

INDUS UNIVERSITY



RENTMENT APPLICATION

A thesis submitted by

Adeel Muhammad (1391-2020)

Farah Riaz (940-2020)

Irfanullah (1313-2020)

Supervisor

Dr. Syed Zafar Nasir

Co-Supervisor

Syed Mushtahir Uddin

THESIS SUBMITTED TOWARDS THE PARTIAL FULFILMENT OF THE
REQUIREMENT FOR THE AWARD OF BACHLOR'S DEGREE IN

.....

Department Of Computing , FCIT
INDUS UNIVERSITY Pakistan
SPRING-2021

CERTIFICATE

This is to certify that the work present in this thesis entitled “RentMent Application” has been carried out by **Farah Riaz, Adeel Muhammad and Irfanullah** under our supervision. The work is genuine, original and, in our opinion, suitable for submission to the Indus University for the award of degree of PhD/MS/MPhil in

SUPERVISOR

Dr. Syed Zafar Nasir

Dean
Department Of FCIT
Indus University Pakistan

CO-SUPERVISOR

Syed Mushtahir Uddin

Lecturer
Department Of FCIT
Indus University Pakistan

DEDICATION

DEDICATED

TO

OUR

**RESPECTED ALLAH (SWT) & PROPHET MUHAMMAD
(PBUH)**

&

PARENTS, TEACHERS, BROTHERS, SISTERS, FRIENDS



FOR

THEIR CONSTANT

AFFECTION AND SUPPORT

ACKNOWLEDGEMENT

‘Then which of the Blessings of your Lord will you deny’

(Surah Ar-Rehman)

First, to our creator, our life coach, the most gracious, the most beneficent, ALLAH S.W.T, we owe it all to you, Thank you! There have been many people who have walked alongside us, who have guided us through all project efforts. We would like to outstretch gratitude to each of them. We would like to extend special gratitude to my supervisor, Dr Syed Zafar Nasir and co-supervisor Sir Mushtahir, whose contributions in simulating suggestions and encouragement, helped us to coordinate my project work and especially in achieving the results. It was because of your support and guidance from the beginning that I have done it!

Furthermore, we also like to acknowledge with much appreciation the crucial role of our Teachers **Mam Shagufta Aftab** and **Sir Mushtahir**, for their support, encouragement and technical advice throughout research work. Without you it was not possible!

Here I am indebted to our parents especially, for his efforts and encouragement throughout our educational career. It was you who stood by my side at every difficult moment and kept our morale high. We would also like to say Special thanks to our Mother for their motivation, prayers and moral support I needed throughout this journey.

At last we must express our sincere heartfelt gratitude to all the staff members of our Department who helped us directly or indirectly during completion of this project.

ABSTRACT

The RentMent application project aims to develop a user-friendly and efficient platform for facilitating the renting and leasing of various items or services. The application provides a seamless interface for both users and vendors, enabling them to connect, engage in transactions, and manage their rental activities. The application allows users to search and browse through a wide range of rental listings, including items such as tools, equipment, vehicles, properties, and services.

Users can create accounts, customize their profiles, and securely communicate with vendors to inquire about availability, negotiate terms, and make reservations. Vendors can register on the platform, create comprehensive listings with detailed descriptions, pricing, and availability information. They can manage their inventory, respond to user inquiries, confirm bookings, and track transaction history.

Vendors can also receive feedback and reviews from users, helping build trust and credibility within the community. The RentMent application incorporates various features to enhance user experience and streamline the rental process.

These include advanced search filters, secure payment processing, rating and review systems, notifications for booking confirmations and reminders, and a user-friendly interface optimized for both web and mobile devices.

The project follows an agile methodology, allowing for iterative development, frequent feedback, and continuous improvement. This approach enables adaptability to changing requirements and ensures a high-quality, user-centric RentMent application. By providing a convenient and reliable platform, the RentMent application project aims to simplify the rental experience, foster trust between users and vendors, and promote the sharing economy by facilitating the efficient utilization of resources.

TABLE OF CONTENTS

CERTIFICATE.....	I
DEDICATION	III
ACKNOWLEDGEMENTS	IV
ABSTRACT	V
TABLE OF CONTENTS.....	VI
LIST OF TABLES	VIII
LIST OF FIGURES.....	IX
ABBREVIATIONS	X
CHAPTER 1	1
INTRODUCTION	1
1.1 MOTIVATION AND PROJECT OVERVIEW	2
1.3 PROBLEM STATEMENT.....	2
1.4 AIM AND OBJECTIVES.....	2
1.5 SCOPE OF THE PROJECT.....	3
1.6 CONTRIBUTIONS OF THE THESIS	10
1.7 STRUCTURE OF THE THESIS	11
CHAPTER 2.....	12
LITERATURE REVIEW	12
2.1 BACKGROUND	12
2.1.1 WHY WE USE MERN STACK FOR RENTMENT.....	13
2.2 RELATED WORK	14
2.3 COMPARATIVE STUDY	18
2.4 SUMMARY	19
CHAPTER 3 RESEARCH /SYSTEM DESIGN	20
3.1 RESEARCH /SYSTEM METHODOLOGY	20
3.1.1 SOFTWARE REQUIREMENTS.....	22
3.1.2 HARDWARE REQUIREMENTSS.....	23
3.1.2 SYSTEM CONSTRAINTS	23
3.1.2 EXTERNAL INTERFACE REQUIREMENTS.....	24
3.2 FUNCTIONAL REQUIREMENTS	26
3.2.1 USE CASES	29

RENTMENT APPLICATION

3.2.2 ACTIVITY DIAGRAM.....	35
3.2.3 SEQUENCE DIAGRAM	38
3.2.4 ER-DIAGRAM	40
3.3 NON-FUNCTIONAL REQUIREMENTS	41
3.3.1 PERFORMANCE REQUIREMENTS	41
3.3.2 SECURITY REQUIREMENTS	41
3.4 SUMMARY.....	42
CHAPTER 4 SYSTEM IMPLEMENTATION.....	43
4.1 SYSTEM DEVELOPMENT PROCESS	43
4.2 SIMULATION MODEL	47
4.3 SYSTEM ARCHITECTURE	50
4.4 MATHEMATICAL MODEL	47
4.5 ALGORITHMS/ API/ PACKAGES USED	47
4.6 DATABASE DESIGN AND DEVELOPMENT.....	53
4.7 SUMMARY.....	56
CHAPTER 5	57
RESULTS AND DISSCUSSION.....	57
5.1 RESULTS OF USER SCREENS	57
5.2 RESULTS OF VENDOR SCREENS.....	62
5.3 RESULTS OF ADMIN SCREENS	63
5.4 SUMMARY.....	68
CHAPTER 6.....	69
CONCLUSION AND FUTURE DIRECTIONS.....	69
6.1 CONCLUSION.....	69
6.2 FUTURE DIRECTIONS	70
REFRENCE.....	72

LIST OF TABLES

Table 2-1 Comparision Table.....	17
Table 3-1 Key Aspects of Agile.....	21
Table 3-2 Software Requirements Table.....	22

LIST OF FIGURES

Figure 2-1 Comparative Chart.....	18
Figure 3-1 Agile Methodology.....	20
Figure 3-2 User case Diagram for user and vendor login.....	29
Figure 3-3 Use Case Diagram for user and vendor Registration.....	30
Figure 3-4 Use case diagram for Admin.....	32
Figure 3-5 Use Case Diagram for RentMent Application.....	33
Figure 3-6 Activity Diagram For user and vendor Login.....	34
Figure 3-7 Activity Diagram For user and vendor Registration.....	36
Figure 3-8 Activity Diagram For upload product.....	37
Figure 3-9 Rental Sequence Diagram.....	38
Figure 3-10 Add Product Sequence Diagram.....	39
Figure 3-11 ER Diagram.....	40
Figure 4-1 Agile SDLC.....	44
Figure 4-2 Software Development Life cycle.....	43
Figure 4-3 Simulation Model.....	47
Figure 4-4 System Architecture.....	50

ABBREVIATIONS

ID: Identification

UI: User Interface

PP: Privacy Policy

UX: User Experience

TOS: Terms of Service

IDV: Identity Verification

KYV: Know Your Vendor

T&C: Terms and Conditions

KYC: Know Your Customer

FAQ: Frequently Asked Questions

API: Application Programming Interface

VRM: Vendor Relationship Management

CNIC: Computerized National Identity Card

HTTPS: Hypertext Transfer Protocol Secure

CHAPTER 1

INTRODUCTION

In this chapter of RentMent application project, typically introduce the core ideas and foundational elements of the project. This chapter will set the stage for our project, explaining why it's necessary, what it aims to accomplish, and how you intend to achieve those goals.

1.1 MOTIVATION AND PROJECT OVERVIEW

The motivation behind developing a RentMent application with product verification, agreements, and CNIC certification is to address the needs and challenges of the rental industry while enhancing security and convenience for both users and vendors. The project overview involves creating a user-friendly and efficient platform that enables individuals or businesses to rent out their products while ensuring the safety and authenticity of the transactions.

The key motivations and benefits of this project are as follows:

- 1.1.1 Convenience:** It eliminates the need for physical visits to rental stores or extensive phone calls, making the entire process more accessible and timesaving.
- 1.1.2 Security:** This verification process helps build trust between users and vendors, reducing the risk of fraudulent or misrepresented listings.
- 1.1.3 Agreements:** This ensures that both parties understand their responsibilities and rights, minimizing disputes and providing a legal framework for the rental process.
- 1.1.4 CNIC Certification:** CNIC (Computerized National Identity Card) helps in verifying the identities of the individuals involved, reducing the chances of fraudulent activities.
- 1.1.5 Product Availability Status:** This feature helps streamline the rental process by avoiding unnecessary inquiries for products.

1.2 BACKGROUND

The background of this product stems from the challenges and concerns associated with the rental industry. Many users face issues related to the quality and reliability of products, transparency in transactions, and trustworthiness of vendors.

