

A

**PROJECT REPORT  
ON**

**LEASEWITHEASE RENTAL SERVICES**

**(White Goods Rental Services)**

Submitted in partial fulfillment for the award of

**Post Graduate Diploma in Advance Computing**

**(PG-DAC) from**

**INSTITUTE OF EMERGING TECHNOLOGIES**

**Authorized Training Centre**



**Under the Guidance of**

**Name of the Guide Mrs. Prachi Godbole**

**BY**

- 1. Sumit Patulkar -220945920102**
- 2. Reva Sakpal-220945920072**
- 3. Rahul Wagh-220945920070**
- 4. Rohini Kuwar-220945920073**
- 5. Pratiksha Chavhan-220945920024**



## **CERTIFICATE**

This is to certify that the project report entitle **LEASEWITHEASE RENTAL SERVICES (White Goods Rental Services)** is a bonfire work carried out by **SUMIT PATKULKAR , REVA SAKPAL ,RAHUL WAGH ,ROHINI KUWAR , PRATIKSHA CHAVHAN** and submitted in partial fulfilment of the requirement for the C-DAC ACTS, DAC course in Institute of Emerging Technology in the batch of Sep 2022.

**Course Coordinator**

**External Examiner**

## **ACKNOWLEDGEMENT**

This project **LEASE WITH EASE RENTAL SERVICES (White Goods Rental Services)** was a great learning experience for us and we are submitting this work to Advanced Computing Training School (CDAC).

We are very glad to mention the **Mrs. Prachi Godbole** for her valuable guidance to work on this project. Her guidance and support helped us to overcome various obstacles and intricacies during the course of project work.

Our most heart full thanks goes to **Mr. Sangram Patil (Director ,IET)** who gave all the required support and kind coordination to provide all the necessities like required hardware , internet facility and extra lab hours to complete the project and throughout the course up to the last day here in C-DAC ACTS, Pune.

### **Sign of student**

- 1. SUMIT PATKULKAR**
- 2. REVA SAKPAL**
- 3. RAHUL WAGH**
- 4. ROHINI KUWAR**
- 5. PRATIKSHA CHAVHAN**

## **Abstract**

The motivation behind this application is to provide a platform for users to purchase product on monthly rent basis. The growing popularity and usage of online applications has led to a need to explore the industrial services who could tap into and enhance their services to the customers. Online White Goods rental system curated for things like furniture, appliances, work from home essentials etc. which benefits the user. A rental service is a service in which customers arrive to request the hiring of the rental unit. It is more convenient than carrying the cost of owning and maintaining the unit. In this paper we are introducing an Application – LeaseWithEase Rental Services, which provides services like renting out day-to-day products like furniture, Fridge, TV, fitness gadgets, electronics appliances etc. Our target audience is mainly anyone who prefers renting out products rather than buying them, they may be either localities, or non-localities or the ones who are up to date. This application aims to rent out products for duration ranging from 3 months to a year. It is an extended form of giving out things often organized with numerous local branches and complemented by an application allowing online reservations.

# INDEX

<b>1. Introduction</b>	<b>6</b>
1.1Concept of Project	7
1.2Necessity	7
1.3Problem Specification	7
1.4 Scope	7
<b>2. Literature Survey</b>	<b>9</b>
<b>3. System Requirement Specification(SRS) Document</b>	<b>11</b>
<b>4. System Deployment</b>	<b>17</b>
4.1Proposed System	17
4.2System Architecture	18
4.3Research Methodology	19
4.4Feasibility Study	21
4.5H/W & S/W Requirement	22
4.6UML Diagrams (Use Case Diagram, Activity Diagram, Class Diagram)	23
4.7Data Flow Diagrams (DFD)	27
4.8Sequence Diagram	31
4.9Snapshots of Web pages	34
<b>5. System Deployment</b>	<b>50</b>
5.1Types of Testing	50
5.2Test Plan	52
5.3Test Cases	53
<b>6. Conclusion</b>	<b>54</b>
6.1 Conclusion	54
6.2 Future Scope	55
<b>7. References</b>	<b>56</b>

## **1.INTRODUCTION**

### **1.1Concept of Project**

Nowadays many people are shifting from one city to another for education purpose or for jobs. This has resulted in the creation of a complex and multi- organizational system of management that includes food, bikes, appliances, furniture and other accessories. Current scenario presents you the system for rentals or buying as a very strenuous work. With the massive expansion of Internet and e-commerce technology, Internet platform is providing a lot of services and advantages for online businesses, especially for online shopping portal. Online shopping has expanded in business more effectively and online services are collaborating with customers and other associations. This study uncovered the needs and expectations of current and potential users of virtual marketplaces of the products on temporary basis. This paper constructs the measurement of four-dimensional models that are appropriate for measuring customer satisfaction of online rentals information platform's security. This paper also conducts the factor and multiple regression analysis to verify the measurement model. There has been a long standing debate of whether to buy or rent, there are many applications in today's competitive digital environment.

## **1.2 Necessity**

**24/7 Operation** – Available all the time. A person who wishes to take goods on rent Can book anytime using the proposed system.

**Door Step Delivery** – The booked order will get delivered door step by registered delivery partner.

**Secured Payments** – Guest can quickly reserve the appliances for rent, also they can book the space with multiple and secured payment gateways. Here the transaction will be fast and secured.

**Cut Users Workload** – Through online booking and management system, a business administrator manages the system seamless as it does not deal with manual action. Transporter will have separate login which will be provided by Admin and he can access the Transporter Dashboard. It is convenient to transporter also to take details of delivery online.

## **1.3 Problem Specification**

Using LeaseWithEase platform users of system Admin, Transporter & Customers all three parties problem will get solved. The admin can manage, display products available with him using Admin dashboard. The Customer can view the available products, can booked, pay deposit amount. The transporter can view the booked orders and delivered the ordered products on given address of the customer.

## **1.4 Scope**

All users of the system Admin, Transporter & Customer can sign-up the system. Using there credential they can access there portal. Admin can manage all his business requirement though his dashboard like managing categories, subcategories & products, hiring transporters, display

products images, deciding product images, payment to transporters, deposit amount return to customer etc. Customer will get rent invoice each month on mail.

#### **Modules:**

- Customer management
- Transporter management
- Product management
- Order management
- Payment management
- Stock management
- Report management
- User management
- Feedback management
- Delivery management'

## 2. LITERATURE SURVEY

As our structure relies upon the useful White Goods Renting System which is an authentic application we inspected the present working circumstance of the renting technique. At present renting, organizations are dependent on manual work which consolidates packages of work area work similarly as a human resource.

The Admin can manage available products category, subcategory. Admin will be able to update product details which have been already added to product catalogue by staff.

Admin will be able to appoint new Delivery partner's & can add there mutually agreed conditions for revenue in the system for delivering goods.

Admin will be able to summary of orders been placed, rejected orders, goods retuned details etc. Admin will get Business insight by observing daily operations with the of getting orders information.

Admin will be able to monitor shipment done by their delivery partner, will be able to track product delivery for customer.

Admin will be responsible for returning deposit amount after completion of rental period and also responsible for the payment to delivery agent.

Any **anonymous User** will be able to view different products available for rent. Any User will be able select product to view from categories available.

**Consumer** will be able to register on portal to book required goods to rent. Registered consumer will be able to view, add or remove products from shopping cart. System will maintain shopping cart for each consumer to maintain list of items selected by him/her. Shopping Cart will present goods details, number of items selected by consumer with rent and total. Consumer will be able process for Order placement. Registered Customer will be able to cancel placed order within 24 hours. Registered customer gets orders history. Registered customer can get details of order from orders been placed.

Consumer will be able to extend the rent period for the previously taken goods before completion of booking period. Minimum period for renting goods will be **3 months**.

Also, consumer will be able to place order for return goods before completion of period on certain conditions.

Consumer will have the monthly invoice for rent generated by the staff. After doing payment the details the payment details get updated in system.

Delivery agents will have their personal dashboard by having login credentials provided by admin where they can view details for today's deliveries, locations, and timings for delivery. As per the instructions for delivery they will do arrangement for product delivery at consumer's address & will able to update the status regarding delivery of the order. Delivery person will be able to change the status of product delivered. Delivery vendor will be able to list all product delivery to be done by their staff.

Consumer will be provided with the payment option for initial deposit amount for the selected goods via easy UPI payment, cash on delivery or net banking.

Consumer will be provided options for payment such as through internet banking or UPI or Online payment option.

### **3. System Requirement Specification Document(SRS)**

**Title:** LeaseWithEase Rental Services (White Goods Rental Services)

**Team:**

Direct Customer, Indirect Customer, Architect, Business Analyst  
Quality Assurance Team, System Analyst

**Objective (Purpose):**

The **LeaseWithEase Rental Services** for renting furniture & home appliances is intended to provide complete solution for Consumers as well as internal user through a single Gateway using internet.

It is known the effort it takes to find the perfect rented house to stay in a new city. And known that turning it into a home is not one bit easy. That is why LeaseWithEase Rental Services (White Goods Rental Services) get planned to developed which will be a one stop shop for easily renting furniture, appliances and other home furnishing essentials. With a range of high-quality stylish products to choose at very affordable prices.

**Scope:**

This System allows Admin to maintain their goods for adding or removing from catalogue based on their availability. Customer will be able to view goods, place orders, review orders history and may able to cancel order within 24 hours.

The System will be able to show live Business Operation statistics trends through Customized dashboard for stakeholders.

**Definitions:**

LWS: Lease With Ease  
WGS: White Goods Services  
QA: Quality Assurance  
Portal: Personalized Online Rental Services  
MIS: Management Information System  
CRM: Customer Relation Management  
BI: Business Intelligence

**Requirements:**

## **Functional Requirements:**

**Admin** will be responsible to add new warehouse for new location & can assign a staff to maintain their activities.

Admin will be able to update product details which have been already added to product catalogue by staff.

Admin will be able to appoint new Delivery partner's & can add there mutually agreed conditions for revenue in the system for delivering goods.

Admin will be able to summary of orders been placed, rejected orders, goods retuned details etc. Admin will get Business insight by observing daily operations with the of getting orders information.

Admin will be able to monitor shipment done by their delivery partner, will be able to track product delivery for customer.

Admin will be responsible for returning deposit amount after completion of rental period and also responsible for the payment to delivery agent.

Any **anonymous User** will be able to view different products available for rent. Any User will be able select product to view from categories available.

**Consumer** will be able to register on portal to book required goods to rent. Registered consumer will be able to view, add or remove products from shopping cart. System will maintain shopping cart for each consumer to maintain list of items selected by him/her. Shopping Cart will present goods details, number of items selected by consumer with rent and total. Consumer will be able process for Order placement. Registered Customer will be able to cancel placed order within 24 hours. Registered customer gets orders history. Registered customer can get details of order from orders been placed.

Consumer will be able to extend the rent period for the previously taken goods before completion of booking period. Minimum period for renting goods will be **3 months**.

Also, consumer will be able to place order for return goods before completion of period on certain conditions.

Consumer will have the monthly invoice for rent generated by the staff. After doing payment the details the payment details get updated in system.

Delivery agents will have their personal dashboard by having login credentials provided by admin where they can view details for today's deliveries, locations, and timings for delivery. As per the instructions for delivery they will do arrangement for product delivery at consumer's address & will be able to update the status regarding delivery of the order. Delivery person will be

able to change the status of product delivered. Delivery vendor will be able to list all product delivery to be done by their staff.

Consumer will be provided with the payment option for initial deposit amount for the selected goods via easy UPI payment, cash on delivery or net banking.

Consumer will be provided options for payment such as through internet banking or UPI or Online payment option.

Consumer will be asked to submit their payment related information. Consumer will be redirected to payment gateway for secure payment transaction. On successful payment processing using payment gateway system will notify consumer about transaction and order placement status. Consumer will be able track order status. Customer will be able track delivery using unique dispatcher id presented.

Customer will be able accept or return delivery based on quality received.

Customer will be able to submit feedback about product they received. Customer will be able to get details about product orders placed. Customer will be able to cancel his placed order within 24 hours. Customer will be able to update his/ her personal information.

Customer will be able to manage his profile maintained by system. Customer will be able to change his credentials if required.

Customer will be notified about order status, delivery status through SMS, Email communication.

Customer will get complete information about his orders, likes, comments, details through a dashboard.

Customer will be able to update his payment related information. System will present dashboard for Customers, Admin, Staff & for Delivery Agent. Dashboard will provide information using graph, score cards, key performance indicators as well Grid data presentation. Admin will be informed about business operations through reports.

Customer will be able to get their purchase related information using Customer Dashboard.

### **Non Functional Requirement:**

#### **Security-**

Registered Customer will be able to place an order. Each stakeholder will be able to access system through authentication process.

Who are you ?

System will provide access to the content , operations using Role based security (Authorization) (Permissions based on Role)

Using SSL in all transactions which will be performed stakeholder. It would protect confidential information shared by system to stake holder of Shared by stakeholder to system.

System will automatically log of all stakeholder after some time due to inactiveness.

System will block operations for inactive stakeholder and would redirect for authentication.

System will internally maintain secure communication channel between Servers ( Web Servers, App Servers, database Server)

Sensitive data will be always encrypted across communication.

User proper firewall to protect servers from outside fishing, vulnerable attacks.

### **Reliability-**

The system will backup business data on regular basis and recover in short time duration to keep system operational

Continuous updates are maintained; continuous Administration is done to keep system operational.

During peak hours' system will maintain same user experience by managing load balancing.

**Availability uptime:** 24\* 7 available

### **Maintainability:**

A Commercial database software will be used to maintain System data Persistence.

A readymade Web Server will be installed to host online shopping portal (Web Site) to management server capabilities.

IT operations team will easily monitor and configure System using Administrative tools provided by Servers.

Separate environment will be maintained for system for isolation in production, testing, and development.

**Portability:**

PDA: Portable Device Application System will provide portable User Interface ( HTML, CSS, JS) through users will be able to access online shopping portal.

System can be deployed to single server, multi-server, to any OS, Cloud (Azure or AWS or GCP)

**Accessibility:**

Only registered customer will be able to place an order after authentication.

