

4.4.2 Normalization

The simplest possible grouping of data is used to put them together so that future changes can be made with little influence on the data structures. Data normalization is a formal process. structures in ways that encourage integrity and remove duplication. Normalization is a method of dividing large datasets into smaller ones and removing superfluous fields. Into a smaller table. Additionally, it serves to prevent additions, deletions, and updates. Anomalies. Keys and relationships are two notions commonly used in data modelling. A table row is uniquely identified by its key. key uniquely identifies a row in a table. A primary key is an element, or set of components, in a table that serves as a means of distinguishing between records from the same table. A column in a table known as a foreign key is used to uniquely identify records from other.

tables. Up to the third normal form, all tables have been normalized means placing things in their natural form, as the name suggests. By using normalization, the application developer aims to establish a coherent arrangement of the data into appropriate tables and columns, where names may be quickly related to the data by the user.

First Normal Form

According to the First Normal Form, any attribute's tuple's value must be a single value from its domain, which must only contain atomic values. the territory of that property. To put it another way, 1NF forbids "relations within relations. "alternatively, "relations as attribute values within tuples." The sole attribute values that are allowed by1NF are indivisible or single-atom values. The data must be entered into Initial as the first step. Standard Form.

By putting the data in separate tables, you may donate this.in each table is of a similar type. A primary key or foreign key is assigned to each table as per the project's requirements. For each non-atomic relationship, we create new ones in this. attribute or nested relation. This eliminated 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. connection or nested attribute. This got rid of data groups that were repeated.

Second Normal Form

Accordance with Second Normal Form No non-key attribute should be functionally dependent on a portion of the primary key for relations when the main key has several attributes. This involves breaking down each partial key into its dependent characteristics and setting up a new relation for each one. Keep the original primary key and any properties that are entirelydependent on it in your database. This procedure aids in removing data that depends only on a small portion of the key. If and only if a relation satisfies all the requirements for first normal form for the primary key and every non-primary key attribute of the connection is completely dependent on its primary key alone, then that relation is said to be in second normal form.

Third Normal Form

According to the Third Normal Form relation should not have a non-key attribute that is functionally determined by another non-key attribute or by a collection of non-key attributes. The primary key should not be transitively dependent, in other words. The non-key attributes that functionally determine other non-key attributes are decomposed in this way put up in relation. This action is made to remove anything that is not completely dependent on the Primary Key. Only when a relation is in second normal form and, more importantly, when its non-key characteristics do not depend on those of other non-key attributes, is it considered to be in third

normal form.

4.4.3 Sanitization

An automated procedure called "sanitization" is used to get a value ready for use in a SQL query. This process typically involves checking the value for particular characters that have a special significance for the target database. To prevent a SQL injection attack, you must sanitise (filter) the input string while processing a SQL query based on user input. For instance, the userand password input is a typical scenario. In that particular scenario, the server response would provide access to the 'target user' account without requiring a password check.

4.4.4 Indexing

By reducing the number of disk accesses needed when a query is completed, indexing helps a database perform better. It is a data structure method used to locate and access data in a database rapidly. Several database columns are used to generate indexes. The primary key or candidate key of the table is duplicated in the first column, which is the Search key. To make it easier to find the related data, these values are kept in sorted order Recall that the information may or may not be kept in sorted order.

4.5 TABLE DESIGN

1.Table Name: **Tbl_user_login**

Use: To store the User Login Details

Primary key: **email**

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	email	Varchar (20)	Primary Key	Email of User
2	Password	Varchar (20)		Password of User
3	Status	Int (10)		Status of User

2. Table Name: **Tbl_user_Registration**

Use: To store the User Registration Details

Primary key: **id**

Foreign Key: email references Tbl_user_login

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	id	Int (10)	Primary Key	Id
2	email	Varchar (20)	Foreign Key	Email of User
3	Phone	Varchar (20)		Phone Number of User
4	Fname	Varchar (20)		First Name of User
5	Lname	Varchar (20)		Last Name of User

3. Table Name: **Tbl_Category**

Primary key: **id**

Use: To store the Category Details

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	Cid	Int (10)	Primary Key	Category Id
2	Cname	Varchar (20)		Name of product category
3	Description	Varchar (20)		Description of category

4. Table Name: **Tbl_House_Product**

Primary key: Hid

Use: To store the House Details

Foreign Key: Cid references Tbl_Category

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	Pid	Int (10)	Primary Key	House Id
2	Cid	Int (10)	Foreign Key	Category id, foreign key
3	type	Varchar (20)		Type of house
4	bedroom	Int (25)		Number of bedrooms
5	bathroom	Int (25)		Number of bathrooms
6	Image	Varchar (20)		Image of house
7	amount	Int (50)		Amount of rent
8	ad_title	Varchar (20)		Advertisement title
9	Add_info	Varchar (100)		Additional information
10	Status	Int (10)		Status of house product availability

5. Table Name: **Tbl_Car_Product**

Primary key: Carid

Use: To store the Car Details

Foreign Key: Cid references Tbl_Category

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	Carid	Int (10)	Primary Key	Car Id
2	Cid	Int (10)	Foreign Key	Category id, foreign key
3	brand	Varchar (20)		Brand of car
4	fuel	Varchar (20)		Fuel type of the car
5	own	Int (5)		Number of ownership
6	Image	Varchar (20)		Image of car
7	driven	Int (50)		Km driven
8	amount	Int (50)		Amount of rent
9	ad_title	Varchar (20)		Advertisement title
10	Add_info	Varchar (100)		Additional information
11	Status	Int (10)		Status of car product availability