Existing rental platforms often lack a standardized approach to certifying vendors and ensuring the quality of products, leading to a lack of confidence among users. In the context of the rental industry, traditional methods of renting products often involve time-consuming processes, limited availability information, and potential risks related to the authenticity and condition of the rented items.

These challenges have created an opportunity for the development of a cross-platform RentMent application that addresses these pain points and provides a more streamlined and secure rental experience.

The background of this project can also be influenced by the need for enhanced security measures in rental transactions. Fraudulent activities, misrepresented products, and disputes are common concerns in the rental industry. By incorporating product verification, agreements, and CNIC certification, the project aims to mitigate these risks and build trust between users and vendors.

1.3 PROBLEM STATEMENT

The problem statement highlights the issues faced by users and product owners in existing rental platforms, such as inadequate user verification, lack of clear damage and return policies, and insufficient product condition verification. It emphasizes the need for a rental cross-platform application that addresses these challenges and provides a secure, trustworthy, and convenient environment for both parties involved in the rental process.

1.4 AIM AND OBJECTIVES

The aims and objectives section outlines the goals of the rental cross-platform web application. It aims to create a platform that ensures proper user verification through NIC, name, email, address, and phone number verification. It also focuses

RENTMENT APPLICATION

on implementing a product damage policy and return policy to protect both users and product owners. Additionally, the application aims to establish a pre-rental and post-rental verification process to document the product's condition before and after each rental transaction.

1.4.1 Key Features:

1.4.1.1 Product Rental: Through the mobile application, users can search and hire several types of products. The rental process is simple and convenient.

1.4.1.2 Pre-Product Verification: Before being listed they go through a verification process and ensure that standards of quality and safety are met.

1.4.1.3 Post-Product Verification: Following the rental, products are inspected to make sure they come back in a similar shape. This ensures that the quality of products in rental stock is maintained.

1.4.1.4 User CNIC Certification: To establish trust and accountability, users need CNIC certification. Certification procedures might require identification and conformity to certain standards.

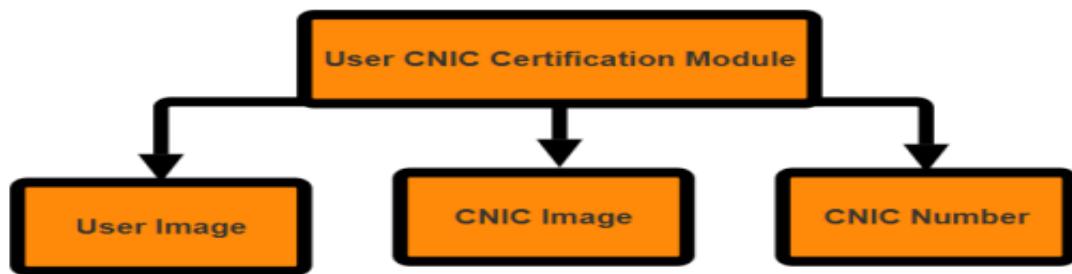
1.4.1.5 Vendor CNIC Certification: Vendors that sell rental items also go through CNIC certification to be considered credible. Certification may include checking their background and validating credentials for the business.

1.5 SCOPE OF THE PROJECT

In this sub chapter project scope involves developing a cross-platform RentMent application focusing on non-reducible products will be carried out what modules were contained inside the Rentment Application.

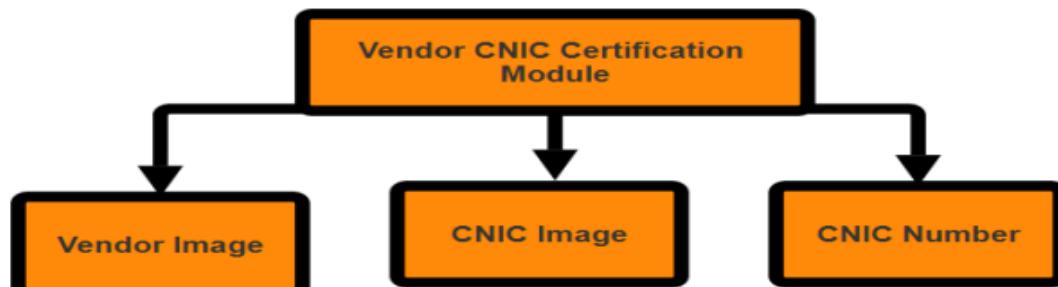
We will describe different parts used in Rentment Application:

1.5.1 User CNIC Certification Module



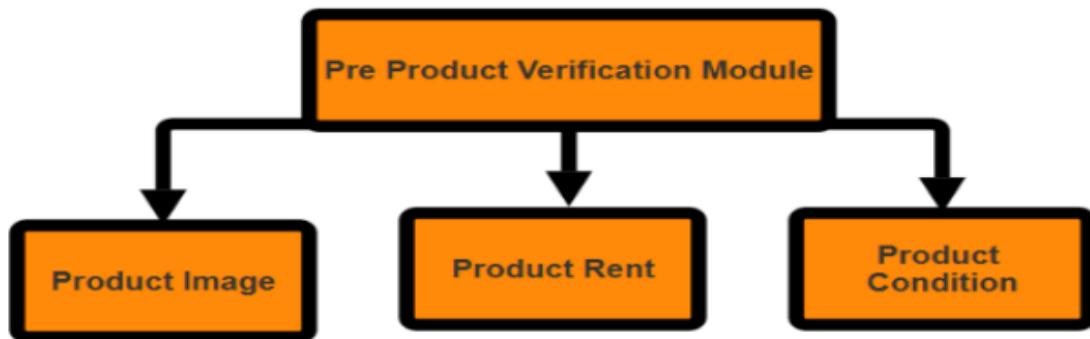
This module is used by users for the registration purpose in the application, for the registration users' needs CNIC certification for the certification user image, CNIC image and CNIC number of users is required if anyone is missing the user cannot be registered in the application because of the Security purpose of the product.

1.5.2 Vendor CNIC Certification Module



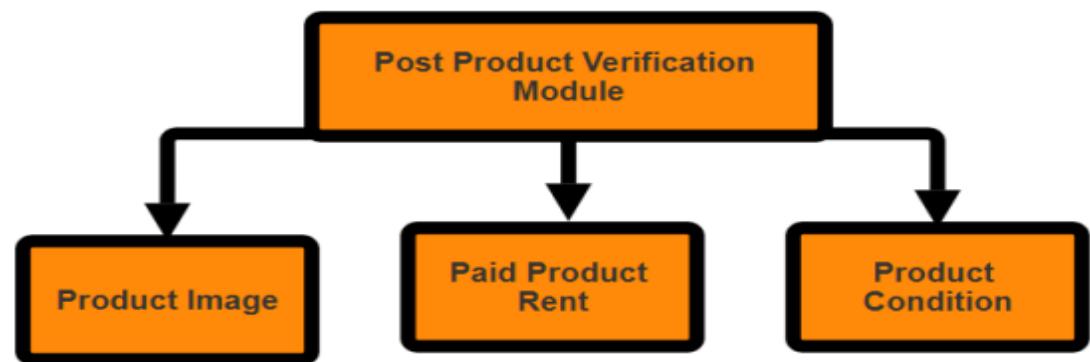
This module is used by vendors for the registration purpose in the application, for the registration vendors needs CNIC certification for the certification vendors image, CNIC image and CNIC number of users is required if anyone is missing the vendors cannot be registered in the application because of the Security purpose of the user's payment and feedback.

1.5.3 Pre-Product Verification Module



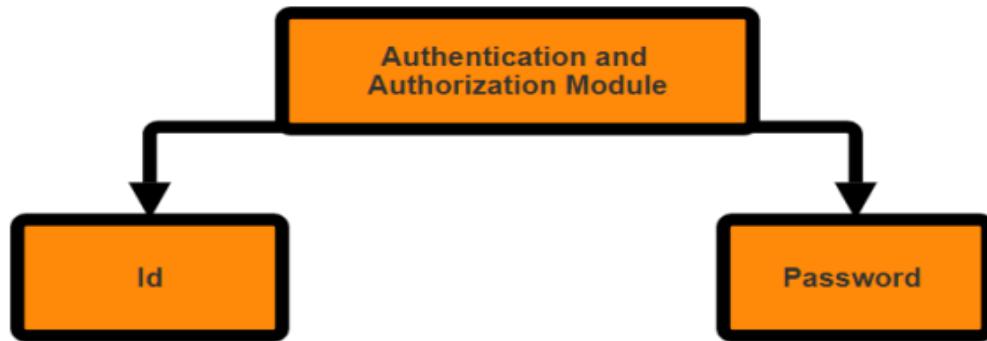
This module is used by moderator for the pre product verification, for the pre product verification moderator needs product recent image, product condition and rent of the product according to the product condition if anyone is missing the moderator cannot be Add product in the application because of the Security purpose of the user's payment and feedback.

1.5.4 Post Product Verification Module



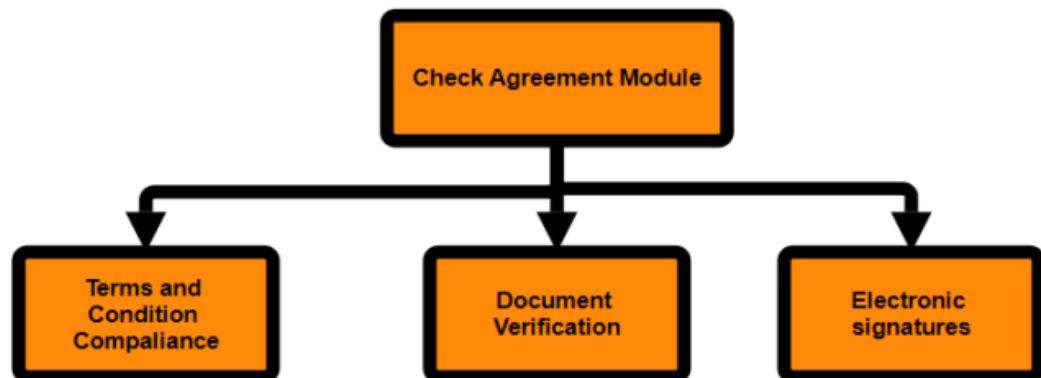
This module is used by moderator for the pre product verification, for the pre product verification moderator needs product recent image, product condition and rent of the product according to the product condition if anyone is missing the moderator cannot be Add product in the application because of the Security purpose of the user's payment and feedback.