Staff can reject or approve orders, shopper requests based on role provided.

Admin will be able to view daily, weekly, monthly, annual business Growth through customized dashboard.

**Durability:**

System will retain customer shopping cart for 15 minutes even though customer loose internet connection and join again.

System will maintain wish list for customer. customer will be able to add products from wish list and add to shopping cart whenever needed.

System will implement backup and recovery for retaining stake holders data, business operation data and business data over time.

**Efficiency:**

Whenever maximum number of users will place order, view goods with same response time. System will be able to manage all transactions with isolation.

**Modularity:**

System will design and developed using reusable, independent or dependent business scenarios in the form of modules.

These modules will be loosely coupled and highly cohesive.

System will contain CRM , Inventory , shopping cart, order processing, payment processing, Delivery module, membership and Roles management modules.

**Scalability:**

System will be able to provide consistent user experience to stake holder as well as visitors irrespective of load.

**Safety:**

LWE Rental System will be secure from malicious attack, fishing.

This portal functionalities are protected from outside with proper firewall configuration.

LWE portal will be always kept updated with latest anti-virus software.

Business data will be backed up periodically to ensure safety of data using incremental back up strategy. Role based security will be applied for Application data and operations.

## **4. SYSTEM DEPLOYMENT**

### **4.1. Proposed System**

LeaseWithEase is an application which is based on very simple ideology, we as humans tend to believe more on what we see rather than what is told to us. This is the same ideology used here, this application has two main characters the seller and the buyer, these are two individuals located at different locations who don't know each other, and it provides a common platform for them the seller and buyer to interact. The seller can put up any product on rent, the buyer will view the product, its specifications and will then contact the seller for further information. LeaseWithEase allows creation of a unique account for every individual buyer and seller, the buyer can directly book the product which he wants to buy it on rent.

The need for such an application has become very crucial, time value of money is a concept greatly accepted by majority of the target audience, and money received in the present time is worth more than the same amount received in the future due to the potential earning capacity, wherein money can earn interest. In the evolving market where we are fascinated with brands and tag names we too need to consider the price tags that come with it, the buzzwords like Digitalization, Online Marketing, Sale, Discounts, Trends have completely taken over the market, we wish to have the best of all but going on buying things and stocking them up as One Time Use products or showpieces isn't viable instead we rent out.

Renting is like a win-win situation for both the buyer and seller, products which aren't used much but can still be beneficial to another individual must be rented out, in this way the products will act as an investment for the seller and the buyer too gets to switch his tastes and the major problems like time, space and money have been solved efficiently. LeaseWithEase is a mobile based android application with which easy monitoring and renting tasks can be

performed in an efficient and centralized manner.

### **a) Objective of the Proposed System**

The primary objective of the program is to effectively access the application thoroughly through a completely automated system that not only helps you to save a lot of time but also gives quick outcomes. It is cost effective and well known indicated of mask assessment program.

### **b) Features:**

- Ease to use
- Reliable & Accurate
- Login System is present and secured by password.
- Ability to save all details required to complete all actions by users & admin for business records.
- Transporter dashboard available which will automate door step goods delivery.
- Automated payment/deposit/rent facility available.
- Logout on user demand.

### **c) Advantages of Proposed System :**

- People from different parts of city can register very easily & new system is more personalized.
- All new users can understand all the functionalities of system by user friendly UI provided by the system.
- It is made in a quick and easy referential manner.
- Quick entries can be made in the system.
- Securities for important data are maintained confidentially.
- Mail system is availed for verification & intimation of registration of users, change password and order booking also.
- Generation of delivery challan to ease of transporter.

## **4.2. System Architecture**

A server side dynamic web page is a web page whose construction is controlled by an application in server side scripting, parameters determined how the assembly of every new web page proceeds, including the setting up of more client side processing.

A client side dynamic web page processes the web page using HTML scripting running in the browser as it loads. React JavaScript Library and other scripting languages determine the way the HTML in the received page is parsed into the document object model or DOM, that represents the loaded page. The same client side techniques can then dynamically update or change the DOM in the same way.

A dynamic web page is then reloaded by the user or by a computer program to change some variable content. The updating information could come from the server from or from changes made to that page's DOM. This may or may not truncate the browsing history or create a saved version to go back to, but a dynamic web page update using AJAX technologies will neither create a page to go back to, nor truncate the web browsing history forward of the displayed page. Using AJAX technology, the end user gets one dynamic page manage as a single page in the web browser while the actual web content rendered on that page can vary. The ajax engine seats only on the browser requesting parts of its DOM, the DOM , for its client , from an application server.

### **4.3. Research Methodology**

The different phase of project development that have actually been put to use are as follow;

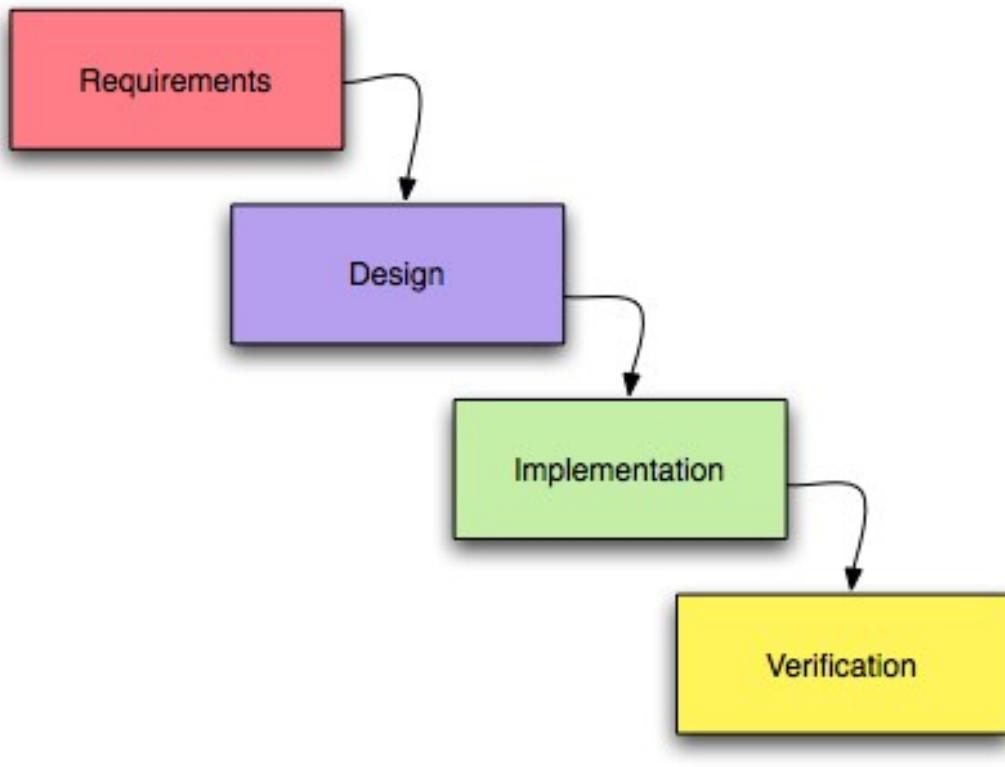


Fig :3.0 Waterfall Model

- **Requirement Gathering and analysis** – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

- **Integration and Testing** – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

## 4.4. Feasibility Study

A feasibility analysis evaluates the project's potential for success; therefore, perceived objectivity is an essential factor in the credibility of the study for potential investors and lending institutions. There are five types of feasibility study—separate areas that a feasibility study examines, described below.

### 1. Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn't want to try to put Star Trek's transporters in their building—currently, this project is not technically feasible.

### 2. Economic Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

### **3. Legal Feasibility**

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let's say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization's ideal location isn't zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

### **4. Operational Feasibility**

This assessment involves undertaking a study to analyze and determine whether—and how well—the organization's needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

### **5. Scheduling Feasibility**

This assessment is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.

When these areas have all been examined, the feasibility analysis helps identify any constraints the proposed project may face, including:

- Internal Project Constraints: Technical, Technology, Budget, Resource, etc.
- Internal Corporate Constraints: Financial, Marketing, Export, etc.
- External Constraints: Logistics, Environment, Laws, and Regulations, etc.

## **4.5. H/W and S/W Requirement**

### **1. Hardware Configuration**

- Processor – Core i5
- Speed – 1.1 Ghz
- RAM – 8 GB Ram
- Hard Disk – 1 TB Hard Disk
- Key Board – Standard Windows Keyboard

### **2. Software Configuration**

Operating System – Windows 10

Front End – React Js

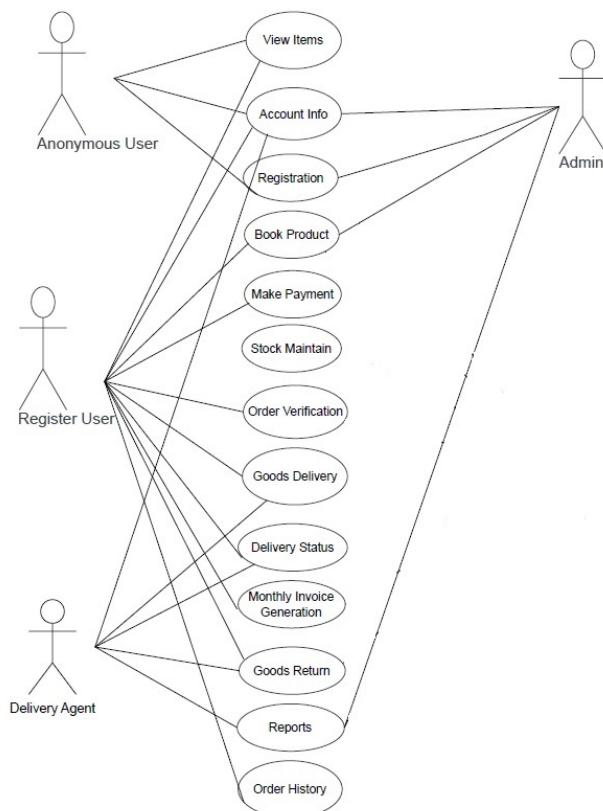
Scripts – JavaScript

Server Side Scripts – Java 8, Spring Boot 2.7.9

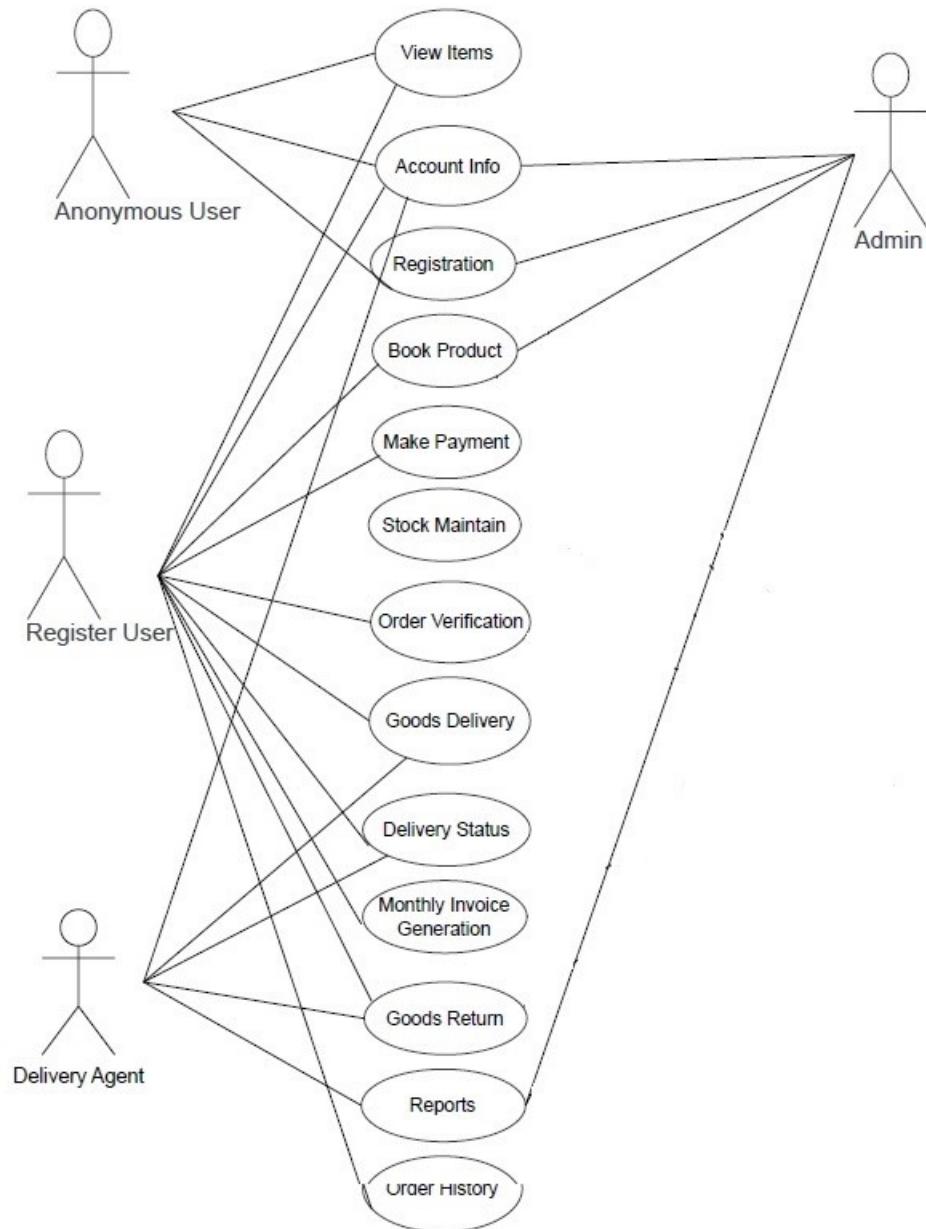
Testing – Postman

Database Connectivity: MySql (using MySql Workbench)

LeaseWithEase Rental Services      **Use Case Diagram**



## Use Case Diagram



**Figure : 3.1 Use Case Diagram**

## **2. Activity Diagram**

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.