6. Table Name: **Tbl_Bike_Product**

Primary key: Bid

Use: To store the Bike Details

Foreign Key: Cid references Tbl_Category

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	Bid	Int (10)	Primary Key	Bike Id
2	Cid	Int (10)	Foreign Key	Category id, foreign key
3	brand	Varchar (20)		Brand of bike
4	driven	Int (20)		Km driven
5	own	Int (20)		Number of ownership
6	Image	Varchar (20)		Image of bike
7	amount	Int (50)		Amount of rent
8	ad_title	Varchar (20)		Advertisement title
9	Add_info	Varchar (100)		Additional information
10	Status	Int (10)		Status of bike product availability

7. Table Name: **Tbl_Furniture_Product**

Primary key: Fid

Use: To store the Furniture Details

Foreign Key: Cid references Tbl_Category

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	Fid	Int (10)	Primary Key	Furniture Id
2	Cid	Int (10)	Foreign Key	Category id, foreign key
3	type	Varchar (20)		Type of furniture
7	Image	Varchar (20)		Image of furniture
8	amount	Int (50)		Amount of rent
9	ad_title	Varchar (20)		Advertisement title
10	Add_info	Varchar (100)		Additional information
11	Status	Int (10)		Status of furniture product availability

8. Table Name: **Tbl_Other_Product**

Primary key: Oid

Use: To store the Other product Details

Foreign Key: Cid references Tbl_Category

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	Oid	Int (10)	Primary Key	Other Product Id
2	Cid	Int (10)	Foreign Key	Category id, foreign key
3	type	Varchar (20)		Type of Other Product
7	Image	Varchar (20)		Image of Other Product
8	amount	Int (50)		Amount of rent
9	ad_title	Varchar (20)		Advertisement title
10	Add_info	Varchar (100)		Additional information
11	Status	Int (10)		Status of other product availability

CHAPTER 5
SYSTEM TESTING

5.1 INTRODUCTION

Software testing is the procedure of meticulously monitoring the way software is used to seeif it works as expected. Software testing is usually used in conjunction with the words validation and verification. A product, including software, is validated by being examined or evaluated to see if it conforms to all pertinent requirements. Software testing, one sort ofverification, also makes use of reviews, analyses, inspections, and walkthroughs. Validationis the process of ensuring that what has been specified corresponds to what the user actuallywants.

Other procedures that are typically connected to software testing include static analysis and dynamic analysis. Without actually running the code, static analysis examines the software's source code to look for errors and gather statistics. Dynamic analysis looks at the behaviour of software while it is in use to provide information like execution traces, timing profiles, and test coverage specifics.

Running a programme with the intention of finding any faults is how a programme is tested. Testing is a group of tasks that can be organized in advance and completed in a systematic way. Individual modules are tested first, followed by the integration of the entire computer-based system. There are numerous regulations that can be utilized as testing objectives, andtesting is essential for the achievement of system testing objectives. The following:

- A test case with a high likelihood of detecting an unknown fault qualifies as a goodtest case.
- A test that finds an error that has not yet been found is successful.

A test that effectively accomplishes the aforementioned goals will reveal software problems. Testing also reveals that the software functions appear to function in accordance with the specification and that the performance requirements appear to have been met. There are three ways to test programme.

- For accuracy
- For effective implementation
- Regarding computational complexity

Testing for correctness is meant to ensure that a program performs exactly as it was intendedto. This is much harder than it might initially seem, especially for big programs.

5.2 TEST PLAN

A test plan suggests a number of required steps that need be taken in order to complete various testing methodologies. The activity that is to be taken is outlined in the test plan. A computer program, its documentation, and associated data structures are all created by software developers. It is always the responsibility of the software developers to test each of the program's separate components to make sure it fulfil the purpose for which it was intended. In order to solve the inherent issues with allowing the builder to evaluate what they have developed, there is an independent test group (ITG). Testings precise goals should be laid forth in quantifiable language. so that the cost to discover and remedy the problem, 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 concentrates verification efforts on the software component or module, which is the smallest unit of software design. The component level design description is used as aguide when testing crucial control paths to find faults inside the module's perimeter. the level of test complexity and the untested area determined for unit testing. Unit testing is white- box focused, and numerous components may be tested simultaneously. To guarantee that data enters and exits the software unit under test properly, the modular interface is tested. To make sure that data temporarily stored retains its integrity during each step of an algorithm's execution, the local data structure is inspected. To confirm that each statement in a module has been executed at least once, boundary conditions are evaluated. Finally, each path for managing errors is examined.

Testing of data flow through a module interface are important before beginning anyadditional tests. If data cannot correctly enter and exit the system, all other tests are useless.The unit test's selective analysis of execution routes is a crucial task. In order to cleanly reroute or stop work when an error does occur, error handling channels must be set up and error scenarios must be anticipated in excellent design. Boundary testing is the last stage ofunit testing.

In the Sell-Soft System, unit testing was carried out by treating each module as a distinct entity and subjecting them to a variety of test inputs. The internal logic of the modules had some issues, which were fixed. Each module is tested and run separately after coding. All unused code was eliminated, and it was confirmed that every module was functional and produced the desired outcome.

5.2.2 Integration Testing

Integration testing is a methodical approach for creating the program's structure while also carrying out tests to find interface issues. The goal is to construct a program structure that has been determined by design using unit tested components. The program as a whole is tested. Correction is challenging since the size of the overall program makes it challenging to isolate the causes. As soon as these mistakes are fixed, new ones arise, and the process repeats itself in an apparently unending cycle. All of the modules were integrated after unit testing was completed in the system to check for any interface inconsistencies. A distinctive program structure also developed when discrepancies in program structures were eliminated.