1.5.5 Authentication and authorization module



This module is used by Admin for the Authentication and Authorization of the vendor, and moderator for the Authentication and Authorization the id and password of the vendor and moderator must be matched otherwise they cannot be login in the application.

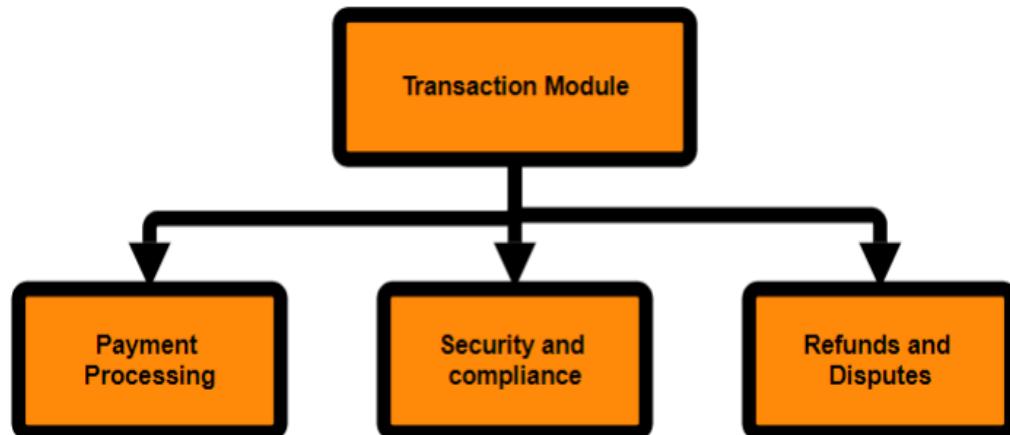
1.5.6 Check Agreement Module



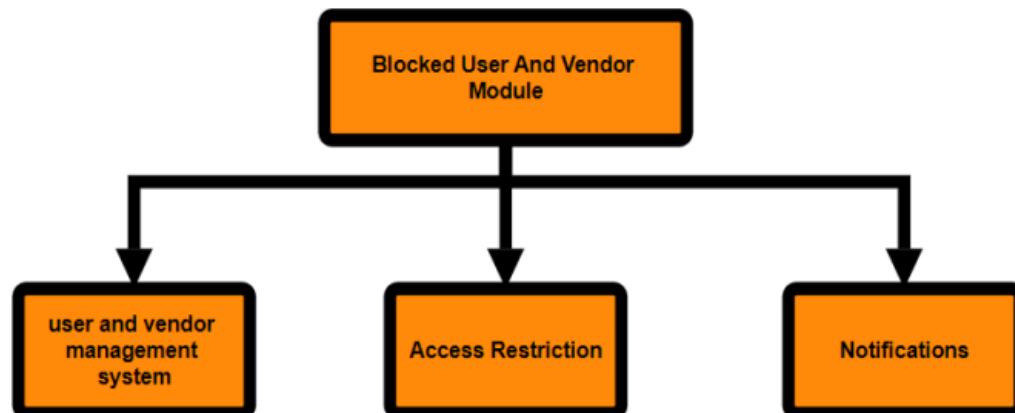
In this module moderator can check Agreement and if any problem or any of one condition could not be agree by the user or vendor the product could not be uploaded by the vendor or the product could not be giving to the user for renting purpose.

1.5.7 Transaction Module

In the transaction module the application can be applying conditions for better and secure transaction for the security purpose of both, users and vendors.

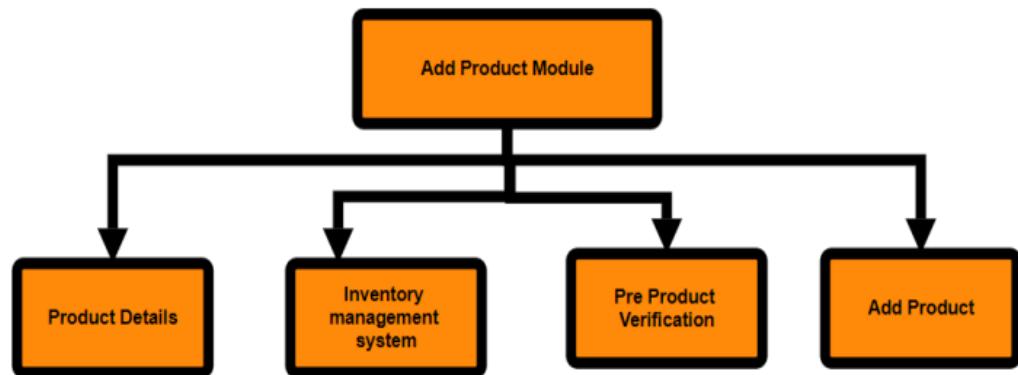


1.5.8 Blocked user and vendor module



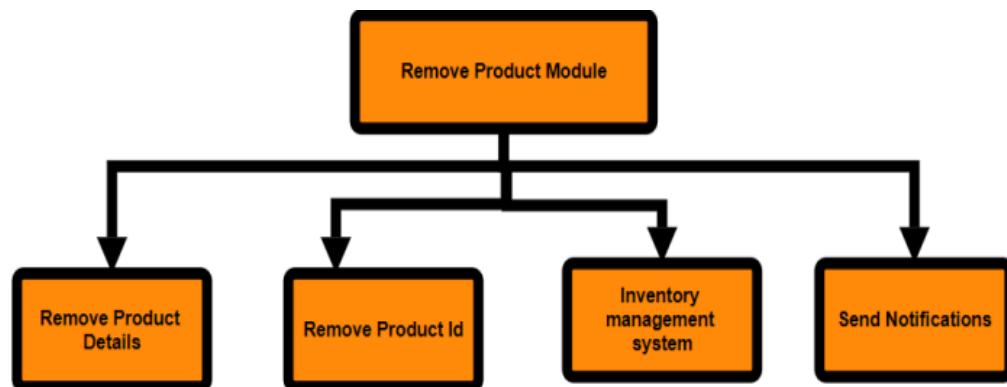
In this Module the vendor and user blocked condition can be define the details of both user and vendor is save in the database to manage them and we can give access and restriction to both according to their needs and requirements and send notifications to inform users and vendors when they have been blocked.

1.5.9 Add Product Module



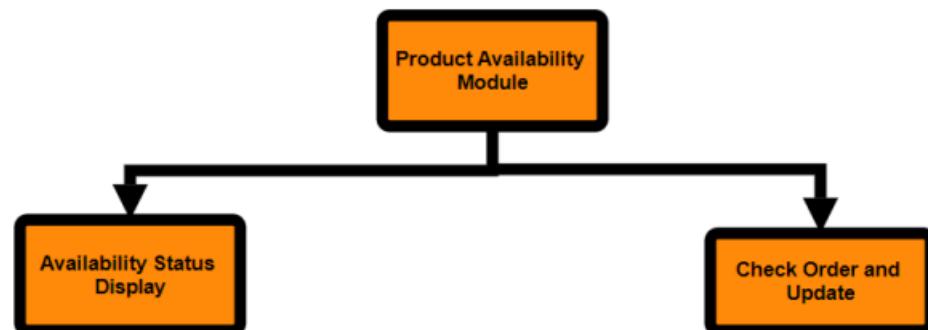
In this module Vendors add product by giving product details and the product details add in inventory and then moderator do pre product verification and then add product to application.

1.5.10 Remove Product Module



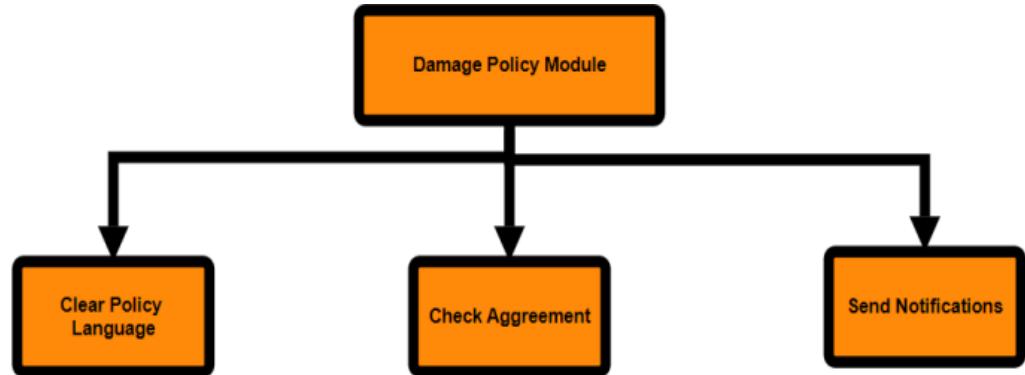
In this module Vendors Remove product by removing product details and product id from inventory and then moderator send notification to vendor that product is removed.

1.5.11 Product availability module



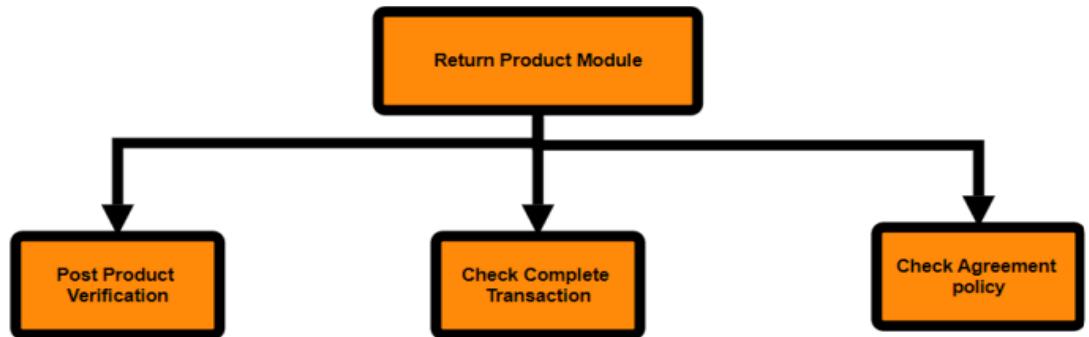
In this module the product availability status can be shown to vendor and moderator check order and update it.