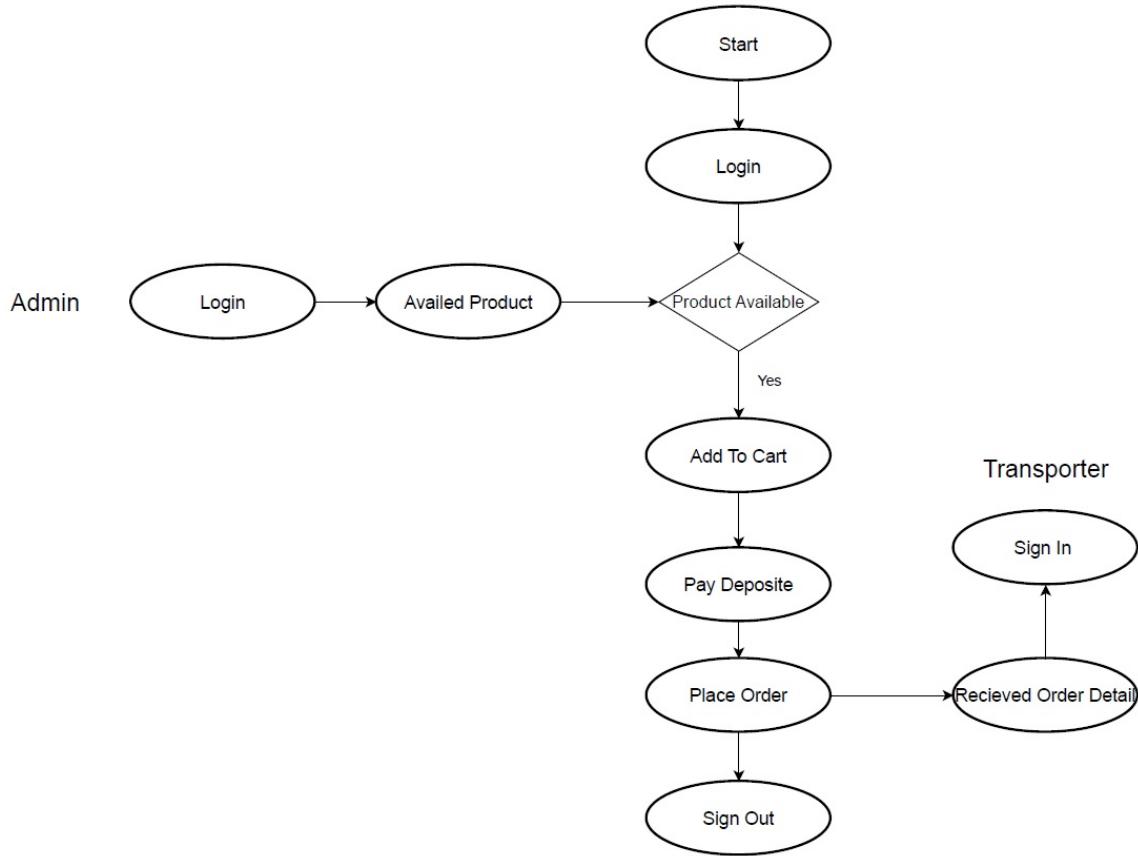
The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

## Activity Diagram - Lease With Ease



**Figure 3.2 Activity Diagram**

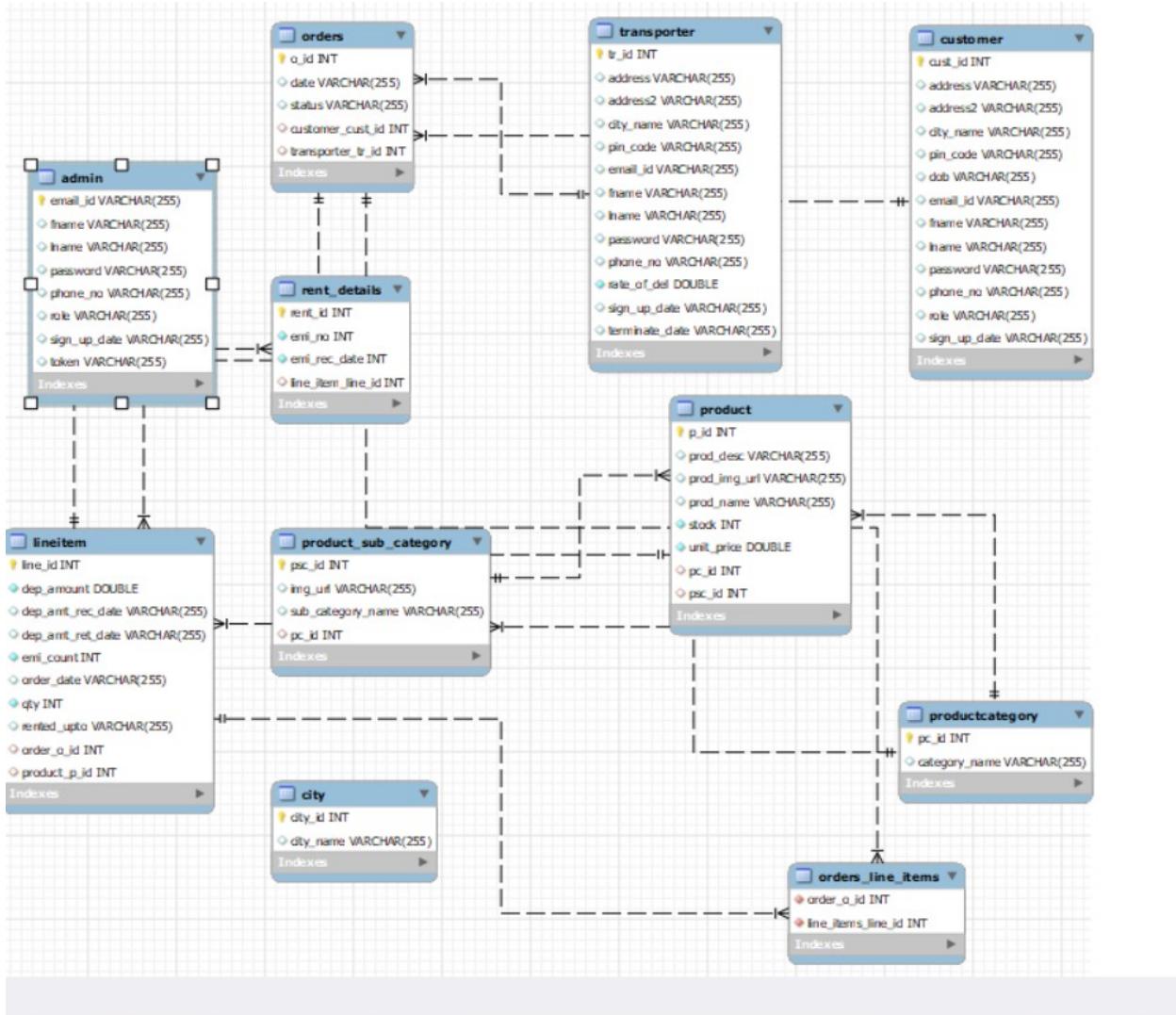
### 3. Class Diagram

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 modelling 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.



**Figure 3.4 Class Diagram**

## 4.7. Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

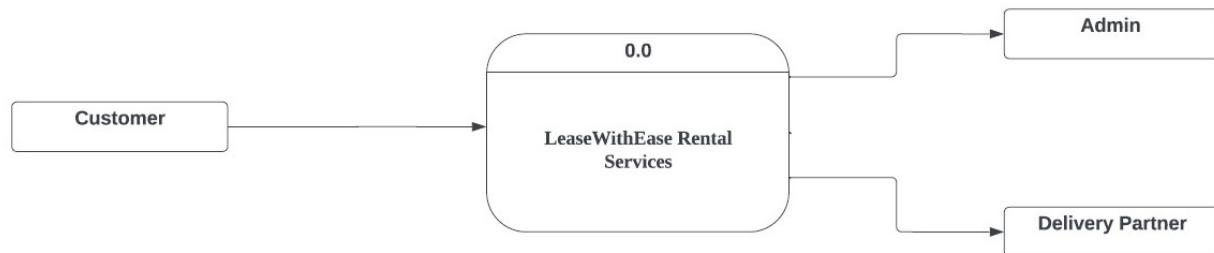
It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

## 0-level DFDM

It is also known as fundamental system model, or context diagram represents the entire software requirement as a single bubble with input and output data denoted by incoming and outgoing arrows. Then the system is decomposed and described as a DFD with multiple bubbles. Parts of the system represented by each of these bubbles are then decomposed and documented as more and more detailed DFDs. This process may be repeated at as many levels as necessary until the program at hand is well understood. It is essential to preserve the number of inputs and outputs between levels, this concept is called levelling by DeMacro.

**Context DFD**



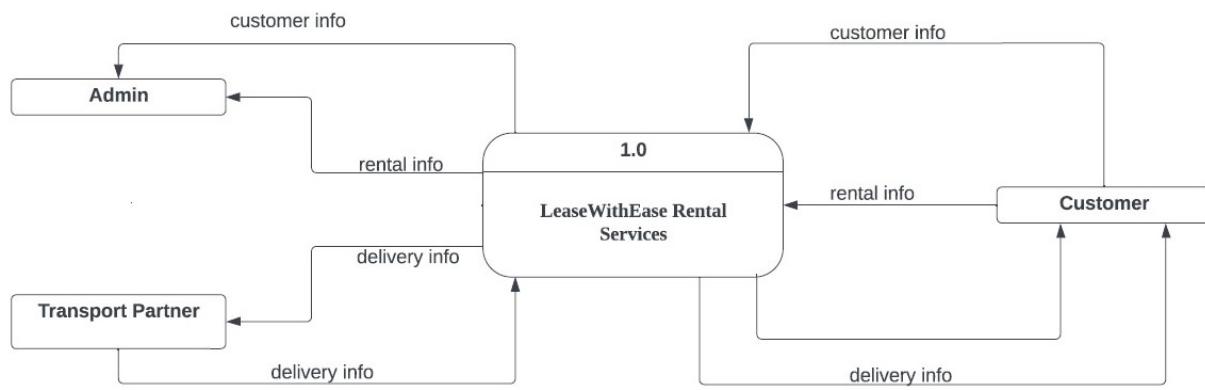
**Figure : Level 0 DFD**

**Figure 3.5 Context DFD**

## 1-level DFD

In 1-level DFD, a context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main objectives of the system and breakdown the high-level process of 0-level DFD into sub processes.

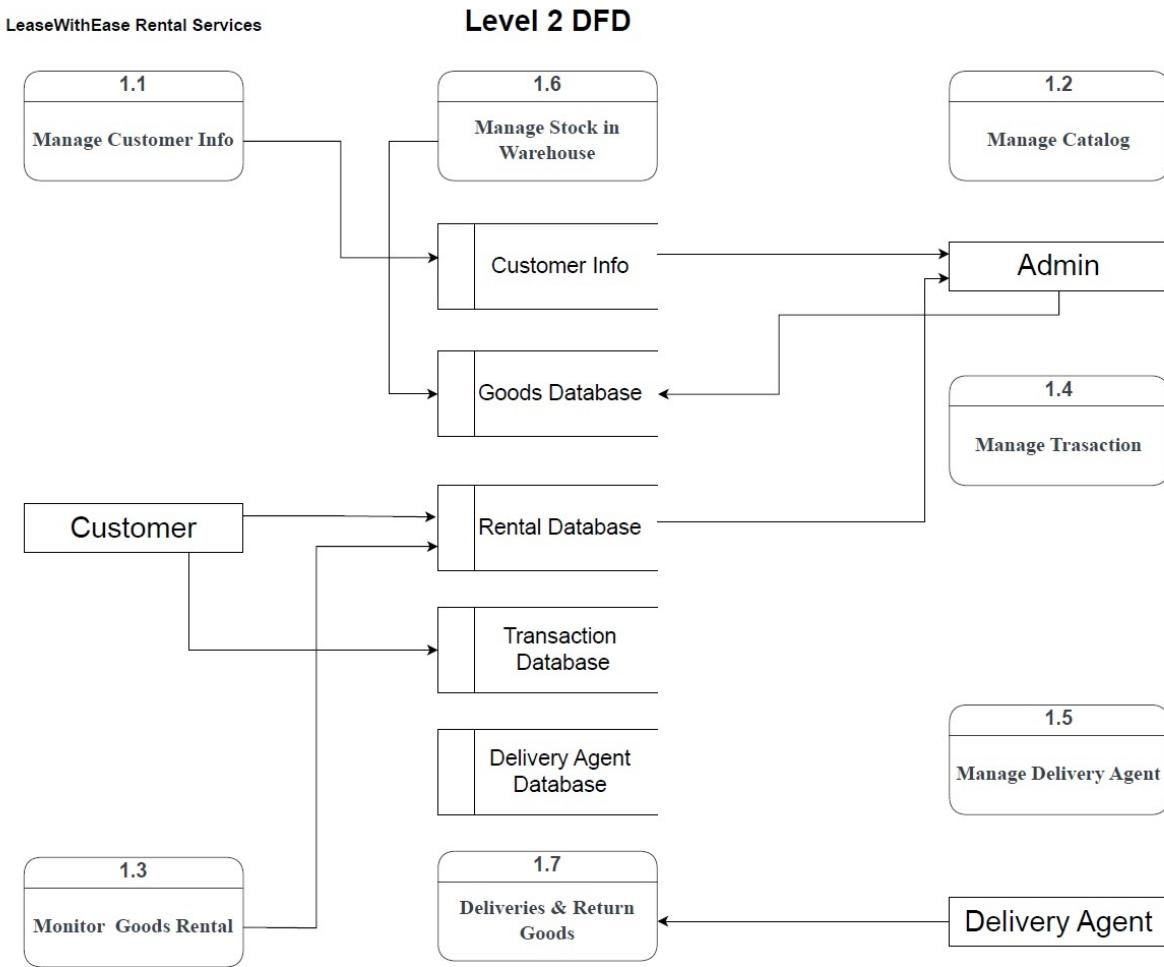
## Level 1 DFD



**Figure 3.6: Level 1 DFD**

## 2-Level DFD

2-level DFD goes one process deeper into parts of 1-level DFD. It can be used to project or record the specific/necessary detail about the system's functioning.