5.2.3 Validation Testing or System Testing

This marks the conclusion of the testing procedure. This required comprehensive testing ofthe system, which covered all forms, codes, modules, and class modules. System tests and black box testing are two common names for this kind of testing. The functional requirements of the software are the main emphasis of the black box testingapproach. That example, using Black Box testing, a software engineer can create sets of input conditions that will fully test every program requirement. erroneous or missing functions, interface faults, data structure or external data access errors,performance errors, initialization errors, and termination errors are the kind of issues that black box testing focuses on.

5.2.4 Output Testing or User Acceptance Testing

The system under consideration is tested for user acceptance; in this case, it must satisfy the business' requirements. The programmer should consult the user and the perspective systemas it is being developed in order to make any necessary adjustments. This is carried out in relation to the following things:

- Screen Designs for Input
- Screen Designs for Output

The aforementioned testing is carried out using a variety of test data. The preparation of testdata

is essential to the system testing process. The system under investigation is then put tothe test using the prepared test data. Errors in the system are once again found during testing,fixed using the methods described above, and logged for use in the future.

5.2.5 Automation Testing

Software and other computer goods are tested automatically to make sure they abide by tight guidelines. In essence, it's a test to ensure that the hardware or software performs exactly as intended. It checks for errors, flaws, and any other problems that might occur throughout the creation of the product. Any time of day can be used to do automation testing. It looks at the software using scripted sequences. It then summarizes what was discovered, and this data can be compared to results from earlier test runs.

Benefits of Automation Testing

Detailed reporting capabilities - Automation testing uses well-crafted test cases for various scenarios. These scripted sequences can be incredibly in-depth, and provide detailed reportsthat simply wouldn’t be possible when done by a human.

Improved bug detection - One of the main reasons to test a product is to detect bugs and other defects. Automation testing makes this process an easier one. It’s also able to analyzes wider test coverage than humans may be able to.

Simplifies testing - Most SaaS and IT organizations routinely include testing in their daily operations. The key is to keep things as basic as you can. Automation has a lotof advantages. The test scripts can be reused when automating test tools.

Quickens the testing procedure - Machines and automated technology operate morequickly than people. This is why we employ them, along with increased accuracy. Your software development cycles are subsequently shortened by this.

Lessens the requirement for human supervision - Tests may be conducted at any timeof day, including overnight. Additionally, when done automatically, this can lessen the possibility of human error.

5.2.6 Selenium Testing

An open-source programme called Selenium automates web browsers. It offers a single interface that enables you to create test scripts in a number of different programming languages, including Ruby, Java, NodeJS, PHP, Perl, Python, and C #. Web application testing for cross-browser compatibility is automated using the Selenium testing tool. Whether they are responsive, progressive, or standard, it is utilized to assure high-quality web apps. Selenium is a free software programme.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

The project's implementation phase is where the conceptual design is transformed into a functional system. It can be regarded as the most important stage in creating a successful new system since it gives users assurance that the system will operate as intended and be reliable and accurate. User documentation and training are its main concerns. Usually, conversion happens either during or after the user's training. Implementation is the process of turning a newly revised system design into an operational one, and it simply refers to placing a new system design into operation.

The user department now bears the most of the workload, faces the most disruption, and hasthe biggest influence on the current system. If the implementation is not well thought out or managed, confusion and mayhem may result.

Implementation encompasses all of the steps used to switch from the old system to the newone. The new system could be entirely different, take the place of an existing manual or automated system, or it could be modified to work better. A reliable system that satisfies organizational needs must be implemented properly. System implementation refers to the process of actually using the built system. This comprises all the processes involved in switching from the old to the new system. Only after extensive testing and if it is determinedthat the system is operating in accordance with the standards can it be put into use. The system personnel assess the system's viability. The effort necessary for system analysis and design to implement the three key components of education and training, system testing, andchangeover will increase in proportion to how complicated the system being implemented is.

The following tasks are included in the implementation state:
Meticulous planning.

- A system and constraint investigation.
- Development of transitional approaches

6.2 IMPLEMENTATION PROCEDURES

Software implementation refers to the complete installation of the package in its intended environment, as well as to the system's functionality and satisfaction of its intendedapplications. The software development project is frequently commissioned by someone who will not be using it. People have early reservations about the software, but we must watch out that they do not become more resistant by making sure that:

- The new system's advantages must be known to the active user.
- Their faith in the software is increased.
- The user receives the appropriate instruction so that he feels confident using the application.

Before examining the system, the user must be aware that the server software needs to be running on the server in order to access the results. The actual process won't happen if the server object is not active and functioning on the server.

6.2.1 User Training

The purpose of user training is to get the user ready to test and modify the system. to fulfil the purpose and get the benefits anticipated from a computer-based system, the participantsmust have faith in their ability to fulfil their duties under the new system. Training is more necessary as systems get more complicated. The user learns how to enter data, handle errorwarnings, query the database, call up routines that will generate reports, and execute other important tasks through user training.

6.2.2 Training on the Application Software

The user will need to receive the essential basic training on computer awareness after whichthe new application software will need to be taught to them. This will explain the fundamental principles of how to use the new system, including how the screens work, whatkind of help is displayed on them, what kinds of errors are made while entering data, how each entry is validated, and how to change the date that was entered. Then, while imparting the program's training on the application, it should cover the information required by the particular user or group to operate the system or a certain component of the system. Depending on the user group and hierarchical level, this training could be different.

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 maintenanceis 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. Softwaremaintenance is of course, far more than "Finding Mistakes".