1.5.12 Damage policy Module



This module is used for the product damage policy we can give a policy in clear language for the damage of the product and user can check agreement and send this notification to vendor.

1.5.13 Return Product Module



This module is used for the returning of product for this moderator can do post product verification and check the transaction details is complete or not and after returning product the vendor can check the agreement that they can receive their product well or not.

1.5.14 User / Vendor Location Module



In this module we can track the user location by using location based services and also protect user/vendor security and privacy for the better user/ vendor experience.

1.6 CONTRIBUTIONS OF THE THESIS

The contribution of the thesis for this project refers to the original and valuable insights, advancements, or improvements that the research and development efforts have brought to the field of study or the specific problem being addressed. The contributions can be categorized into different aspects:

CONTRIBUTION	DEFINITION
Practical Contribution	The thesis may contribute to solving real-world challenges in the rental industry, such as inefficient rental processes, lack of verification mechanisms, and limited availability information. By addressing these issues, the thesis offers practical solutions that enhance the rental experience for both renters and vendors.
Technological Contribution	The development of this application contributes to the advancement of technology in the rental industry by providing a secure and convenient platform for users and vendors
Methodological Contribution	The thesis may propose and implement innovative methodologies, algorithms, or frameworks for product verification, agreement management, or CNIC certification.
User-Experience Contribution	The thesis may focus on improving the user experience of the RentMent application by conducting user research. The resulting user experience improvements contribute to creating a more intuitive, efficient, and enjoyable rental platform.

Table 1.6-1 Contribution of the thesis

1.7 STRUCTURE OF THE THESIS

In the structure of a thesis, we can organize our document by using headings and subheadings to establish the hierarchy of sections. We can apply predefined heading styles to section titles, insert a table of contents for easy navigation, separate different parts of the document with section breaks, add headers and footers for consistent information. By applying consistent styles and formatting, we can create a clear and organized structure that enhances readability and comprehension of our thesis.

This thesis contains the following four chapters:

Chapter 1 – Introduction introduces the thesis, providing a summary of context, objectives and achievements.

Chapter 2 – Background introduces the background to RentMent Application. This includes a full review of existing Renting Applications and their services.

Chapter 3 – Describes the underlying architecture of RentMent Application. The design and implementation details are then described, providing an explanation of how RentMent Application was built and operates.

Chapter 4 – Conclusions Concludes the work presented in this thesis by exploring the strengths and weaknesses of the work, reviewing the extent to which the research objectives have been met, and discussing possible future work directions.

CHAPTER 2

LITERATURE REVIEW

In this chapter we present the comparison between our application and similar application existing platforms available in market. We can study all the features of existing platforms and find research gap and add this features to our application we can also find the reason behind the failure of existing platform rentable in Pakistan and working on the failure reason to make our platform successful. We can also give the background of our project and motive, objective and need of our RentMent Application.

2.1 BACKGROUND

The RentMent application project is developed to address the growing demand for a centralized platform that simplifies and streamlines the process of renting and leasing various items or services.

Traditional rental processes often involve time-consuming manual interactions, limited availability information, and fragmented communication channels, leading to inefficiencies and inconvenience for both users and vendors. With the rise of the sharing economy and the increasing popularity of renting instead of buying, there is a need for a comprehensive solution that connects users and vendors, provides transparency in rental listings, and offers a user-friendly interface for managing rental transactions.

The project takes inspiration from successful rental marketplaces and platforms that have revolutionized the way people access and utilize resources. These platforms have demonstrated the potential for creating a vibrant ecosystem where individuals and businesses can share their underutilized assets, generate additional income, and reduce waste.

The idea behind the RentMent application project is to leverage technology to bridge the gap between renters and vendors, facilitating seamless interactions, secure transactions, and efficient management of rental activities. By creating a

RENTMENT APPLICATION

User-friendly and intuitive application, the project aims to empower individuals and Businesses to rent out their assets and provide renters with a wide range of options for their specific needs.

To develop the RentMent application, the project team has adopted an agile methodology. This approach allows for iterative development, frequent feedback, and continuous improvement based on user input. By following agile practices, the team can deliver a high-quality, user-centric application that evolves with the changing demands of the rental market.

The RentMent application project aims to revolutionize the rental industry by providing a reliable, efficient, and user-friendly platform that facilitates seamless interactions between users and vendors. By simplifying the rental process, the project aims to promote resource sharing, reduce waste, and contribute to the growth of the sharing economy.

2.1.1 Why we use MERN STACK for RentMent Application?

The MERN stack, consisting of MongoDB, Express.js, React.js, and Node.js, is ideal for the RentMent application due to its full-stack JavaScript environment, which allows for seamless development across both the front-end and back-end. MongoDB, a NoSQL database, is highly flexible and scalable, making it perfect for handling the diverse data requirements of a rental application, such as storing user profiles, product listings, and transaction histories. Express.js simplifies the creation of robust APIs that facilitate communication between the database and the front-end, while Node.js provides a fast, event-driven server environment that supports real-time updates, which is essential for booking and rental notifications. React.js ensures a responsive, dynamic user interface for both users and vendors, allowing them to manage their rentals, view product verifications, and interact with the platform efficiently. The MERN stack's ability to deliver high-performance, scalable web applications with efficient data handling and real-time capabilities makes it a strong choice for the RentMent application.

2.2 RELATED WORK

In our Literature review we can define the features of different existing rental platforms and find research gap in this platforms and find the reason behind their failure and success and make our application according to the features find in the research gaps to make our application success full and user friendly and popular and secure.

The different platforms is given below:

2.2.1 RENTABLE.PK

Existing platform that gives product on rent in Pakistan:

<https://rentable.pk/>

Rentable has rental ads available in Pakistan of goods from cars, furniture, and electronics to every other rentable service listing.

2.2.1.1 RESEARCH GAP:

1. The Existing application cannot give the online transaction Option.
2. The Existing application not given the option for the different rent according to the product or the user or vendor choice.
3. The Existing application cannot give the Option of product availability.
4. The Existing application cannot give user verification options. The Existing application cannot Offering Product damage policy.
5. The Existing application cannot verify the product before and after giving it on rent.
6. The Existing application have No vendor detail or verification option is given.
7. The website does not provide an automated option for the calculation of the amount for the rent based on the number of days provided by the customer.

2.2.2 YOODLIZE.COM:

Existing platform that gives product on rent in United States:

<https://www.yoodlize.com/>

Yoodlize is the peer-to-peer app that lets you rent anything from anyone and rent out your own stuff too. Browse a variety of listings in your area like paddle boards, bounce houses, e-bikes, trailers, and just about anything else.

2.2.2.1 RESEARCH GAP:

1. In this app a policy is the item must be cleaned before returned.
2. Users receive real-time notifications about new messages, booking requests, and other important events.
3. Users can request items or services, and owners can accept or decline these requests.
4. Users can communicate with each other within the platform to discuss rental details.
5. Users receive notifications about messages, booking requests, and other activities related to their listings or inquiries.
6. Users can search and browse through the available listings based on criteria like location, category, price, and date.
7. Some amount should be paid before renting the product.

2.2.3 FATLLAMA.COM:

Existing platform that gives product on rent in United States:

"<https://fatllama.com/>"

Fat Llama is a platform that allows individuals to rent out their items to others and also rent items from people in their local area.

2.2.3.1 RESEARCH GAP:

1. Users can list their items for rent on the platform by providing details and photos of the items. Those looking to rent can browse the available listings and search for specific items they need.
2. Fat Llama offers insurance coverage to protect both renters and owners in case of damage or theft of rented items.
3. To help operate the Fat Llama platform, we charge a service fee on all transactions of 25% from the owner and 25% from the renter.
4. It depends on your location, you can rent items locally or, in some cases, items can be shipped nationwide.

2.2.4 RENTANYTHING:

Existing platform that gives product on rent in India:

"[HTTPS://WWW.RENTANYTHING.IO/](https://www.rentanything.io/)"

RentAnything is a web application where you can rent different types of products. Its mobile application option is coming soon.

2.2.4.1 RESEARCH GAP:

1. The platform should have a user-friendly search and filtering system.
2. An admin dashboard is necessary to manage users, listings, and resolve disputes if they arise.
3. The platform should support secure payment transactions, including options for credit cards, PayPal, or other payment gateways.
4. This platform gives products availability option and how many products are available at this time.
5. This platform gives Return date and Booking Date Option.
6. This platform informs you how many items are to be rented.
7. This platform gives you a renter's policy if you can agree you can rent a product.
8. This platform gives you a Product location.

RENTMENT APPLICATION

Application Features	Rentable	Yoodlize	Fat Llama	RentAnything	RentMent
User verification	NO	YES	YES	NO	YES
Product Verification	NO	YES	YES	NO	YES
Vendors detail	NO	YES	NO	NO	YES
Damage policy	NO	YES	YES	YES	YES
Transaction option	NO	YES	YES	NO	NO
Product Availability	NO	NO	NO	YES	YES

Table 02-1 Comparision Table

After doing the above literature review and find research gaps in different applications we can find out the Fatllama is the most Popular and secure application with the lots of available product in their application and lots of products on rent so we can apply the features of Fatllama to make our application success full in Pakistan.

The application is available in Pakistan is rentable so as shown in graph rentable is less popular and failed because they cannot give the features like Fatllama and Yoodlize to make our application successful we can add the features of other existing platform available in different region.