**Figure 3.7: Level 2 DFD**

## 4.8. Sequence Diagrams

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

1. To model high-level interaction among active objects within a system.
2. To model interaction among objects inside a collaboration realizing a use case.
3. It either models generic interactions or some certain instances of interaction.

### 1) Login/Validation Sequence Diagram

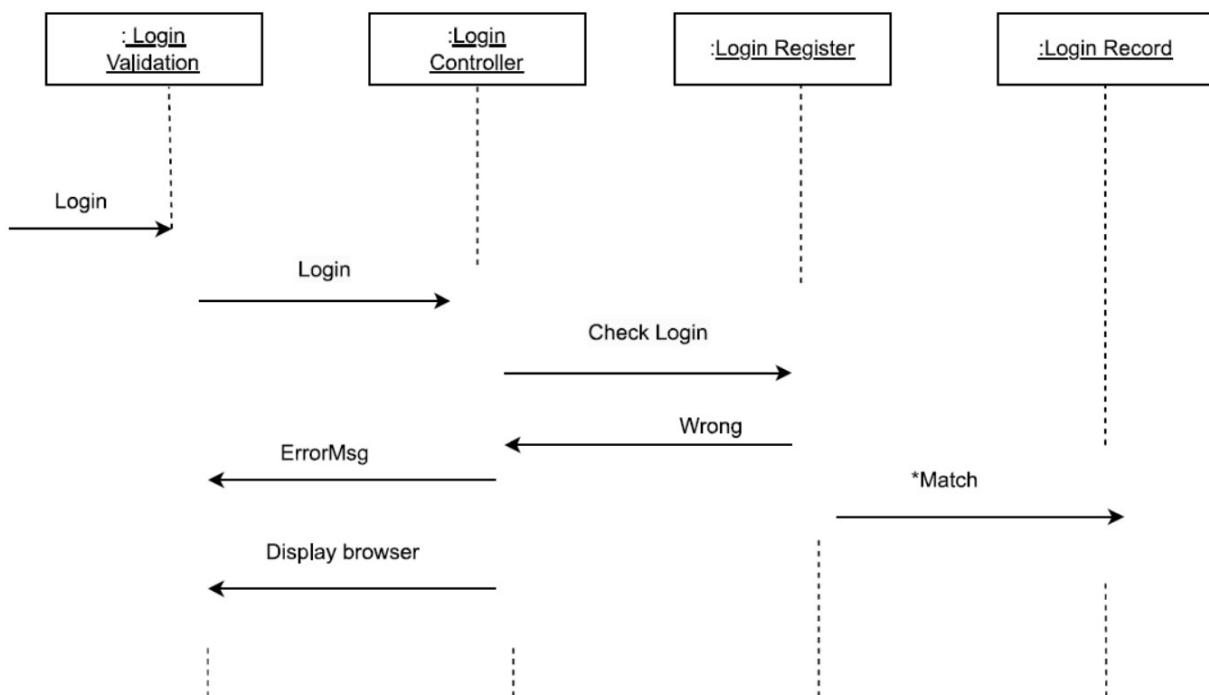
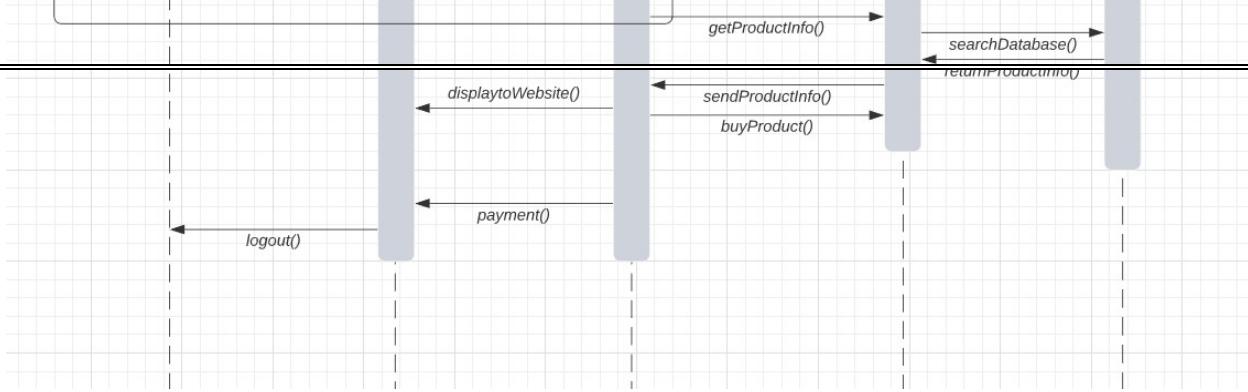
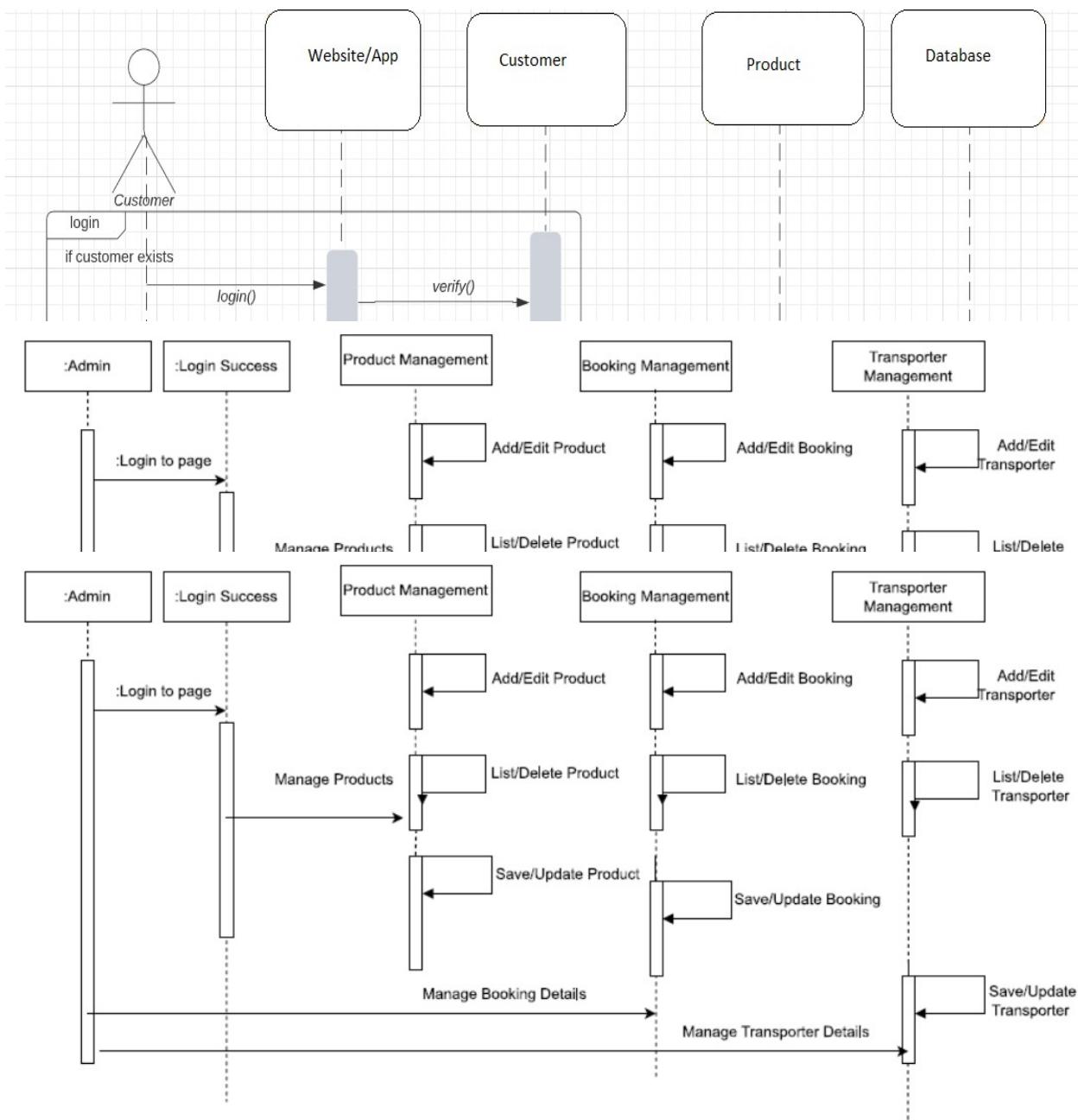