CHAPTER 7
CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The project entitled '**AnyRent** ' have been developed to help people to rent their products through online. We mainly focus on people who move to another place for a short period of time and if they need products for a period, we provide them essential needs as rent through this system. The users can also keep their products for advertisement through this platform which could make an income for them. Using HTML as the front end and Django as back end, along with SQLite as the database, the system was developed and tested with all possible samples of data.

AnyRent provides a fast and efficient web application that can be used to make advertisement for their products like houses and apartments, cars, bikes, furniture etc. People who would like to move to new places can easily find required services vendors through this platform. They can find the products by searching the location and they can contact with the vendors for further details and dealings. The performance of the system is provided to be efficient and can meet all the requirements of the user.

7.2 FUTURE SCOPE

- Placing product images in the google map according to the location where the product is available using Maps Static API provided by google.
- Live Chat Application for the user and vendor for better communication.
- The web application can be migrated to Android so that the users can easily access.
- Advertisement post reach analysis using machine learning.
- Search Engine Optimization(SEO) to optimize the search strategy of the products.

CHAPTER 8
BIBLIOGRAPHY

REFERENCES:

- Python: The complete reference python – 20 March 2018 by Martin C.Brown
- Let us Python by Kanetkar Yashavant
- Reger S Pressman, “Software Engineering”, 2006
- Pankaj Jalote “Software Engineering: a precise approach”, 2006

WEBSITES:

- https://www.djangoproject.com/
- https://www.w3schools.com/Css/css_intro.asp
- https://www.geeksforgeeks.org/how-to-move-an-image-with-the-mouse-in-pygame/
- https://stackoverflow.com/questions/730207/django-model-set-foreign-key-to-a-fieldofanothermodel#:~:text=You%20need%20to%20use%20%22to_field%22%//20in%20%22models.ForeignKey%20%28%29%22,a%20different%20field%2C%20that%20field%20must%20have%20unique%3DTrue.

CHAPTER 9

APPENDIX

9.1 Sample Code

9.1.1 Accounts.py

```
from django.contrib import messages, auth
from django.http import JsonResponse,HttpResponse
from django.shortcuts import render, redirect
from .models import Account
from django.contrib.auth import authenticate


#email verification import files
from django.contrib.sites.shortcuts import get_current_site
from django.template.loader import render_to_string
from django.utils.http import urlsafe_base64_encode, urlsafe_base64_decode
from django.utils.encoding import force_bytes
from django.contrib.auth.tokens import default_token_generator
from django.core.mail import EmailMessage
from django.core.mail import send_mail

def login(request):
    if request.method == 'POST':
        email=request.POST['email']
        password=request.POST['password']
        user=authenticate(email=email, password=password)
        if user and user.is_active:
            auth.login(request, user)
            messages.success(request, 'you are logged in')
            request.session['email']=email
            if user.is_admin:
                return redirect('admin/')
            else:
                return redirect('home')
        else:
            messages.error(request, 'invalid login credentials')
            return redirect('login')
    return render(request, 'login.html')

def register(request):
    if request.method == 'POST':
        email=request.POST['email']
        password=request.POST['password']
        fname=request.POST['fname']
        lname=request.POST['lname']
        phone_number=request.POST['tel']
        if Account.objects.filter(email=email).exists():
            messages.success(request, 'email already exists')
            return redirect('register')
        else:
            user=Account.objects.create_user(email=email, password=password,
fname=fname, lname=lname,  phone_number=phone_number)
            user.save()
            messages.success(request, 'Thank you for registering with us.')
            messages.success(request, 'Please verify your email for login!')

            #code for email verification also check validate function
            current_site = get_current_site(request)
            message = render_to_string('account_verification_email.html', {
                'user': user,
                'domain': current_site,
                'uid': urlsafe_base64_encode(force_bytes(user.pk)),
```

```
                'token': default_token_generator.make_token(user),
            })

            send_mail(
                'Please activate your account',
                message,
                'anyrentplatfrom@gmail.com',
                [email],
                fail_silently=False,
            )

            return redirect('/login/?command=verification&email=' + email)
        # return redirect('/login')
    return render(request, 'register.html')


def logout(request):
    auth.logout(request)
    return redirect('/')

def activate(request, uidb64, token):
    try:
        uid = urlsafe_base64_decode(uidb64).decode()
        user = Account._default_manager.get(pk=uid)
    except(TypeError, ValueError, OverflowError, Account.DoesNotExist):
        user = None

    if user is not None and default_token_generator.check_token(user, token):
        user.is_active = True
        user.save()
        messages.success(request, 'Congratulations! Your account is activated.')
        return redirect('login')
    else:
        messages.error(request, 'Invalid activation link')
        return redirect('register')

def forgotPassword(request):
    if request.method == 'POST':
        email = request.POST['email']
        if Account.objects.filter(email=email).exists():
            user = Account.objects.get(email__exact=email)

            # Reset password email

            current_site = get_current_site(request)
            message = render_to_string('accounts/reset_password_email.html', {
                'user': user,
                'domain': current_site,
                'uid': urlsafe_base64_encode(force_bytes(user.pk)),
                'token': default_token_generator.make_token(user),
            })

            send_mail(
                'Please activate your account',
                message,
                'sankartstore@gmail.com',
                [email],
                fail_silently=False,
            )

            messages.success(request, 'Password reset email has been sent to
your email address.')
```

```
            return redirect('login')
        else:
            messages.error(request, 'Account does not exist!')
            return redirect('forgotPassword')
    return render(request, 'accounts/lost-password.html')
```