All the platforms is famous and improve the economy of their country but rentable is not even 5% famous and cannot give anything to economy of Pakistan because they cannot do online transaction this application is failed in 2015 -2016 because of their poor performance.

RENTMENT APPLICATION

So we can make application who can contribute in the economy of our country and trying to make our economy better.

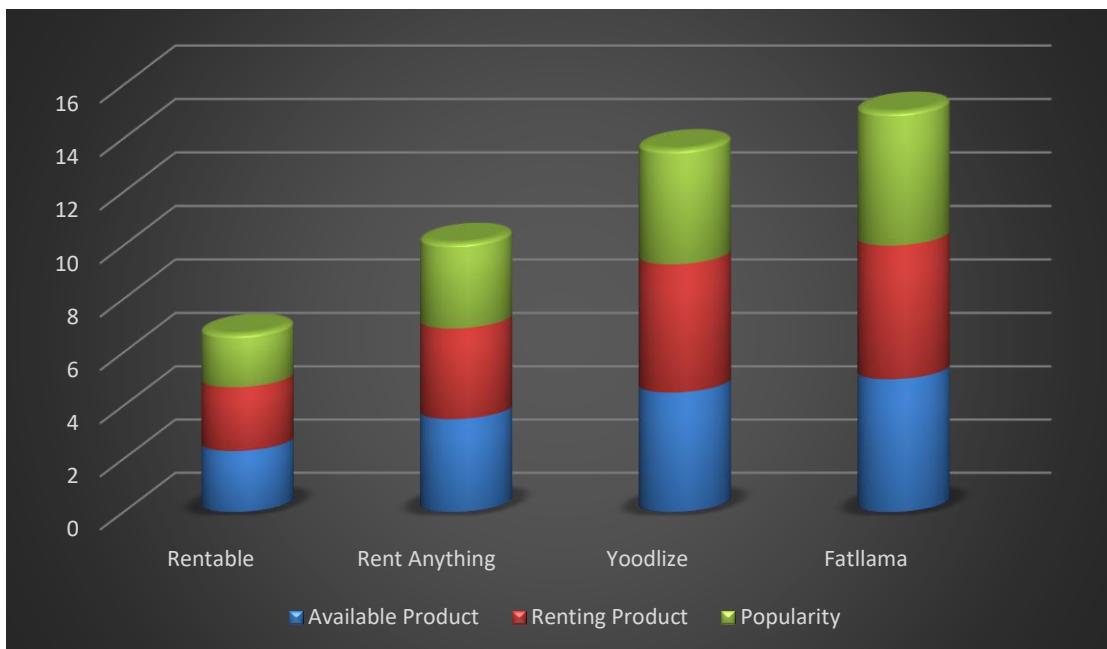


Figure 2-1 Comparative Chart

2.3 COMPARATIVE STUDY

For the comparative study we can compare the features available in different application and best features of the thesis added to our application, the main difference between our and other application is that we provide a platform for people willing to give their product on rent our application will provide security of product by their pre and post product verification techniques and we can provide a one stop for all the categories.

The Rentable application who is current renting app in Pakistan is failed due to lack of security and trust issue for user and vendor and the features mentioned in above table cannot be available in this application that's why the application is failed, the other failure reason of this application is that they cannot do marketing of their application and they only made web not available, so in this era customer prefer mobile application more instead of web application. Application and mobile application is So we can do marketing of our application more , and as we are making a cross platform application who

RENTMENT APPLICATION

is available on both android and IOS , and we can give security of product and added above mentioned product features in table so our application can be more successful and secure and user friendly.

2.4 SUMMARY

This chapter discusses the background and objectives of the RentMent application, which aims to provide a centralized platform for renting and leasing various items or services. The text emphasizes the need for a user-friendly and efficient rental process, inspired by successful rental marketplaces in the sharing economy.

This chapter also includes a literature review of existing rental platforms such as Rentable.pk, Yoodlize.com, Fat Llama, and RentAnything.io. The review identifies the features and gaps in these platforms, highlighting the importance of online transactions, product availability, user and product verification, vendor details, damage policies, and secure transactions.

A comparative study is conducted to compare the features of different rental platforms, including Rentable. The study reveals that Rentable lacked many essential features and failed due to security and trust issues, as well as a lack of marketing efforts and a limited mobile application presence.

Based on the findings, the RentMent application aims to address the shortcomings of existing platforms by incorporating comprehensive features, ensuring security, and providing a user-friendly experience. The application intends to promote resource sharing, reduce waste, and contribute to the growth of the sharing economy

CHAPTER 3

SYSTEM DESIGN

In this chapter we can find the different requirements of RentMent Application and research and System methodology of RentMent Application by applying this we can make our project.

This chapter also present some diagrams like ERD, Use case, Sequence and Flow Chart diagram by the help of this diagram we can easily understand the architecture of our project.

3.1 SYSTEM METHODOLOGY

The RentMent application followed the Agile methodology to ensure flexibility, adaptability, and continuous improvement throughout the development process. The Agile methodology involves iterative development cycles and close collaboration between the project team and stakeholders.

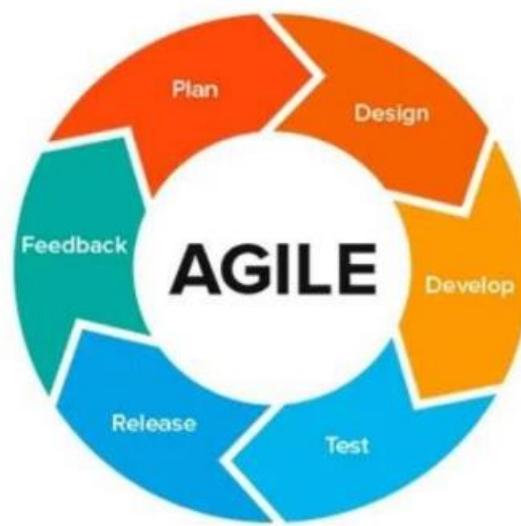


Figure 3-1 Agile Methodology

By following the agile methodology, the RentMent Application project fostered a collaborative and iterative approach, allowing for flexibility, rapid development, and continuous improvement based on user feedback and changing market need.

The key aspects of the agile methodology in this project include



Flexibility and Adaptability: Agile methodologies are designed to be flexible and adaptable to changes in project requirements.



Customer Satisfaction: Agile focuses on delivering value to the customer early and continuously throughout the development process.



Iterative Development: Agile promotes iterative development, where small, incremental improvements are made to the product in short cycles



Risk Management: Agile methodologies help in managing project risks by breaking down the project into smaller, manageable iterations.



Continuous Improvement: Agile encourages a culture of continuous improvement.

Table 3-1 Key Aspects of Agile

3.1.1 SOFTWARE REQUIREMENTS

Project Constraints	Software Required
Operating System	Windows, IOS
Documentation	Microsoft Word, Excel
Version Control System	GitHub
Project Management	Jira
Application Diagrams	Smart Draw
Application Architecture	Figma, Canva
Front End Development	Android Studio, Expo
APIS Integration	
Back End Development	Android Studio
Database	Firebase
Testing	Selenium
Deployment	Docker

Table 3-2 Software Requirements Table

This table can be define us the software required in our whole software from beggining to end.

3.1.2 HARDWARE REQUIREMENTS

Server Infrastructure: Sufficient hardware resources (servers, storage, networking) to host the rental application and handle user traffic.

Database Server: A robust database server to store and manage application data efficiently.

Client Devices: The rental application should be accessible from a range of client devices, including smartphones, tablets, and computers, with compatibility across different operating systems.

3.1.3 SYSTEM CONSTRAINTS

The system constraint refers to that required when using the Application either it is the specific platform which the Application run or the system which is required to run the application. If the system Constraint are not meet they affect the many factor of the RentMent application various aspects of the system, including performance, security, scalability, and compatibility.

Here are some common system constraints for a rental application:

Required Hardware: The RentMent application must perform their operation within the constraints of the hardware it runs on, including processing power, memory, storage capacity, and network bandwidth.

Platform compatibility: The application may be constrained by compatibility with specific operating systems. It must be designed and tested to run on the targeted operating systems and versions.

Required Network Bandwidth: The application may need to operate under specific network conditions, such as bandwidth limitations, latency constraints, and the ability to handle intermittent connectivity.

Require CIA of Data: Security constraints include the need for data encryption, secure user authentication, and authorization mechanisms to protect sensitive information.

RENTMENT APPLICATION

Performance Requirements: The application must meet certain performance standards, including response time, throughput, and resource

UI Constraints: Design and usability constraints may include adherence to specific user interface guidelines, accessibility requirements, and compatibility with different devices and screen sizes.

Data Storage Constraints: Limitations on data storage capacity, database performance, and data retention policies may impact the design and functionality of the rental application

3.1.3 EXTERNAL INTERFACE REQUIREMENTS

3.1.3.1 Hardware Interfaces

The hardware interfaces of a Rentment Application refer to the physical connections and requirements related to the hardware components that the application interacts with. In modern web and mobile applications, hardware interfaces are often abstracted and standardized by the operating systems and platforms. However, it's still important to consider the hardware-related aspects that may impact the performance and functionality of the application.

Here are some considerations for hardware interfaces:

Client Devices: Define the minimum hardware specifications for devices used by users (renters and property owners). Consider compatibility with various device types, including desktops, laptops, tablets, and smartphones.

Cameras: Camera are required when the pre and post product verification are captured Payment Processing Hardware: If the application supports in person transactions (e.g., property viewings with on-site payments), specify any hardware requirements for point-of-sale (POS) devices.

Networking Equipment: Specify the hardware requirements for networking equipment, including routers, switches, and firewalls.

3.1.3.2 Software Interfaces

Software interfaces for a Rentment Application refer to the points of interaction between the application and other software components, systems, or services. These interfaces ensure seamless communication and data exchange.