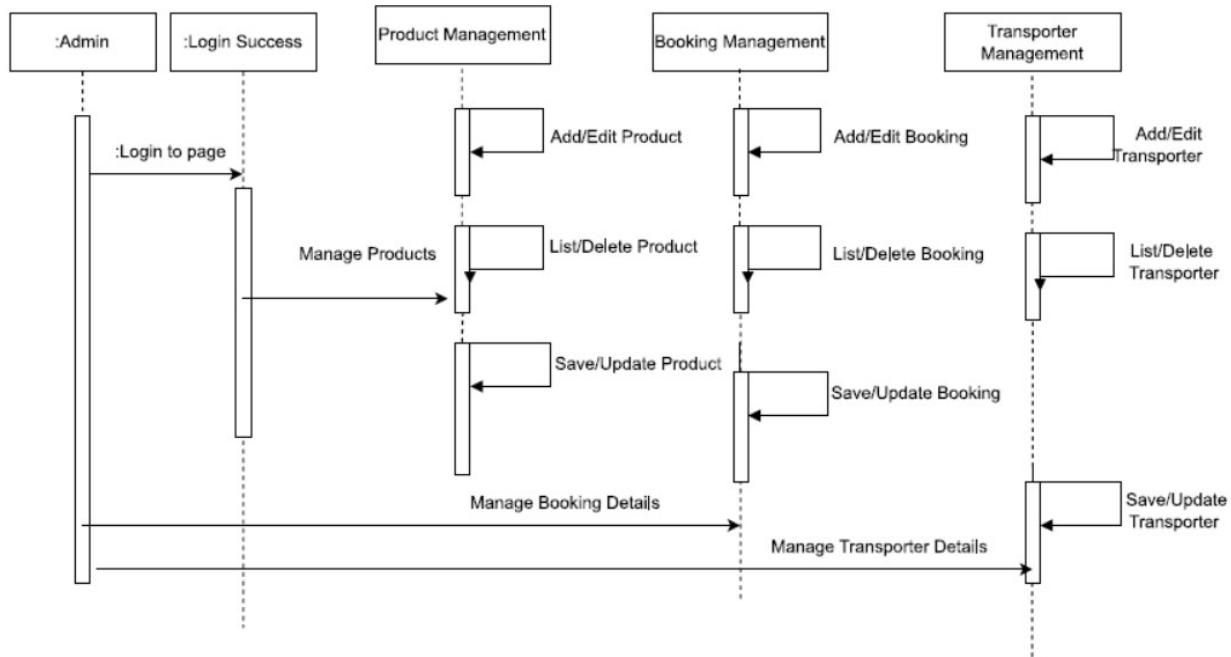
Figure 3.8.1: Sequence Diagram



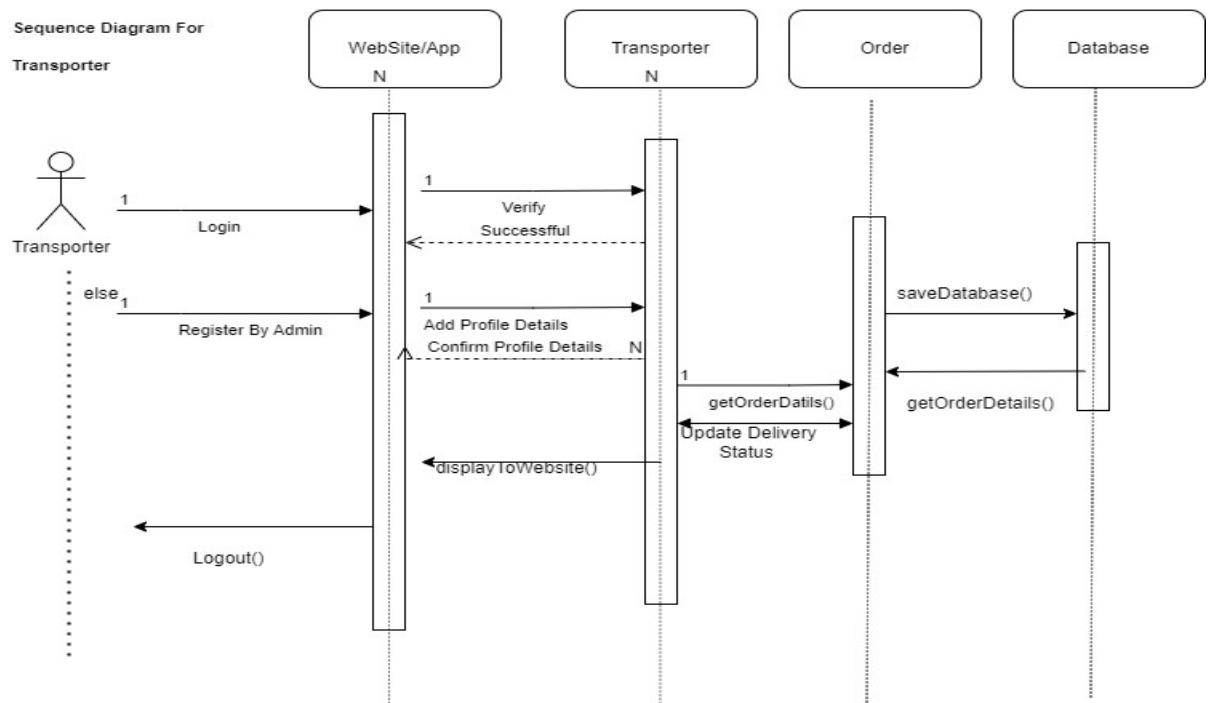
## 2) Customer Sequence Diagram



### 3) Admin Sequence Diagram



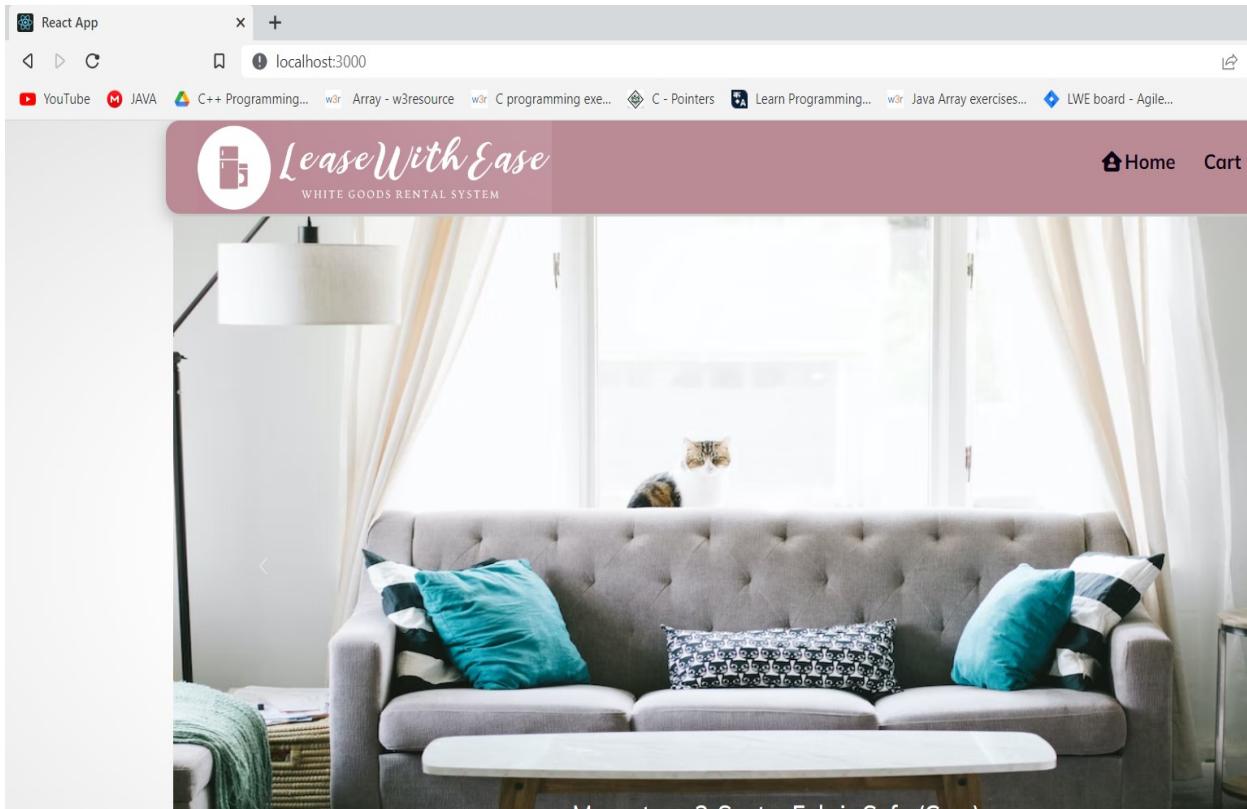
### 4) Transporter Sequence Diagram



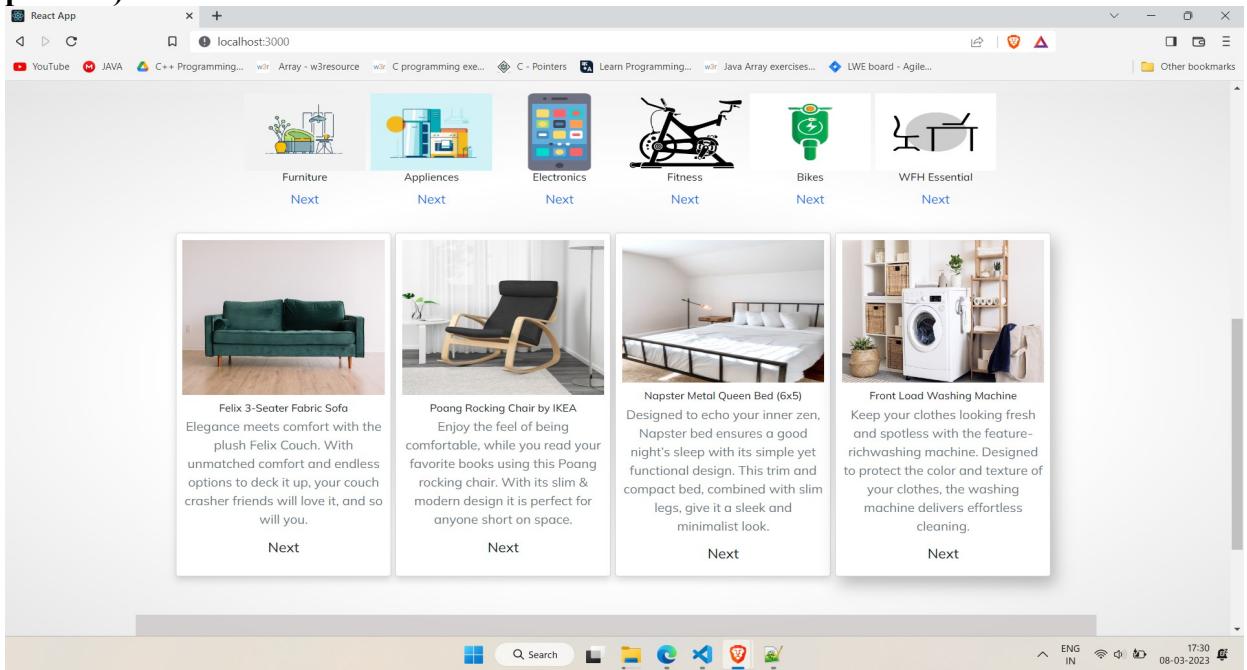
**Figure 3.8.4: Sequence Diagram**

## 4.9. Snapshots of form

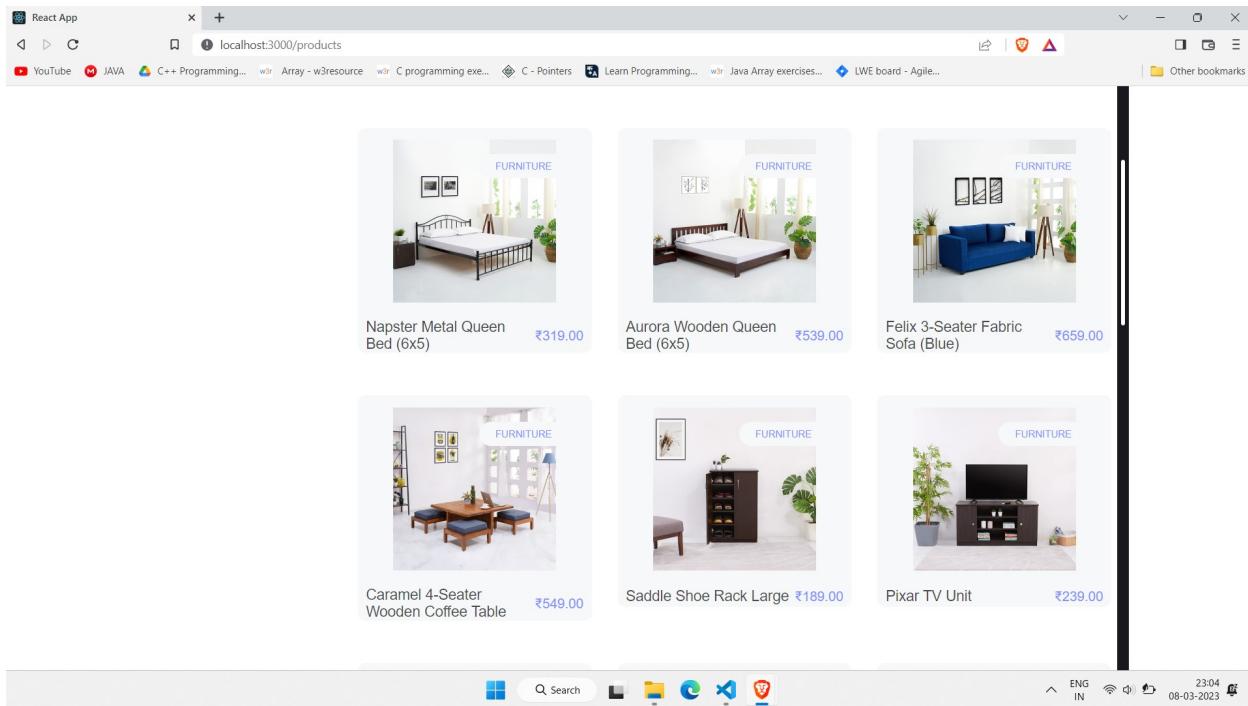
### 1) Home Page



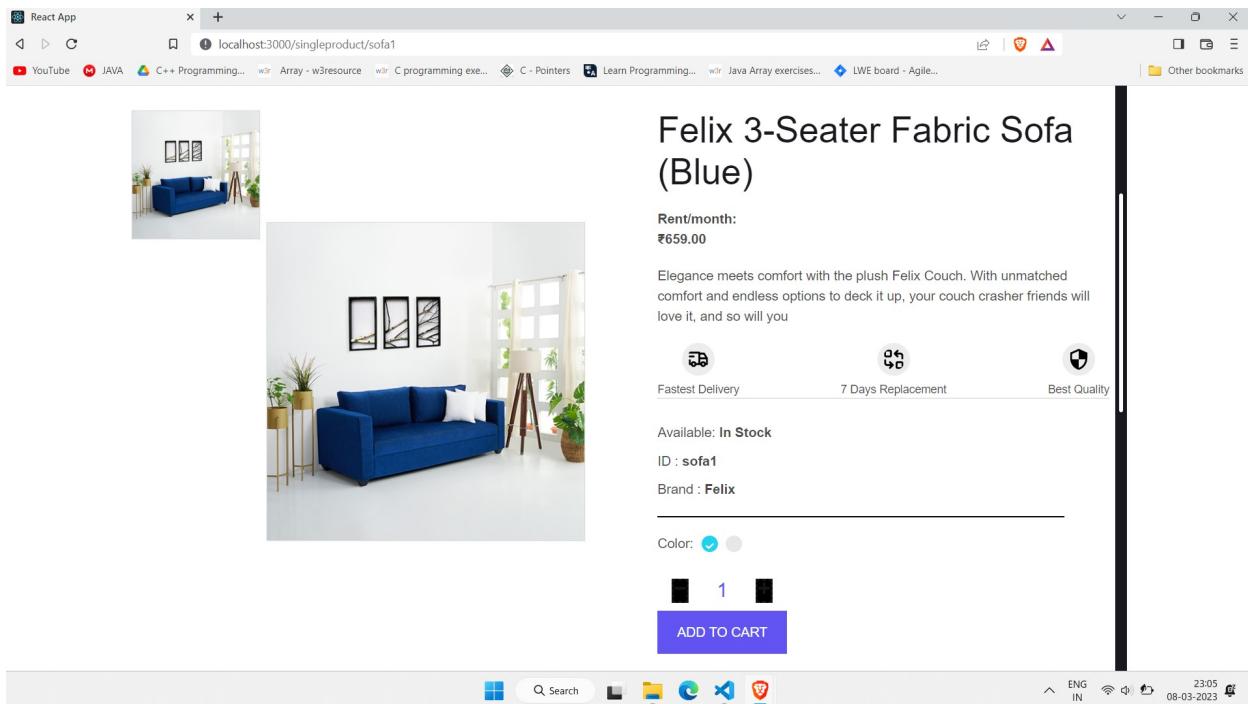
### 2) Home Page (displays all categories & product)