9.1.2products.py

```
from django.contrib import messages
from django.contrib.auth.decorators import login_required
from django.core.exceptions import ObjectDoesNotExist
from django.core.paginator import Paginator
from django.shortcuts import render, redirect

from .models import House_Product
from .models import Car_Product

def Category(request):

    return render(request, 'category.html')

@login_required(login_url='login')
def add_house(request):
    current_user = request.user
    if request.method=="POST":
        type=request.POST.get('type')
        furnish=request.POST.get('furnish')
        bedroom=request.POST.get('bedroom')
        bathroom=request.POST.get('bathroom')

        builtup = request.POST.get('builtup')
        capacity = request.POST.get('capacity')
        rent = request.POST.get('rent')
        ad_title = request.POST.get('ad_title')
        add_info = request.POST.get('add_info')
        images=request.FILES['images']
        state = request.POST.get('state')
        city = request.POST.get('city')
        location = request.POST.get('location')

house=House_Product(user=current_user,type=type,furnish=furnish,bedroom=bedroom,
bathroom=bathroom,builtup=builtup,capacity=capacity,

rent=rent,state=state,city=city,location=location,ad_title=ad_title,add_info=add
_info,images=images,)
        house.save()
        messages.success(request, 'Your product is kept for rent!')

        return redirect('shop')
    return render(request, 'add_house.html')



@login_required(login_url='login')
def add_car(request):
    current_user = request.user
    if request.method=="POST":
        brand=request.POST.get('brand')
        fuel=request.POST.get('fuel')
```

```
        driven=request.POST.get('driven')
        own=request.POST.get('own')
        rent = request.POST.get('rent')
        ad_title = request.POST.get('ad_title')
        add_info = request.POST.get('add_info')
        images=request.FILES['images']
        state = request.POST.get('state')
        city = request.POST.get('city')
        location = request.POST.get('location')

car=Car_Product(user=current_user,brand=brand,fuel=fuel,driven=driven,own=own,

rent=rent,ad_title=ad_title,add_info=add_info,images=images,state=state,city=cit
y,location=location)
        car.save()
        messages.success(request, 'Your product is kept for rent!')

        return redirect('shop')
    return render(request, 'add_car.html')


from django.shortcuts import render, redirect

# Create your views here.
from .models import  Bike_Product

@login_required(login_url='login')
def add_bike(request):
    current_user = request.user
    if request.method=="POST":
        brand=request.POST.get('brand')
        driven=request.POST.get('driven')
        own=request.POST.get('own')
        rent = request.POST.get('rent')
        ad_title = request.POST.get('ad_title')
        add_info = request.POST.get('add_info')
        images=request.FILES['images']
        state = request.POST.get('state')
        city = request.POST.get('city')
        location = request.POST.get('location')
        bike=Bike_Product(user=current_user,brand=brand,driven=driven,own=own,

rent=rent,ad_title=ad_title,add_info=add_info,images=images,state=state,city=cit
y,location=location)
        bike.save()
        messages.success(request, 'Your product is kept for rent!')

        return redirect('shop')
    return render(request, 'add_bike.html')

from .models import Furn_Product


@login_required(login_url='login')
def add_furn(request):
    current_user = request.user
    if request.method=="POST":
        type = request.POST.get('type')

        rent = request.POST.get('rent')
        ad_title = request.POST.get('ad_title')
```