Here are some key software interfaces to consider for a Rentment Application:

Mobile Device Interfaces: Specify the compatibility and supported versions of mobile operating systems (e.g., iOS, Android) for the Rentment mobile application. Define the minimum required specifications for mobile devices (e.g., screen size, resolution).

Application Programming Interfaces (APIs): Define APIs for communication between different components of the Rentment Application. Document the request and response formats, authentication mechanisms, and endpoints. Specify any third-party APIs integrated into the application.

Database Interfaces: Specify the type of database system used (e.g., MySQL, PostgreSQL, and MongoDB) and outline the database schema. Define data access and manipulation methods.

External Service Interfaces: Document interfaces with external services, such as payment gateways, mapping services, or social media platforms. Specify the communication protocols and data formats use.

1. **Authentication Interfaces:** Specify the authentication mechanisms used for user access. Document how the application interfaces with identity providers if using third-party authentication services.
2. **Payment Gateway Interfaces:** If the application involves financial transactions, specify the interfaces with payment gateways. Document supported payment methods and currencies.
3. **Communications Interfaces:** Communication interfaces in a Rentment Application involve the protocols, methods, and mechanisms used for communication between different components of the system, as well as between the application and external entities.

Communication interfaces

1. **Restful API:** Define a Restful API for communication between different components of the Rentment Application. Specify resource endpoints, request methods, and response formats.
2. **Graph QL:** If applicable, define a Graph QL API with appropriate query and mutation endpoints.
3. **Database Connectivity:** Specify the communication protocols and methods for interactions between the application and the database. Define how the application retrieves, updates, and manipulates data in the database.
4. **Restful APIs:** Define communication interfaces with external services, such as payment gateways, mapping services, or social media platforms. Document authentication mechanisms and data formats.
5. **Authentication Communication:** If using third-party authentication services, define the communication interfaces with OAuth or OpenID Connect. Specify how the application exchanges authentication token.

3.2 FUNCTIONAL REQUIREMENTS

Login to User

The first step of the functional requirement is Login step. In this function user login with user name and password to authorization. All users need to make authorized to log in their account.

Customer Registration

The customer give their credentials to take rental services from the Rentment Application if they verified. With the help of that functionality user can easily perform different operation Application

1. Customer Name
2. Customer Email
3. Customer NIC
4. Customer Password

RENTMENT APPLICATION

5. Customer city
6. Customer Mobile no

Vendor Registration

The vendor give their credentials to provide rental services from the Rentment Application to customer with their specify product if they verified. With the help of that functionality vendor can easily perform different operation Application the Vendor Registration include:

1. Vendor Name
2. Vendor Email
3. Vendor NIC
4. Vendor Password
5. Vendor city
6. Vendor Mobile no

Vendor Product category

Vendor Provide Product After Enter the vendors the all their credential so they proceed into their dashboard/Homepage vendors will provide their product filling add product form by providing sufficient detail about product such as:

1. Product id
2. Available Date
3. Cost according to duration

It consist of following sub Functionalities:

- i. **Add Product:** Vendor can add product according to category which it depend on.
- ii. **Update Product:** Vendor can update the existing product details such as rent available dates etc. by entering the product id of that product.
- iii. **Delete Product:** Vendor can delete the product by his own choice.

Database Maintenance

The application Entity (customer or the vendor) such as product detail, personal detail, etc. and data provided by the customer such us booking detail their credential maintained in a data base by the application administrative.

Finding and Booking the Products

The customer after accessing the application he searches the product if he they find the required product then he need to fill the booking form and submit to the database

Verify Booking and Intimate the vendors

The administrator after Login to the site they verifies the recent bookings from customer Then he need to forward all bookings detail to vendor.

Authentication

Authentication is nothing but providing the security to the system here every one must enter their valid credentials into the system throw the login page. The login page will restrict unauthorized users. A users must provide the credentials like user id and password for log into the system otherwise the request will throw back.

Accept Damage policy

Agreement Vendors have authority generate before they provide their product rent to the customers they create Damage policy agreement between them and the customers if there is any harm to the product.

Log out

After using all operation user need to sign out to close the session.

3.2.1 USE CASES

1. Use case diagram for user and vendor login

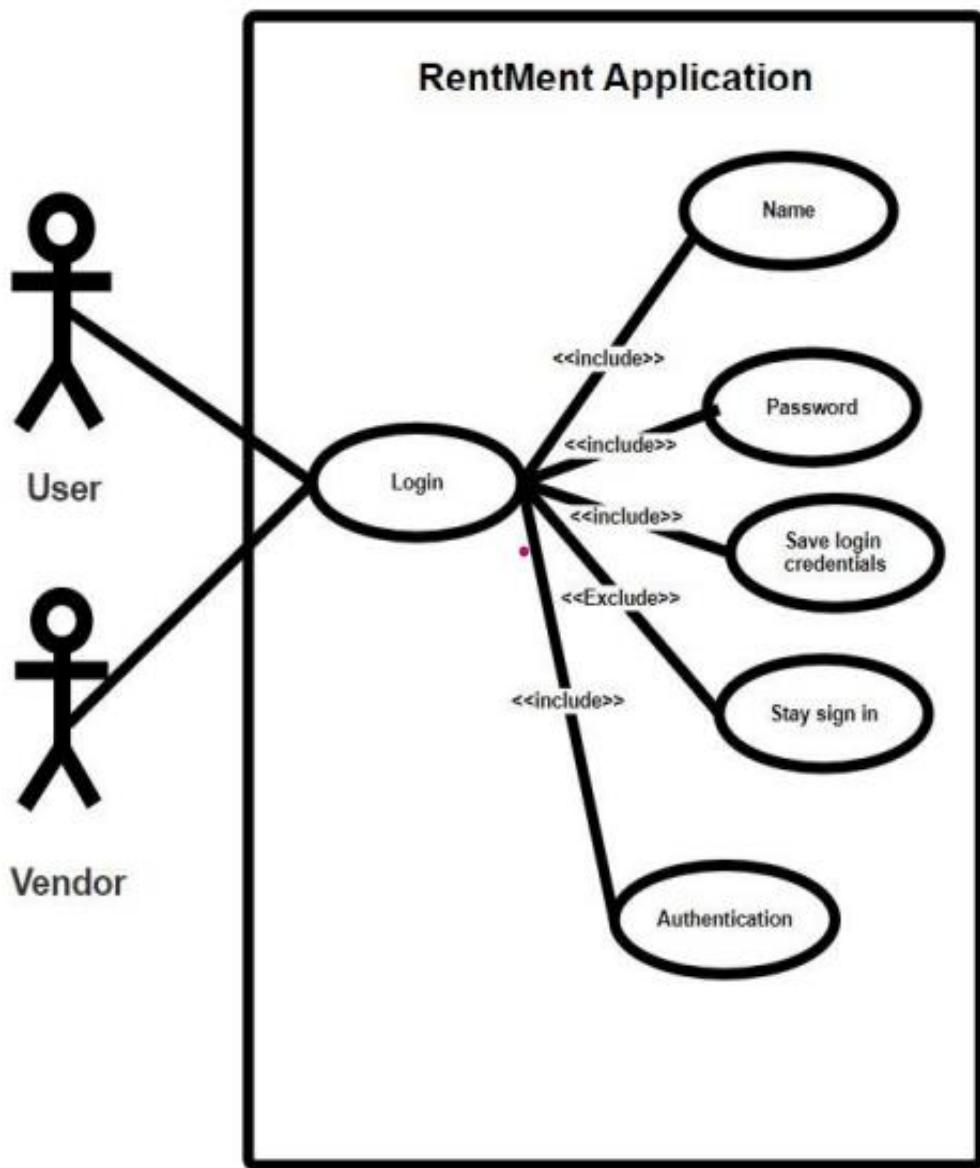


Figure 3-2 User case Diagram for user and vendor login

AS shown in the figure 3-1 the function and role of the vendor and user to login is define the following options.

- I. Enter the username associated with the vendor account and user accounts.
- II. Input the password for the vendor or user account.
- III. If credentials are correct, access to the vendor/User dashboard is granted.