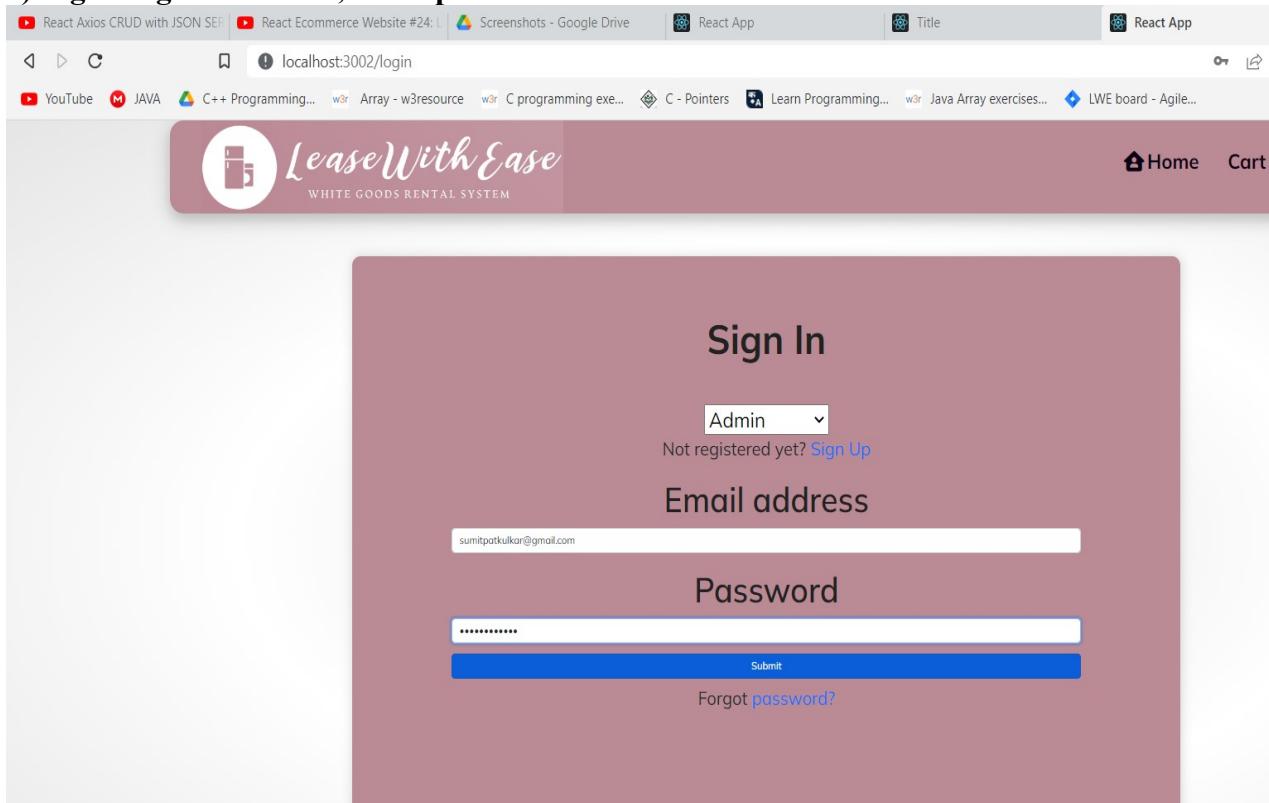
### 3) Category wise Products



### 4) Product Specification Page :



## 5) Login Page for Admin, Transporter & Customer



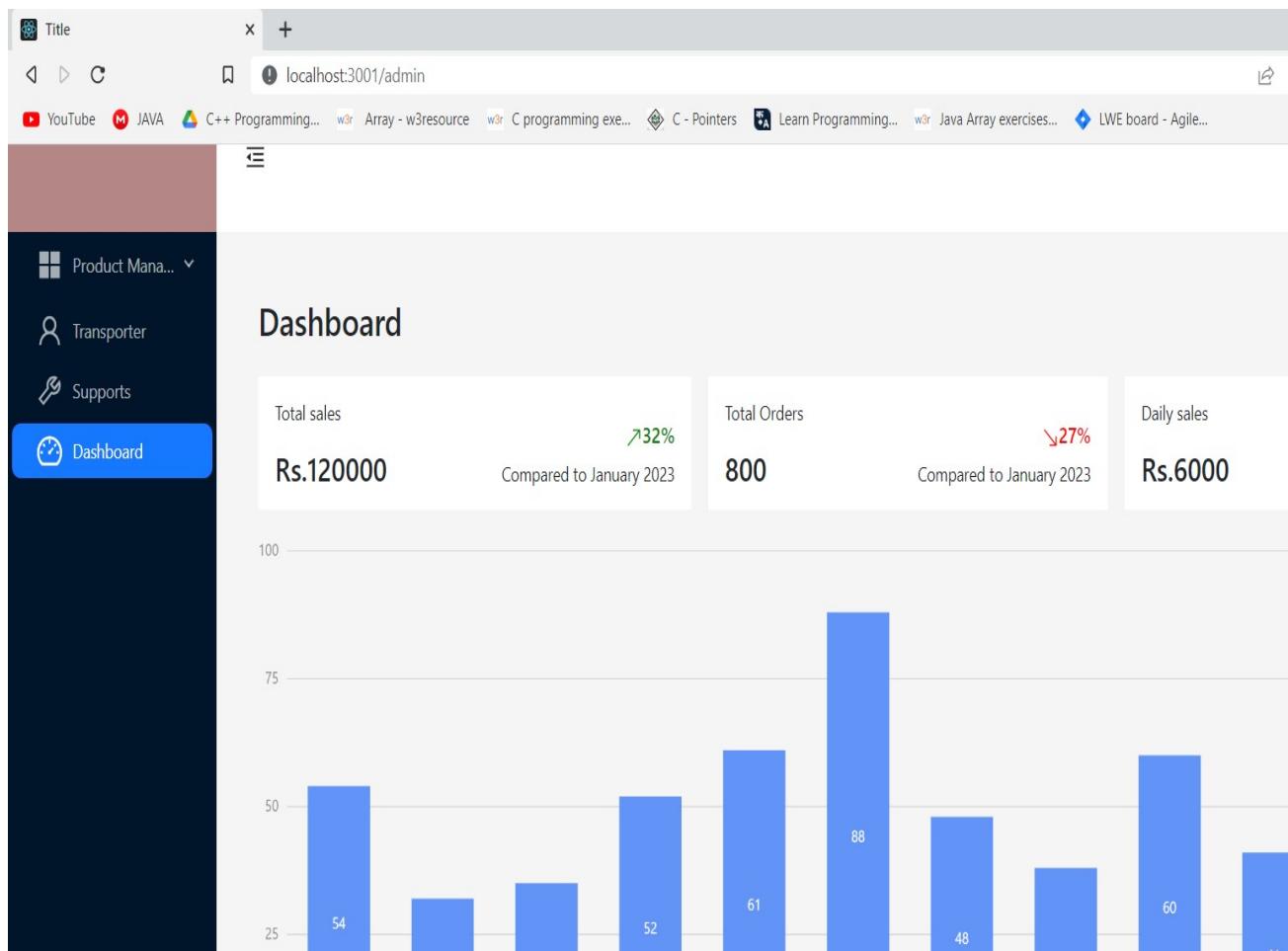
## 6) Shopping Cart

**Shopping Cart**

You Have 4 Items In Shopping Cart

	<b>Rex 3-Seater Leather Sofa (Black)</b> Meet Rex, The Couch That's Here To Redefine Class.	-	<input type="button" value="1"/>	+	659₹	
	<b>Felix 1-Seater Fabric Sofa (Blue)</b> Elegance Meets Comfort With The Plush Felix Couch.	-	<input type="button" value="1"/>	+	479₹	
	<b>Queen Spring Mattress (6x5)</b> This Mattress Is Perfect To Bring Out That Aesthetic Touch.	-	<input type="button" value="1"/>	+	529₹	
	<b>Napster Metal Queen Bed (6x5)</b> Find Ultimate Peace With Napster Bed	-	<input type="button" value="1"/>	+	900₹	

## 7)Admin Dashboard



## 8) Admin Product Management(Product Category)

The screenshot shows the "Category" section of the Admin Product Management interface. The sidebar on the left includes links for Product Manager, Category, Add Category, Update Category, Subcategory, Add Subcategory, Products, Add Product, and Update Product. The main content area displays a table of categories:

ID	Category...	Action
1	Furniture	<button>Edit</button> <button>Delete</button>
2	Appliances	<button>Edit</button> <button>Delete</button>
3	Electronics	<button>Edit</button> <button>Delete</button>
4	Fitness	<button>Edit</button> <button>Delete</button>
5	WFH Essentials	<button>Edit</button> <button>Delete</button>

At the bottom of the table, it says "1-5 of 5" and has navigation arrows. Below the table is a button labeled "Add Category". The top of the window shows the URL "localhost:3001/admin/categoryList" and the title "Title". The bottom right corner shows system status: ENG IN, 19:24, 08-03-2023, and a user icon for "admin@gmail.com".

## 9) Admin Product Management (Product Sub-Category)

The screenshot shows a Windows desktop environment with a browser window titled "Title" open at "localhost:3001/admin/subcategoryList". The browser's address bar also displays "localhost:3001/admin/subcategoryList". The page content is a table listing product sub-categories. The table has columns: ID, Subcate..., Image url, and Action. The data in the table is as follows:

ID	Subcate...	Action
1	Kitchen & Dining	Edit
2	Bedroom	Edit
3	Living Room	Edit
4	Washing Machines	Edit
5	Refrigerators	Edit

Below the table, there is a message "1-5 of 13" followed by navigation arrows. At the bottom of the page is a red button labeled "Add Subcategory". The browser's toolbar includes icons for search, file operations, and other browser functions. The system tray shows the date and time as 08-03-2023.

## 10) Admin Product Management (Product Display)

The screenshot shows a Windows desktop environment with a browser window titled "Title" open at "localhost:3001/admin/productList". The browser's address bar also displays "localhost:3001/admin/productList". The page content is a table listing products. The table has columns: ID, Product Title, Unit Price, Rent, Available Stock, Image Url, and Product Add Date. The data in the table is as follows:

ID	Product Title	Unit Price	Rent	Available Stock	Image Url	Product Add Date
1	Single Wooden Bed	12000	1200	12	https://p.rmjo.in/product...	7/2/2023
2	Dinning Table	24000	2400	16	https://p.rmjo.in/product...	7/2/2023
3	Washing Machine	25000	2500	23	https://p.rmjo.in/product...	7/2/2023
4	Single Door Refrigerator	8000	800	12	https://p.rmjo.in/product...	7/2/2023
5	Foldable Treadmill	86000	8600	8	https://p.rmjo.in/product...	7/2/2023

Below the table, there is a message "1-10 of 10" followed by navigation arrows. At the bottom of the page is a red button labeled "Add Product". The browser's toolbar includes icons for search, file operations, and other browser functions. The system tray shows the date and time as 08-03-2023.

## 11) Product Management (Add New Product)

The screenshot shows the 'Add Product' form. On the left is a sidebar with a dark theme containing a navigation menu:

- Product Menu... ▾
- Category
- Add Category
- Update Cat...
- Subcategory
- Add Subcate...
- Update Subc...
- Products
- Add Product** (highlighted in blue)
- Update Prod...

Below the sidebar is the main content area titled 'Add Product'. It contains the following fields:

- Enter Product Id
- Enter Product Name
- Enter Product Price
- Enter Product Rent
- Enter Product Stock
- Select Category
- Select Subcategory
- Enter Product Description
- Enter Product Add date  
dd-mm-yyyy
- File upload input field with placeholder: Click or drag product image to this area to upload  
Support for a single or bulk upload. Strictly prohibit from uploading company data or other bad files.
- Save Product button

## 12) Transporter Home Page

The screenshot shows the 'Transporter' home page. The browser title is 'React App' and the URL is 'localhost:3003/order-details'. The page has a sidebar on the left with a dark theme:

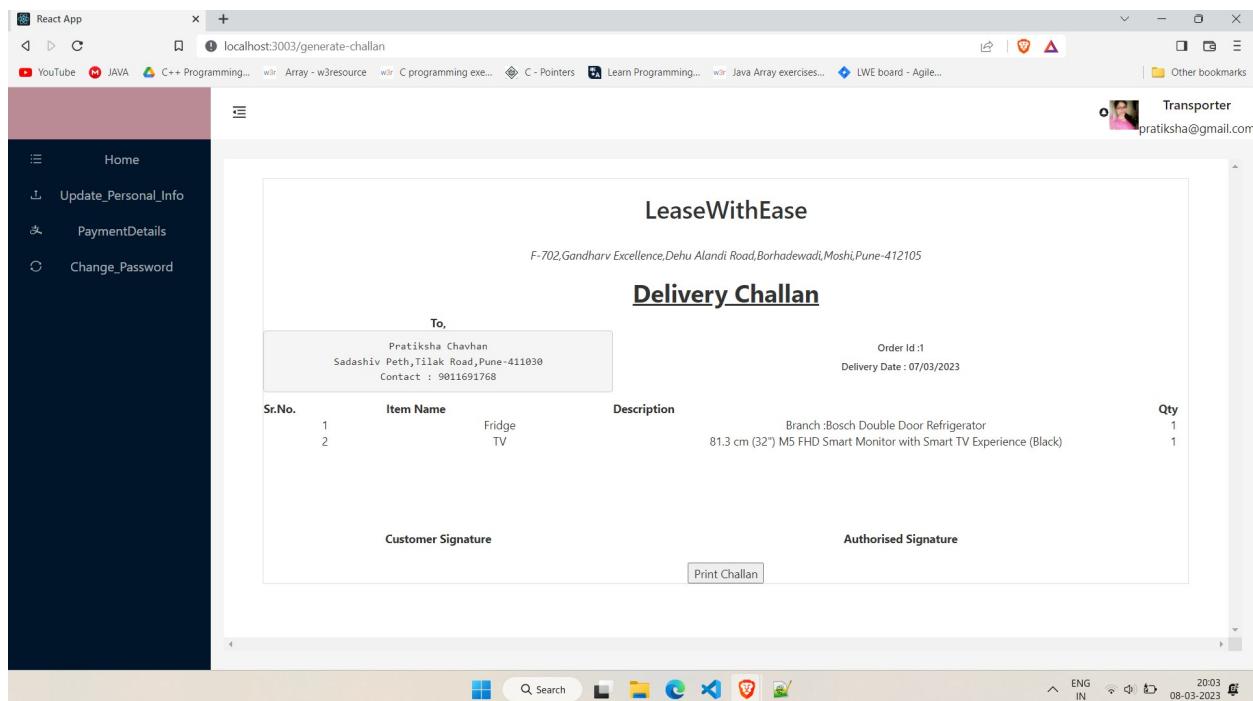
- Home
- Update\_Personal\_Info
- PaymentDetails
- Change\_Password

The main content area displays a table of order details:

OrderId	CustomerName	Address	OrderDate	Product	Quantity	Action
1	Pratiksha	Sadashvpath	03/03/2023	Dining Table	2	Dispatched <input type="button" value="Update"/> Generate Challan
2	Reva	Moshi	03/03/2023	King Size Bed	1	Delivered <input type="button" value="Update"/> Generate Challan
3	Rohini	Vimannagar	03/03/2023	Cupboard	3	Delivered <input type="button" value="Update"/> Generate Challan

At the bottom, there is a navigation bar with icons for search, file, browser, and system status (ENG IN, 20:02, 08-03-2023).