9.2 Screen Shots

9.2.1 Login.html

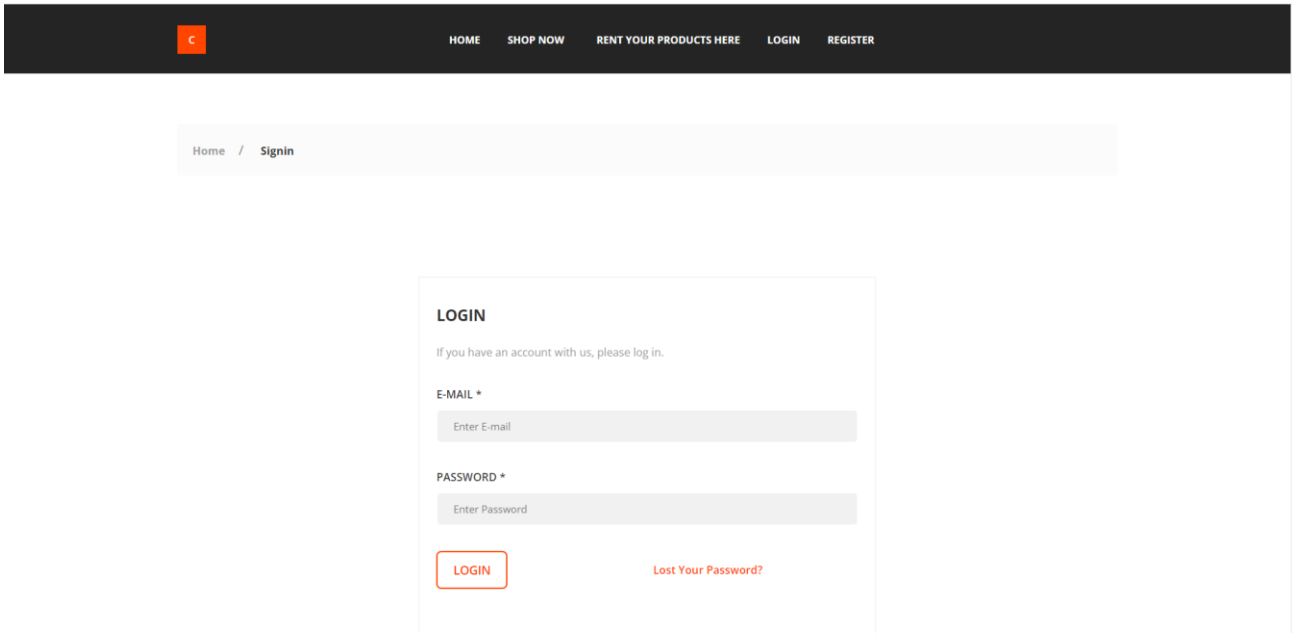


Figure 9.2.1

9.2.2 Registration.html

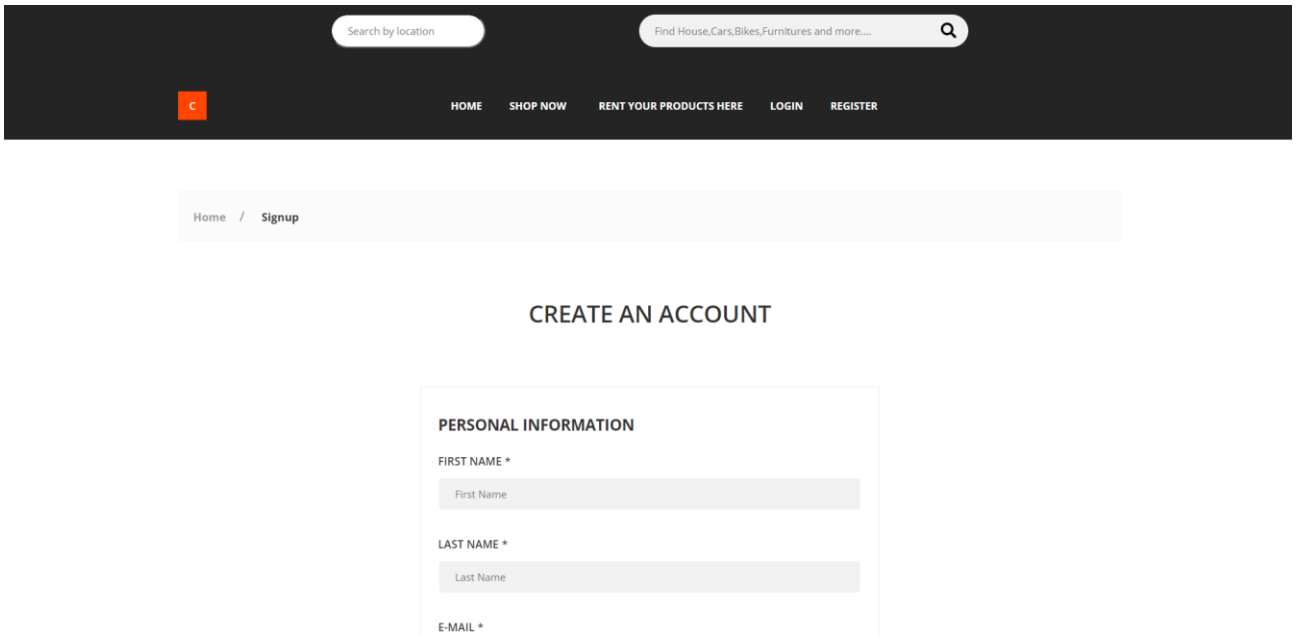


Figure 9.2.2

9.2.3 Home.html