### 13) Generating Delivery Challan-



### 14) Utility page (Cancellation & Return Policy)

The utility page has a green header bar with the title "Cancellation And Return Policy". Below the header, there are four numbered sections with questions and answers:

- 1. How and till when can I modify my order ?**

You can modify your order items up to one day before the scheduled delivery date by reaching us on customer helpline, website/app chat, or email. Please note, date and time of delivery are subject to change due to modification of your order. On changing your order, rent and deposit will be automatically adjusted as per current rates at the time of placing your order. After successful confirmation, your monthly rental amount will be calculated based on the newly chosen items in your order. As a customer and party to the rental contract, your products cannot be shifted from the address mentioned in this delivery address to any other location without notifying Rentomojo. In case of a change of address and assistance with relocation, please contact us two weeks prior to the relocation or raise a relocation request via the customer dashboard for further assistance.
- 2. How can I cancel my existing order ?**

You can cancel an order up until one day before the agreed date of delivery without any extra cost. Once delivered, an existing order cannot be cancelled.
- 3. Can I return a few or all items of the order if I'm not happy with the products at the time of delivery ?**

Yes. While all our products pass a series of stringent quality checks, if you're not happy with the product's condition due to a defect or non-functionality, you may return the same at the time of delivery. If notified at the time of delivery, some exceptions can be considered in a valid case or circumstance. Please note, a return option will not be available at the time of delivery once our team exits your premises after successful confirmation of your order delivery. We also advise checking the product's specified dimensions before placing the order, as item rejection will not be considered eligible for return.
- 4. Can I close my rental subscription prior to the committed tenure ?**

Yes, you can. In case of early termination of your order, you can request for closure for one or all your items by informing us 7 days before the selected preferred date.

## 15) Testing backend On Postman ( Sign Up New Customer)

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:8080/api/customer/signup`. The response body contains the message: "1 New customer added successfully!!".

## 16) Confirmation mail to Customer on Sign up

The email subject is "Registration Confirmation!!". It is addressed to "leasewittheasedac@gmail.com" with a reply-to link. The message content is:

**Hello Sumit ,**

We're super excited to see you join the LeaseWithEase Rental Platform community.

We hope you will be happy with the products offered by the online store and that you will shop with us again and again.

Our goal is to offer the widest range of products offered by the online store at the highest quality.

If you think we should add any items to our store, don't hesitate to contact us and share your feedback.

Until then, enjoy your shopping!

Best,

LeaseWithEase Team

## 17) Testing backend on Postman (Add new Transporter)

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections like Scratch Pad, Collections, APIs, Environments, Mock Servers, Monitors, and History. The main area shows a request for 'http://localhost:8080/api/transporter/addnew' with a POST method. The 'Body' tab is selected, displaying a JSON payload:

```
2
3   "email_id": "pratikshachavhan08@gmail.com",
4   "fname": "Pratiksha",
5   "lname": "Chavhan",
6   "phone_no": "9811691768",
7
8   "addr": {
9     "address": "Sadashiv Peth, *",
10    "address2": "Pune",
11    "city_name": "Pune",
12    "pin_code": "412121"
13  },
14  "rate_of_del": 250.0
15
```

Below the body, the response status is 200 OK with a message: 'New Transporter added successfully!!'. The bottom of the screen shows a taskbar with various icons.

## 18) Mail received to Transporter regarding registration confirmation

Fwd: Registration Confirmation!! Inbox



**Pratiksha Chavhan**

to me ▾

----- Forwarded message -----  
From: <[leasewiththeasedac@gmail.com](mailto:leasewiththeasedac@gmail.com)>  
Date: 8 Mar 2023 11:07 pm  
Subject: Registration Confirmation!!  
To: <[pratikshachavhan08@gmail.com](mailto:pratikshachavhan08@gmail.com)>  
Cc:

**Hello Pratiksha,**

**You are successfully registered on LeaseWithEase as our valuable delivery partner.**

**You are authorised to access all services on transporter portal!!**

**Details for agreement and Login as mentioned Below :**

**Agreement Date : 2023/03/08 23:07:30**

**Delivery Rate : 250.0**

**User Id : [pratikshachavhan08@gmail.com](mailto:pratikshachavhan08@gmail.com)**

**Password: prat@1768**

**Thank You!!**

**Best**

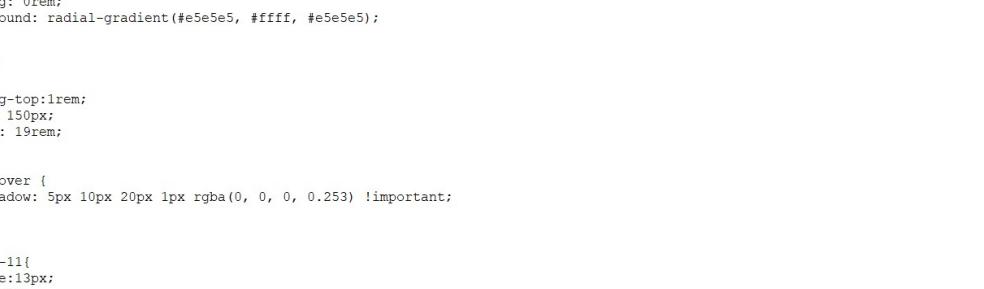
## 19) Use of GET method to get all transporter details –

The screenshot shows the Postman interface. On the left, the 'Scratch Pad' sidebar lists 'Collections', 'APIs', 'Environments', and 'Mock Servers'. The main area displays a 'GET' request to 'http://localhost:8080/api/transporter'. The 'Body' tab is selected, showing a JSON response with 19 lines of code. The response status is 200 OK, and the time taken is 34 ms. The JSON data includes fields like tr\_id, email\_id, fname, lname, password, phone\_no, sign\_up\_date, terminate\_date, address, city\_name, pin\_code, rate\_of\_del, and rate\_of\_del.

## 20) Database view in MYSQL Workbench

The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left lists databases like 'city', 'customer', 'hibernate\_sequence', 'linenum', 'orders', 'orders\_line\_items', 'product', 'product\_sub\_category', 'productcategory', 'rent\_details', 'transporter', 'sakila', 'sys', and 'world'. The 'Schemas' tab is selected. In the center, the 'Query 1' tab is active, showing a query: 'SELECT \* FROM leasewithleasedb.transporter;'. The 'Result Grid' shows one row of data for a transporter. Below the grid, the 'Action Output' pane displays the execution history of the query. The bottom status bar shows the date as 08-03-2023 and the time as 23:23.

## 21) React Code Screen Shots



The screenshot shows a Notepad++ window with the following CSS code:

```
C:\Users\Admin\Downloads\Screenshots\card.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log card.txt
1 body {
2     padding: 0rem;
3     background: radial-gradient(#e5e5e5, #ffff, #e5e5e5);
4 }
5
6 .card1 {
7
8     padding-top:1rem;
9     width: 150px;
10    height: 19rem;
11 }
12
13 .card1:hover {
14     box-shadow: 5px 10px 20px 1px rgba(0, 0, 0, 0.253) !important;
15     ;
16 }
17
18 .col-md-11{
19     font-size:13px;
20     padding: 0%;
21
22     border-radius: 20px;
23     width: 150px;
24 }
25
26 .col-md-12{
27     font-size:13px;
28
29     width: 150px;
30 }
31
32
33 .col-md-13{
34     font-size:13px;
35     padding-left: 15px;
36
37     width: 150px;
38 }
39
```

```
1 import React, { useContext } from "react";
2 import { Scrollbars } from "react-custom-scrollbars-2";
3 import Items from "./Items";
4 import { CartContext } from "./Cart";
5
6 const ContextCart = () => {
7   const { item, clearCart, totalItem, totalAmount } = useContext(CartContext);
8
9   if (item.length === 0) {
10     return (
11       <>
12         <section className="main-cart-section">
13           <h1>shopping Cart</h1>
14           <p className="total-items">
15             you have <span className="total-items-count">{totalItem} </span>{" "}
16             items in shopping cart
17           </p>
18         </section>
19       </>
20     );
21   }
22
23   return (
24     <>
25       <section className="main-cart-section">
26         <h1>shopping Cart</h1>
27         <p className="total-items">
28           you have <span className="total-items-count">{totalItem} </span> items
29           in shopping cart
30         </p>
31
32         <div className="cart-items">
33           <div className="cart-items-container">
34             <Scrollbars>
35               {item.map((curItem) => {
36                 return <Items key={curItem.id} {...curItem} />;
37               })}
38             </Scrollbars>
39           </div>
40         </div>
41       </section>
42     </>
43   );
44
45   <div className="card-total">
46   </div>
47 }
```

```

C:\Users\Admin\Downloads\Screenshots>Login.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log card.txt ContextCart.txt Login.txt
1 import React, { useState } from "react"
2 function Auth(props) {
3   let [authMode, setAuthMode] = useState("signin")
4
5   const changeAuthMode = () => {
6     setAuthMode(authMode === "signin" ? "signup" : "signin")
7   }
8
9   if (authMode === "signin") {
10     return (
11       <div className="Auth-form-container">
12         <form className="Auth-form">
13           <div className="Auth-form-content">
14             <h3 className="Auth-form-title">Sign In</h3>
15             <div className="text-center">
16               Not registered yet?!
17               <span className="link-primary" onClick={changeAuthMode}>
18                 Sign Up
19               </span>
20             </div>
21             <div className="form-group mt-3">
22               <label>Email address</label>
23               <input
24                 type="email"
25                 className="form-control mt-1"
26                 placeholder="Enter email"
27               />
28             </div>
29             <div className="form-group mt-3">
30               <label>Password</label>
31               <input
32                 type="password"
33                 className="form-control mt-1"
34                 placeholder="Enter password"
35               />
36             </div>
37             <div className="d-grid gap-2 mt-3">

```

## 22) Java Spring Boot Rest Code

### Creating maven spring boot project using spring initializer

The screenshot shows the Spring Initializr web application interface. The left sidebar lists tabs: Spring Initializr, Java Downloads, How to Download, Download JDK 8, Project build error, Error On Version, and localhost. Below the sidebar, there are sections for Project, Language, Spring Boot, Project Metadata, and Dependencies.

- Project:** Maven selected (radio button checked).
- Language:** Java selected (radio button checked).
- Spring Boot:** 2.7.10 (SNAPSHOT) selected (radio button checked).
- Project Metadata:**
  - Group: com
  - Artifact: LeaseWithEaseBackend
  - Name: LeaseWithEaseBackend
  - Description: Online Rental Platform
  - Package name: com.LeaseWithEaseBackend
  - Packaging: Jar (radio button checked)
- Dependencies:**
  - Spring Boot DevTools** (Developer Tools): Provides fast application restarts, LiveReload, and configurations development experience.
  - Spring Web** (WEB): Build web, including RESTful, applications using Spring MVC. Use as the default embedded container.
  - Spring Data JPA** (SQL): Persist data in SQL stores with Java Persistence API using Spring Hibernate.
  - MySQL Driver** (SQL): MySQL JDBC driver.
  - Lombok** (Developer Tools): Java annotation library which helps to reduce boilerplate code.

## Entity Class with annotations

The screenshot shows an IDE interface with the following details:

- Project Explorer** (left sidebar): Shows the project structure with files like `ProductController.java`, `ProductSubCategoryController.java`, `TransporterController.java`, and `ProductCategory.java`.
- Code Editor** (center): Displays the `ProductCategory.java` file content. The code defines a class `ProductCategory` with annotations for database mapping and relationships. It includes methods for getting and setting the primary key (`pc_id`), getting and setting the category name, and getting the sub-categories.
- Outline** (left sidebar): Shows the outline of the `ProductCategory` class, listing fields like `pc_id` and `categoryName`, and methods like `getSubCategories()`.
- Console** (bottom): Shows the output for the application named `LeaseWithEaseBackendApplication [Java Application]`.
- Task List** (bottom): Shows a list of tasks.
- Terminal** (bottom): Shows a terminal window.

## Controller class

The screenshot shows an IDE interface with the following details:

- Project Explorer:** Shows the project structure under "com.LeaseWithEaseBackend.Controller".
- Code Editor:** Displays the Java code for `ProductCategoryController.java`. The code includes annotations like `@CrossOrigin`, `@RestController`, and `@RequestMapping` for handling RESTful API requests.
- Outline:** Shows the class hierarchy and methods for `ProductCategoryController`.
- Task List:** Shows a list of tasks related to the current file.
- Console:** Shows the output for the application named `LeaseWithEaseBackendApplication`.
- Tasks & Terminal:** Shows the terminal interface for running commands.

```
package com.LeaseWithEaseBackend.Controller;

import java.util.List;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.ModelAndView;
import com.LeaseWithEaseBackend.Except.*;
import com.LeaseWithEaseBackend.Model.*;

@RestController
@RequestMapping("/api")
public class ProductCategoryController {
    @Autowired
    ProductCategoryService pcService;

    @GetMapping("categories")
    public List<ProductCategory> getAllPCategory() {
        List<ProductCategory> pclist=pcService.getAllCategory();
        return pclist;
    }

    @GetMapping("categories/{pc_id}")
    public ProductCategory getCategoryById(@PathVariable int pc_id) {
        ProductCategory pcat=pcService.getCategoryById(pc_id);
        return pcat;
    }

    @PostMapping("categories/addNew")
    public ResponseEntity<String> addNewCategory(@RequestBody ProductCategory category) {
        pcService.addNewCategory(category);
        return ResponseEntity.ok("New Category Added Successfully!!!");
    }

    @DeleteMapping("categories/{pc_id}")
    public ResponseEntity<String> deleteCategory(@PathVariable int pc_id) {
        pcService.deleteCategory(pc_id);
        return ResponseEntity.ok("Selected Category Deleted Successfully!!!");
    }
}
```

## Custom Exception Class

The screenshot shows an IDE interface with a Java project named "LeaseWithEaseBackend". The code editor displays the file `com.LeaseWithEaseBackend.Exception.BadRequestException.java`. The class implements the `RuntimeException` and extends `BadHttpRequestException` from the `HttpStatus` package. It includes constructors for both message and message with cause, and a static final long serialVersionUID.

```
package com.LeaseWithEaseBackend.Exception;

import org.springframework.http.HttpStatus;

@ResponseStatus(HttpStatus.BAD_REQUEST)
public class BadRequestException extends RuntimeException {
    private static final long serialVersionUID = 1L;

    public BadRequestException (String message) {
        super(message);
    }

    public BadRequestException(String message, Throwable cause) {
        super (message, cause);
    }
}
```

## Service Class

The screenshot shows an IDE interface with a Java project named "LeaseWithEaseBackend". The code editor displays the file `com.LeaseWithEaseBackend.Service.CustomerServiceImpl.java`. The class implements the `CustomerService` interface and uses `CustomerRepository` for operations. It overrides methods to get all customers, get customer by ID, add new customer, delete customer, and update customer.

```
* @Service
public class CustomerServiceImpl implements CustomerService {
    @Autowired
    CustomerRepository customerRepository;

    @Override
    public List<Customer> getAllCustomer() {
        return customerRepository.findAll();
    }

    @Override
    public Customer getCustomerById(int cust_id) {
        Optional<Customer> op=customerRepository.findById(cust_id);
        return op.get();
    }

    @Override
    public void addNewCustomer(Customer cust) {
        customerRepository.save(cust);
    }

    @Override
    public void deleteCustomer(int cust_id) {
        customerRepository.deleteById(cust_id);
    }

    @Override
    public void updateCustomer(Customer cust) {
        Optional<Customer> op=customerRepository.findById(cust.getCust_id());
        ...
    }
}
```

## Repository Class

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

- Project Explorer:** Shows the project structure with packages like com.leaseWithEaseBackend.Repository and com.leaseWithEaseBackend.Repo.
- Code Editor:** Displays the `ProductCategoryRepository.java` file containing the following code:

```
package com.leaseWithEaseBackend.Repository;

import org.springframework.data.jpa.repository.JpaRepository;
@Repository
public interface ProductCategoryRepository extends JpaRepository<ProductCategory, Integer> { }
```
- Outline View:** Shows the class definition and its methods.
- Task List:** Shows the tasks related to the current file.
- Console:** Shows the output of the Java Application.
- Terminal:** Shows the terminal interface.
- System Tray:** Shows the Windows taskbar with icons for search, file explorer, browser, and system status.