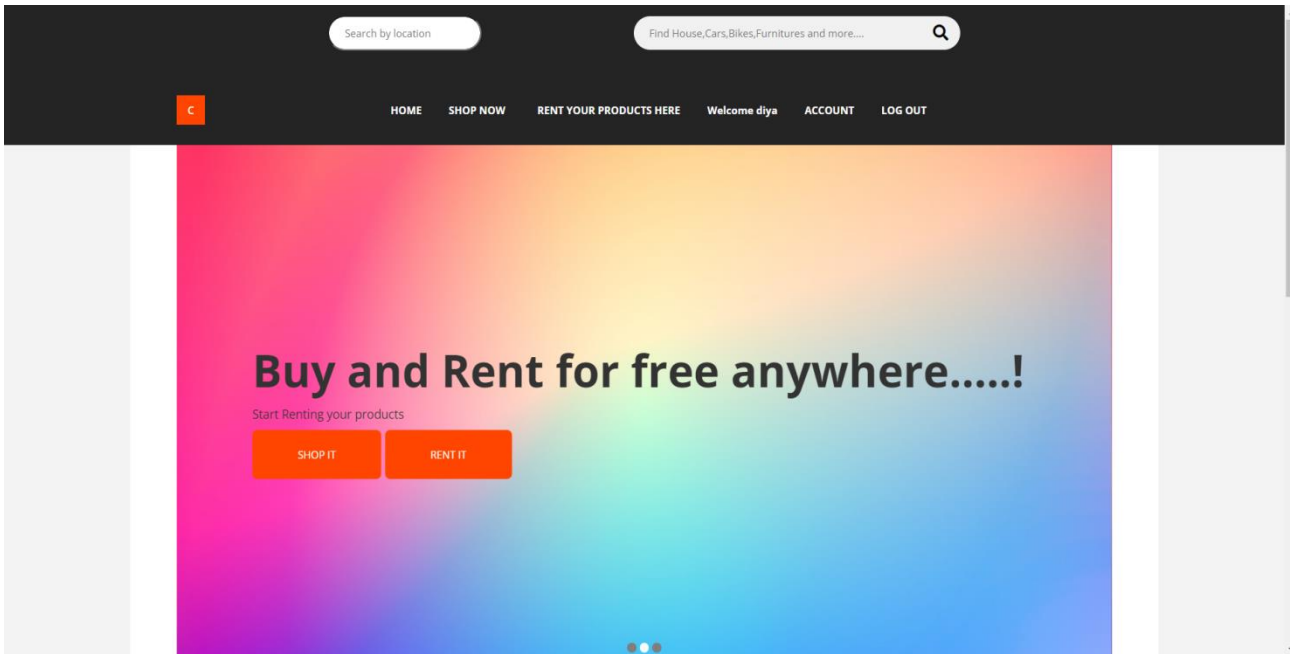


Figure 9.2.3

9.2.4 Profile.html

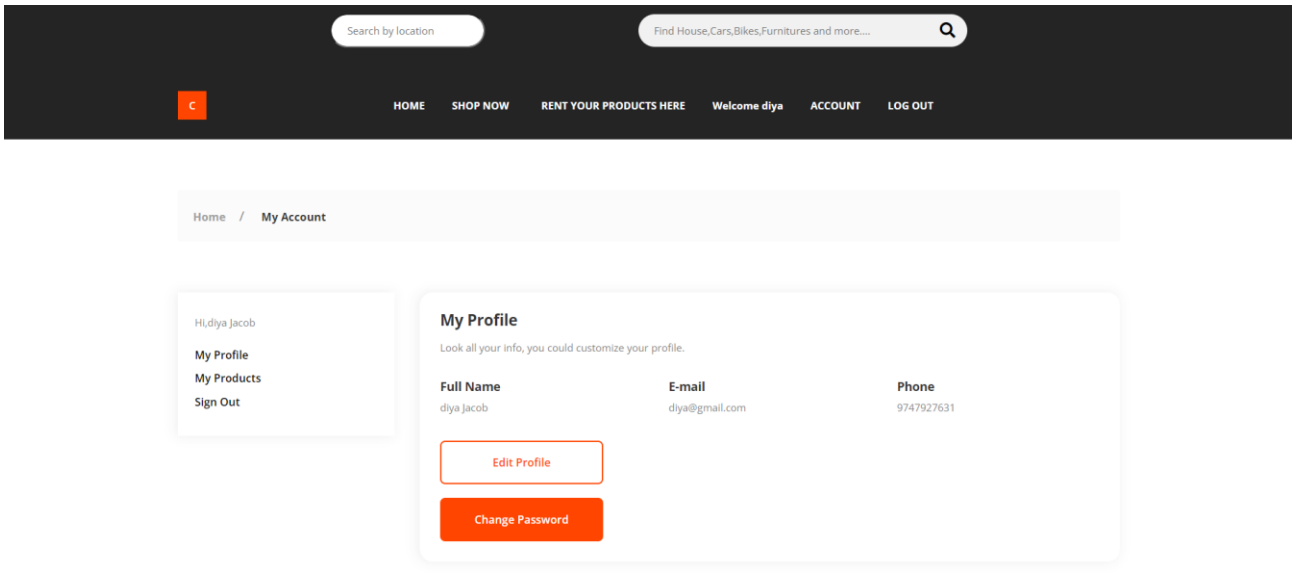


Figure 9.2.4

9.2.5 Categories.html

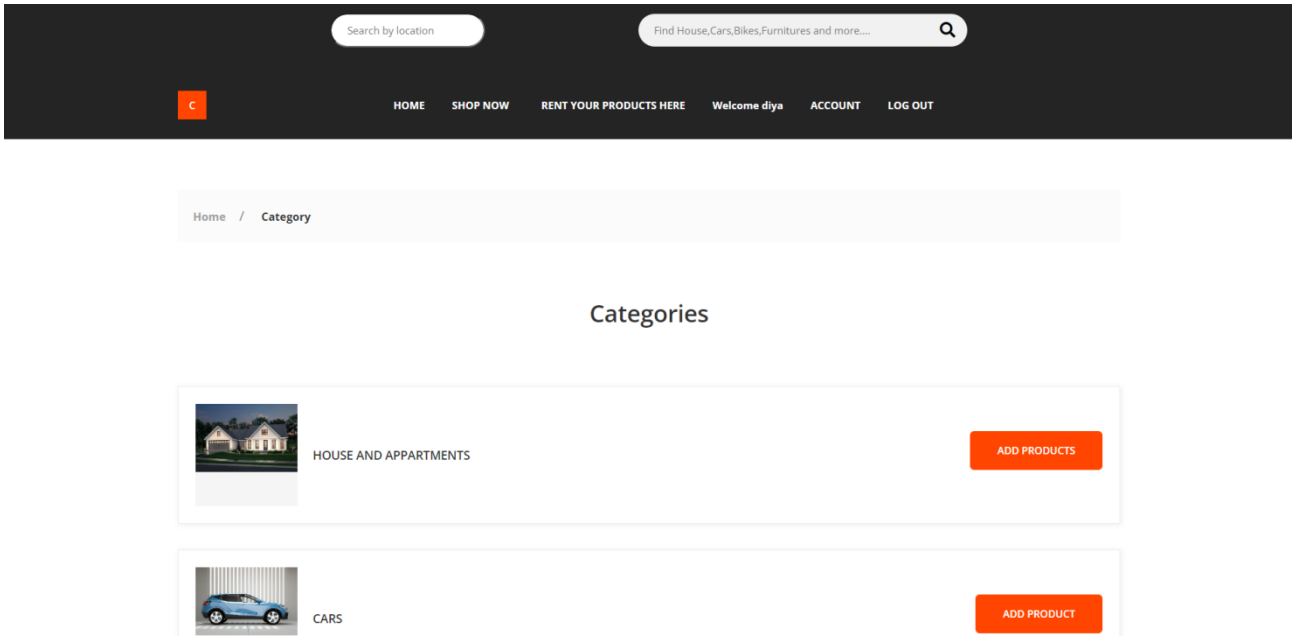