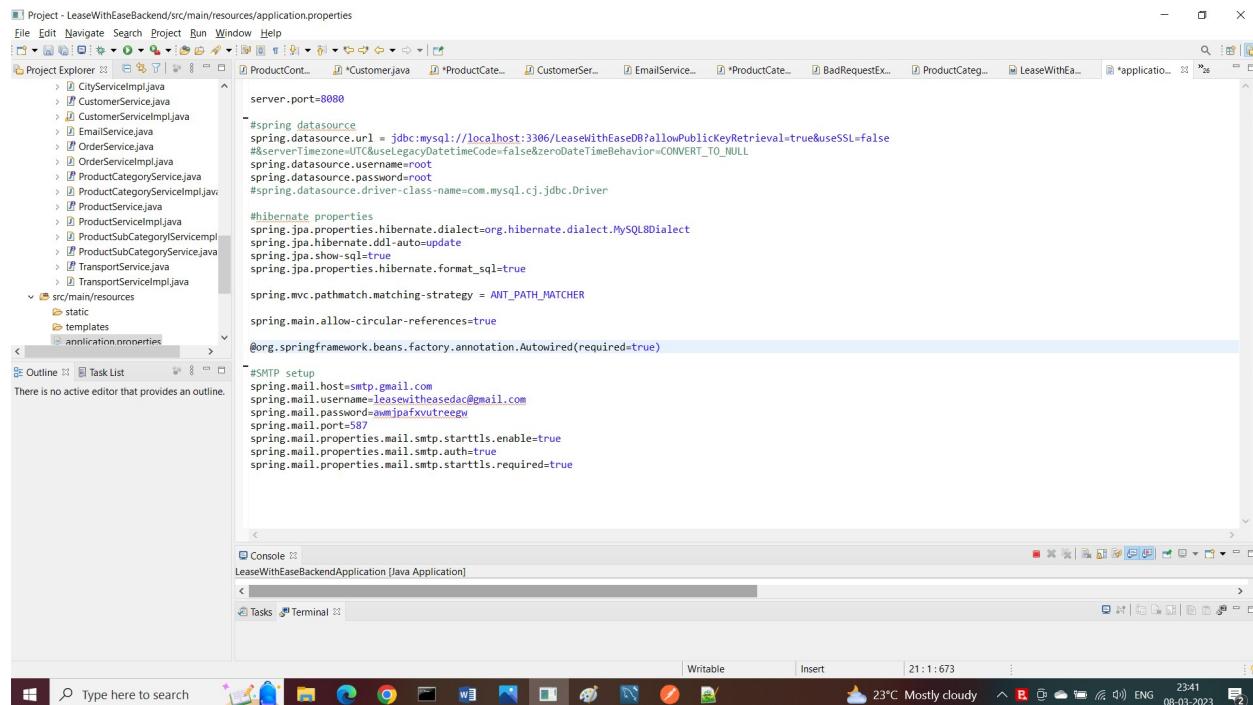
## Pom.xml

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

- Project Explorer:** Shows the project structure with packages like com.leaseWithEaseBackend and com.leaseWithEaseBackend.
- Code Editor:** Displays the `pom.xml` file content:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.9</version>
    <relativePath/> <!-- lookup parent from repository --&gt;
  &lt;/parent&gt;
  &lt;groupId&gt;com.leaseWithEaseBackend&lt;/groupId&gt;
  &lt;artifactId&gt;leaseWithEaseBackend&lt;/artifactId&gt;
  &lt;version&gt;0.0.1-SNAPSHOT&lt;/version&gt;
  &lt;name&gt;leaseWithEaseBackend&lt;/name&gt;
  &lt;description&gt;Online Rental Platform&lt;/description&gt;
  &lt;java.version&gt;1.8&lt;/java.version&gt;
  &lt;properties&gt;
    &lt;dependencies&gt;
      &lt;dependency&gt;
        &lt;groupId&gt;org.springframework.boot&lt;/groupId&gt;
        &lt;artifactId&gt;spring-boot-starter-data-jpa&lt;/artifactId&gt;
        &lt;dependency&gt;
          &lt;groupId&gt;org.springframework.security&lt;/groupId&gt;
          &lt;artifactId&gt;spring-security-crypto&lt;/artifactId&gt;
          &lt;version&gt;5.0.9.RELEASE&lt;/version&gt;
        &lt;/dependency&gt;
      &lt;dependency&gt;
        &lt;groupId&gt;org.springframework.boot&lt;/groupId&gt;
        &lt;artifactId&gt;spring-boot-starter-mail&lt;/artifactId&gt;
      &lt;/dependency&gt;
    &lt;/dependencies&gt;
  &lt;/properties&gt;
  &lt;build&gt;
    &lt;plugins&gt;
      &lt;plugin&gt;
        &lt;groupId&gt;org.springframework.boot&lt;/groupId&gt;
        &lt;artifactId&gt;spring-boot-maven-plugin&lt;/artifactId&gt;
        &lt;configuration&gt;
          &lt;skipTests&gt;true&lt;/skipTests&gt;
        &lt;/configuration&gt;
      &lt;/plugin&gt;
    &lt;/plugins&gt;
  &lt;/build&gt;
&lt;/project&gt;</pre>
```
- Outline View:** Shows the XML structure of the pom.xml file.
- Task List:** Shows the tasks related to the current file.
- Console:** Shows the output of the Java Application.
- Terminal:** Shows the terminal interface.
- System Tray:** Shows the Windows taskbar with icons for search, file explorer, browser, and system status.

## Application. Properties



The screenshot shows the Eclipse IDE interface with the application.properties file open in the editor. The code in the file is as follows:

```
server.port=8080

#spring datasource
spring.datasource.url = jdbc:mysql://localhost:3306/LeaseWithEaseDB?allowPublicKeyRetrieval=true&useSSL=false
#serverTimezone=UTC&useLegacyDatetimeCode=false&zeroDateBehavior=CONVERT_TO_NULL
spring.datasource.username=root
spring.datasource.password=root
#spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

#hibernate properties
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true

spring.mvc.pathmatch.matching-strategy = ANT_PATH_MATCHER

spring.main.allow-circular-references=true

@org.springframework.beans.factory.annotation.Autowired(required=true)

#SMTP setup
spring.mail.host=smtp.gmail.com
spring.mail.username=leasewithaeasedac@gmail.com
spring.mail.password=auimpafxfutreegu
spring.mail.port=587
spring.mail.properties.mail.smtp.starttls.enable=true
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.required=true
```

## **5.SYSTEM DEPLOYMENT**

### **5.1Types of Testing**

#### **1. Unit Testing**

Unit testing is a method of testing individual units or components of a software application. It is typically done by developers and is used to ensure that the individual units of the software are working as intended. Unit tests are usually automated and are designed to test specific parts of the code, such as a particular function or method. Unit testing is done at the lowest level of the software development process, where individual units of code are tested in isolation.

#### **2. Integration Testing**

Integration testing is a method of testing how different units or components of a software application interact with each other. It is used to identify and resolve any issues that may arise when different units of the software are combined. Integration testing is typically done after unit testing and before functional testing, and is used to verify that the different units of the software work together as intended.

#### **3. System Testing**

This software is tested such that it works fine for the different operating systems. It is covered under the black box testing technique. In this, we just focus on the required input and output without focusing on internal working.

**There are seven principles in software testing:**

1. Testing shows the presence of defects
2. Exhaustive testing is not possible
3. Early testing
4. Defect clustering
5. Pesticide paradox
6. Testing is context-dependent
7. Absence of errors fallacy

**Testing shows the presence of defects:** The goal of software testing is to make the software fail. Software testing reduces the presence of defects. Software testing talks about the presence of defects and doesn't talk about the absence of defects. Software testing can ensure that defects are present but it cannot prove that software is defect-free. Even multiple testing can

never ensure that software is 100% bug-free. Testing can reduce the number of defects but not remove all defects.

**Exhaustive testing is not possible:** It is the process of testing the functionality of the software in all possible inputs (valid or invalid) and pre-conditions is known as exhaustive testing. Exhaustive testing is impossible means the software can never test at every test case. It can test only some test cases and assume that the software is correct and it will produce the correct output in every test case. If the software will test every test case then it will take more cost, effort, etc., which is impractical.

**Early Testing:** To find the defect in the software, early test activity shall be started. The defect detected in the early phases of SDLC will be very less expensive. For better performance of software, software testing will start at the initial phase i.e. testing will perform at the requirement analysis phase.

**Defect clustering:** In a project, a small number of modules can contain most of the defects. Pareto Principle to software testing state that 80% of software defect comes from 20% of modules.

**Pesticide paradox:** Repeating the same test cases, again and again, will not find new bugs. So it is necessary to review the test cases and add or update test cases to find new bugs.

**Testing is context-dependent:** The testing approach depends on the context of the software developed. Different types of software need to perform different types of testing. For example, The testing of the e-commerce site is different from the testing of the Android application.

**Absence of errors fallacy:** If a built software is 99% bug-free but it does not follow the user requirement then it is unusable. It is not only necessary that software is 99% bug-free but it is also mandatory to fulfil all the customer requirements.

## 5.2 Test Plan

- First, analyse product structure and architecture.
- Now design the test strategy.
- Define all the test objectives.
- Define the testing area.
- Define all the useable resources.
- Schedule all activities in an appropriate manner.
- Determine all the Test Deliverables.

### 5.3 Test Cases

Sr.No	Test	Expected Result	Result
1	Login	The user should be able to access his account using the preset credentials	Successful
2	Sign-up	Users data to be successfully stored in the database	Successful
3	Forgot-Password	A mail is sent to the registered email-id.	Successful
4	Upload Product	The user should be able to upload the pictures of the product, along with its specification s.	Successful
5	Remove Product	The product no longer exists in the database.	Successful
6	Message passing	Communication is established between the buyer and seller, through email .	Successful
7	Logs	The products previously rented out or put on rent can be viewed along with their reviews.	Successful
8	Sign-out	Exit from the application	Successful

## **6.Conclusion**

### **6.1Conclusion**

Through this paper we conclude that:

With web-based rental management information system, hassle free renting can be provided. There is efficiency in paper procurement for charging the product. The data of all the products is stored in a centralized manner and the costs can be controlled and monitored by the operational manager and owner thus avoiding the over-budgeting. Data storage which is already computerized will ease the process for companies and the users for performing preprocessing, recognizing the buying patterns and maintaining the integrity of the data and use this information to a personal benefit. Through this application we are trying to promote renting out products used on a daily basis instead of buying and discarding them. Our application is user-friendly, open source and is Free to use. It positively impacts the environmental situation by using fewer products more number of times. Hiring products provides a simple way of collecting useful information to measure this service. Concentrating on customer satisfaction and the four dimensions,

“Reliability”, “Responsiveness”, “Tangibles” and “Quality” helps us to serve the users in a better manner and thus give us a competitive edge over the others.

### **6.2 Future Scope**

#### ***1. Location Wise Filtering***

This is the next add on we are currently working on,

wherein using Google Maps services, we would be connecting buyers and sellers staying in the same or nearby locality, this would the process of renting out. This way we target to mainly

saving on travelling time and enhance the process of renting out amongst the users.

## ***2. Validity Extension***

When the seller keys in the product details, there is an option stating for how long does he wish to rent out his/her product, the buyer and the seller agrees upon a fixed time and price, sometimes the buyer may like to keep the product a little longer than the time agreed upon, in this scenario we would be adding new functionality of validity extension. Here, the buyer can notify the seller through a text message or an email that he or she would be renting the product a little longer than expected, the renting prices would now be negotiated accordingly.

## ***3. Recommendation***

We would be extending our domain knowledge in the field of information filtering system, we would be studying the user profile, product profile, past rental purchase, the products mostly rented, its duration and make a suggestion to the buyer regarding the best deals. For this we would be using content-based filtering techniques.

## ***4. Comparisons***

Another new feature which would compare the buyers filtered out product with different seller options, and provides the buyer with the best deal.

## **References**

- 1) Learning Spring Boot 2.0 - Second Edition: Simplify the development of lightning fast applications based on micro services and reactive programming 2nd Revised edition by Greg L. Turnquist
- 2) Spring in Action 5th Edition by Craig Walls (Author)
- 3) The Road to React: The React.js with Hooks in JavaScript Book by Robin Wieruch
- 4) Spring Microservices in Action 1st Edition by John Carnell
- 5) Efficient MySQL Performance: Best Practices and Techniques 1st Edition by Daniel Nichter
- 6) MySQL and JSON: A Practical Programming Guide 1st Edition by David Stokes