Figure 9.2.5

9.2.6 shop.html

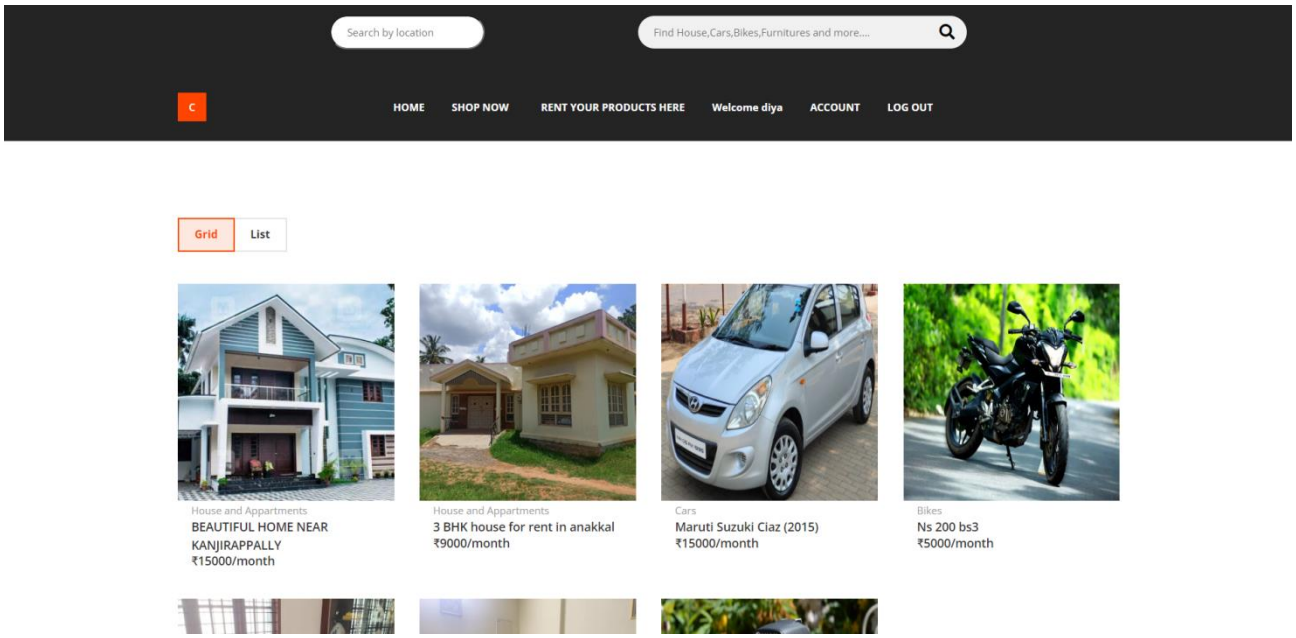


Figure 9.2.6

9.2.7 Myproducts.html

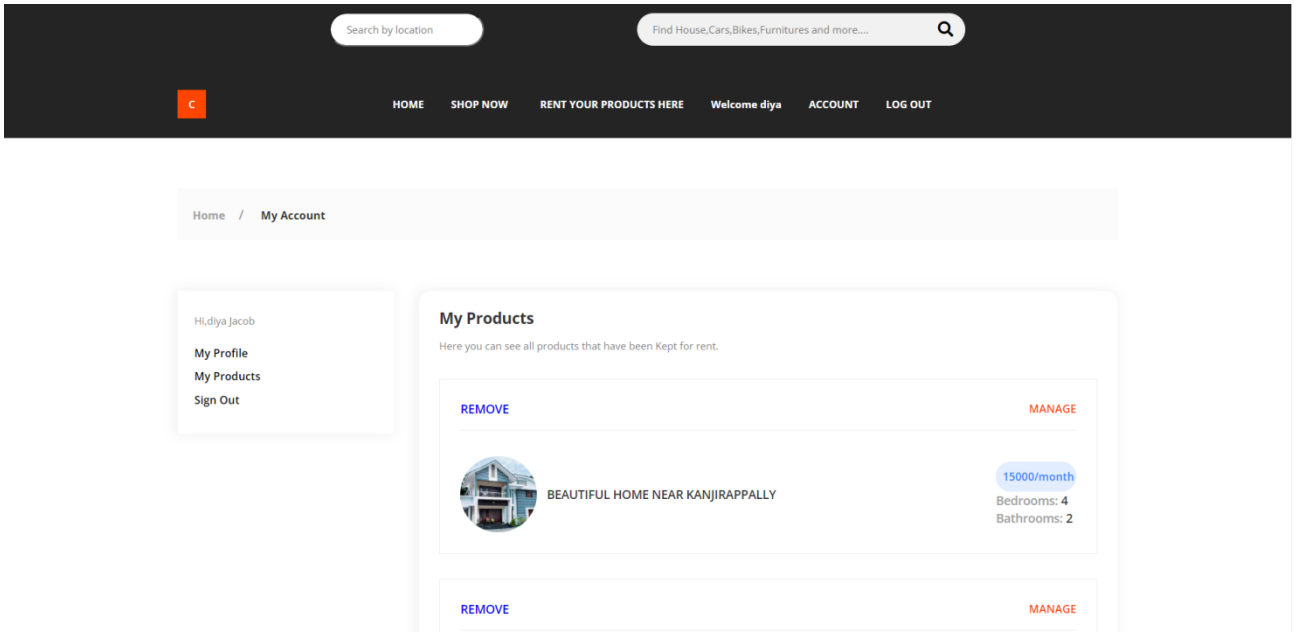


Figure 9.2.7