2. Use case diagram for user and vendor registration

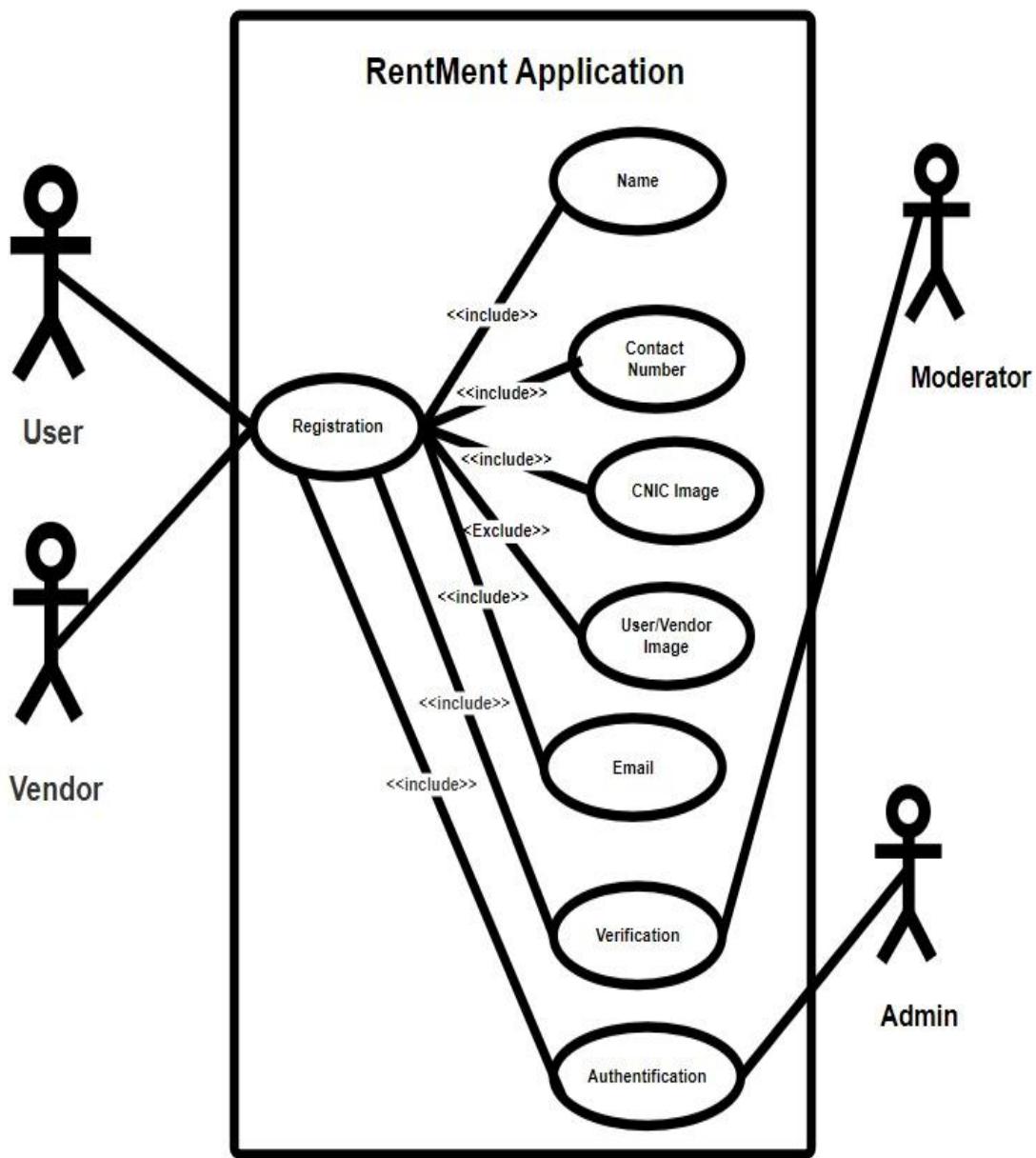


Figure 3-3 Use Case Diagram for user and vendor Registration

RENTMENT APPLICATION

AS shown in the above figure 3-2 the function and role of the vendor and user to register their accounts is define the following options.

Vendor

1. Access the vendor registration page.
2. Fill out the required information such as company name, contact details, etc.
3. Choose a username for the vendor account.
4. Create a strong password for the account.
5. Agree to terms and conditions,
6. Click on the "Register" or "Sign Up" button.

User

1. Navigate to the user registration page.
2. Fill in the necessary personal information like name, email address, etc.
3. Choose a unique username for the user account.
4. Create a secure password for the account.
5. Agree to terms of service and privacy policy.
6. Click on the "Register" or "Sign Up" button.

3. Use Case Diagram For Admin

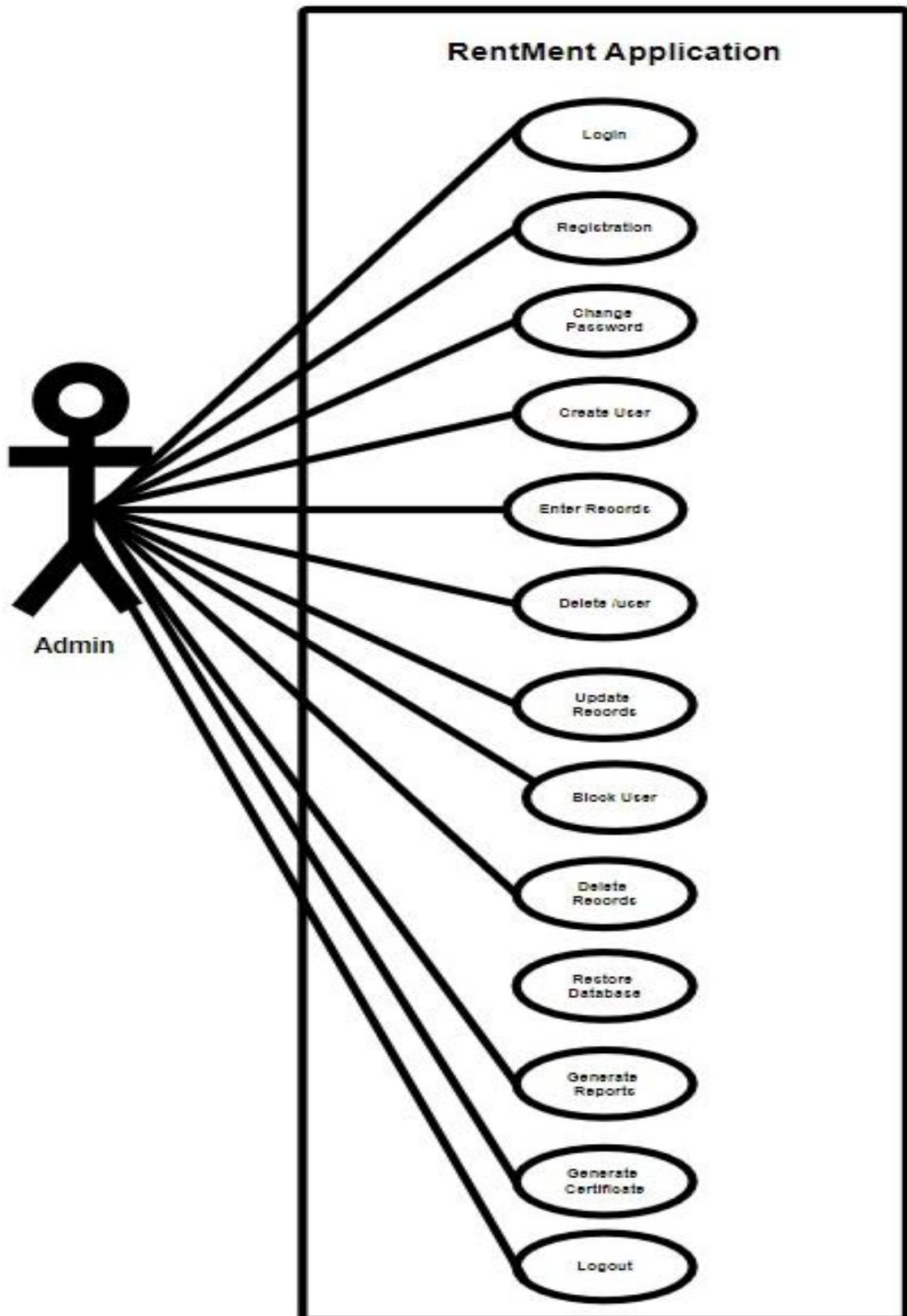


Figure 3-4 Use case diagram for Admin

RENTMENT APPLICATION

AS shown in the above figure 3-3 the function and role of the admin in our project is define the admin can control the following options.

1. Login session until the user logout
2. Editing of any account
3. Can view delete and update any products
4. Can delete any scam/ fake account
5. Can view the customer feedback

4. Use case diagram for RentMent application

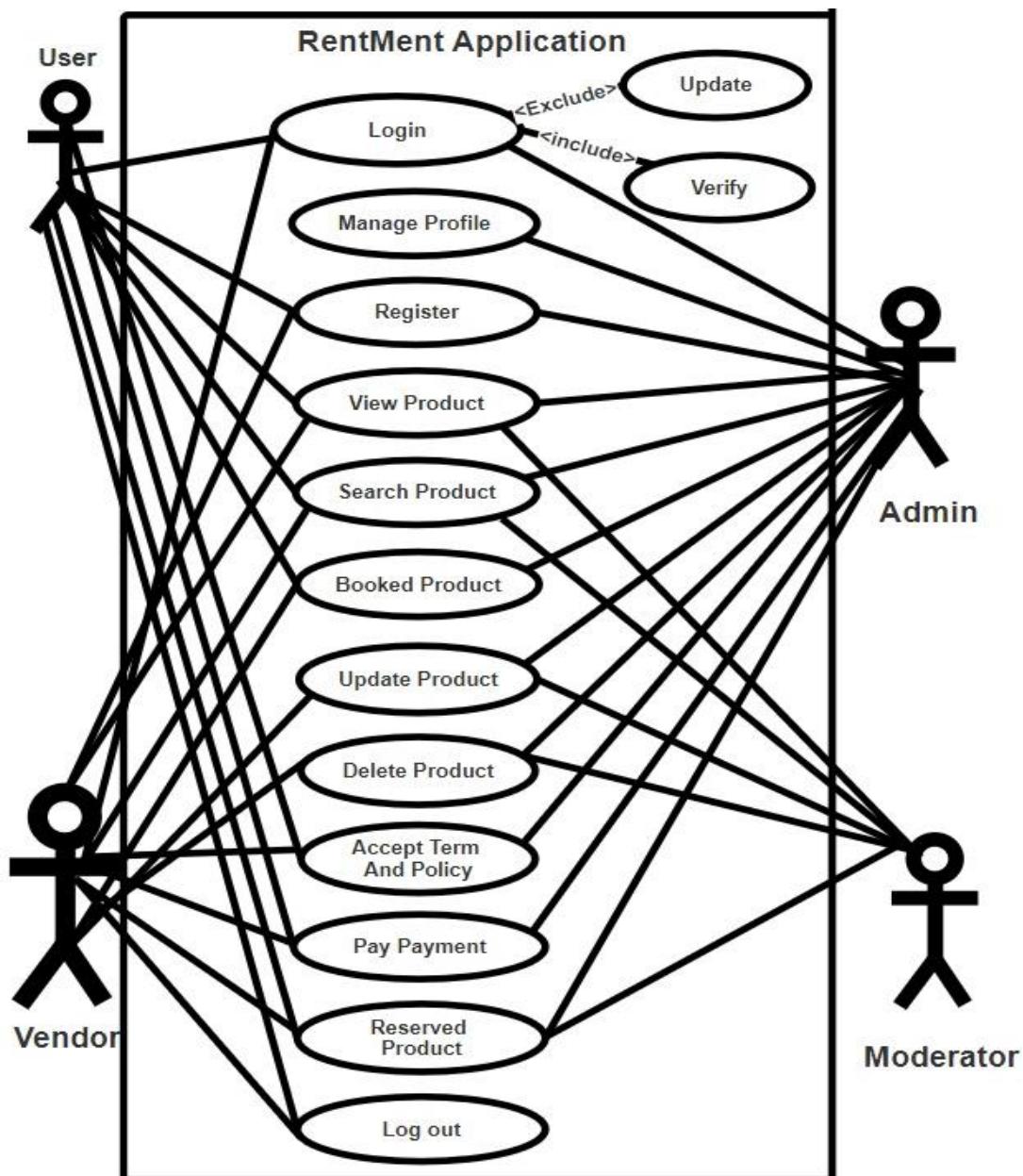


Figure 3-5 Use Case Diagram for RentMent Application

RENTMENT APPLICATION

The above figure 3-4 show the RentMent application process of the Admin, Vendor, User and Moderator define each role bellow:

Admin

1. Login session until the user logout
2. Editing of any account
3. Can view delete and update any products
4. Can delete any scam/ fake account
5. Can view the customer feedback

Moderator

1. Can View product
2. Can Search products
3. Can update products
4. Can delete products
5. Reserved products

User

1. User Register and login
2. Can view and Edit their accounts
3. Can search various products
4. Can make and cancel products
5. Can delete their accounts
6. Can give feedback

Vendor

1. Vendor Register and login
2. Can view and Edit their accounts
3. Can upload various products
4. Can delete the products
5. Can delete their accounts
6. Can give feedback

3.2.2 ACTIVITY DIAGRAM

1. Activity diagram for login

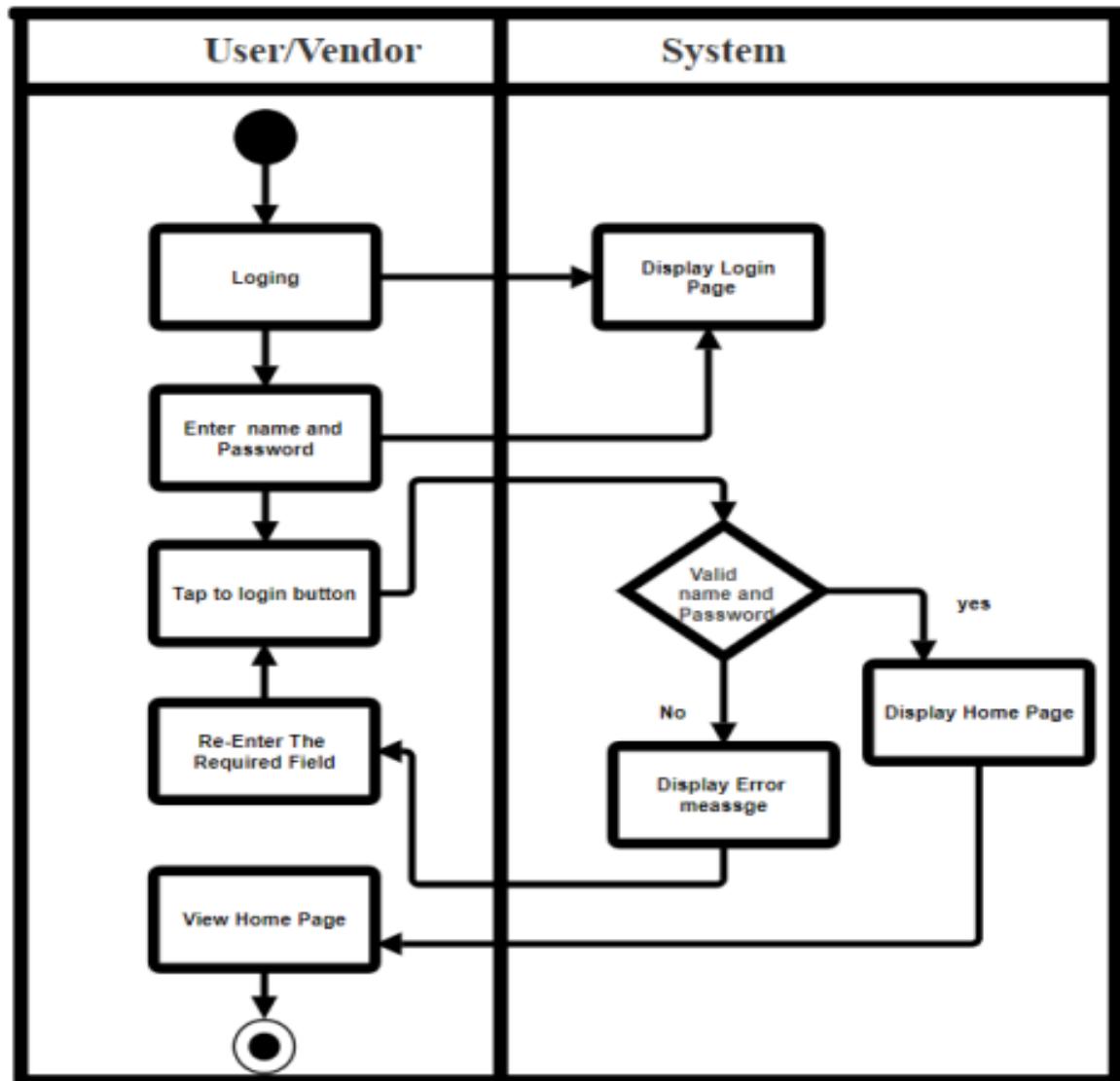


Figure 3-6 Activity Diagram For user and vendor Login

The figure 3-5 define as the role of login in the RentMent Application

1. View the user interface
2. Can create accounts
3. Allow the user and vendors to send the feedback
4. User can book the products
5. Vendor can upload delete and view their products

2. Activity Diagram For Registration

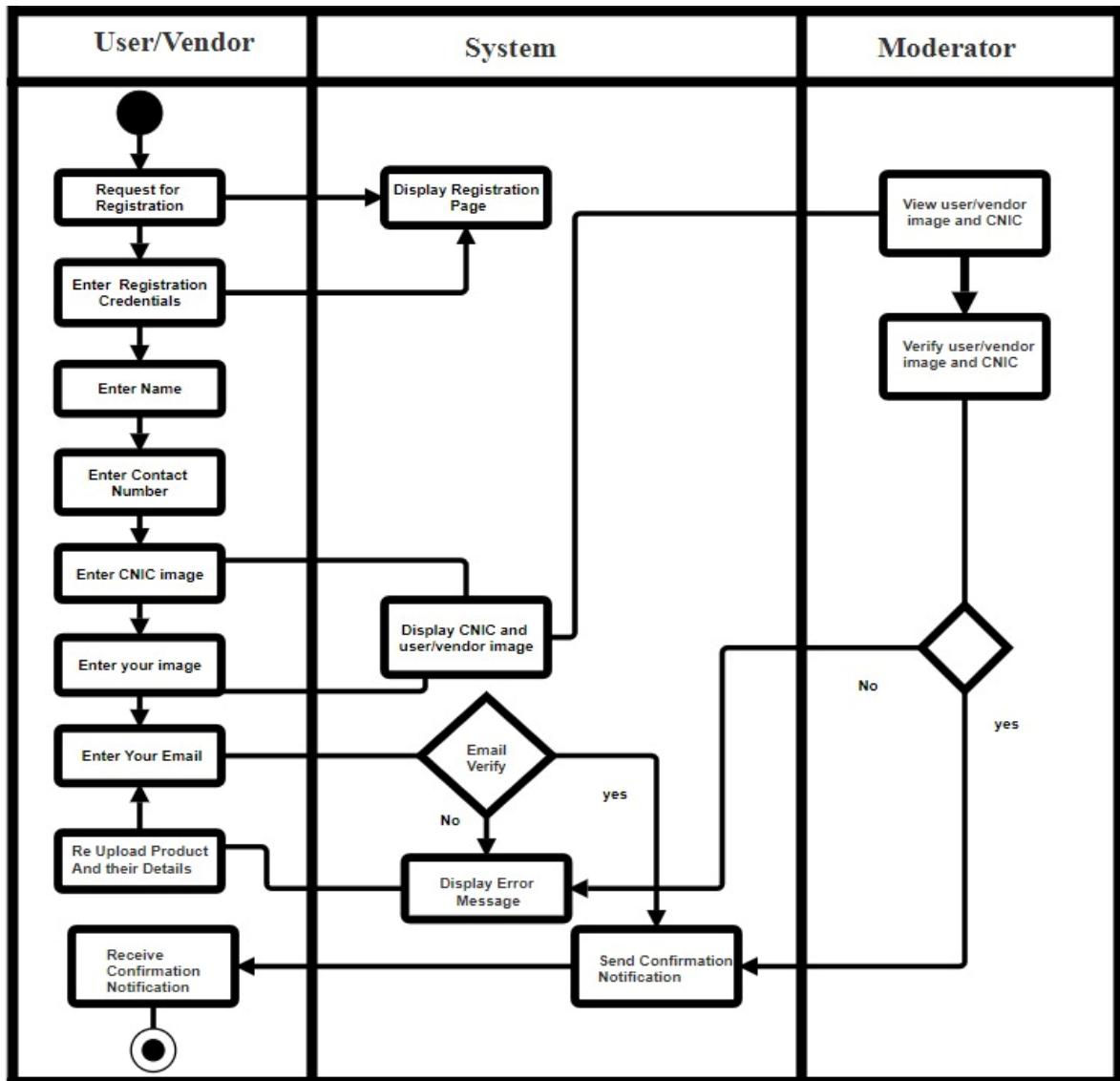


Figure 3-7 Activity Diagram For user and vendor Registration

In the figure 3-6 we have activity diagram for registration the vendor and user can register themselves in our RentMent application

1. Enter registration credentials like name, email, and CNIC picture and user image.
2. Verify the CNIC picture with the user photo
3. Send verification Email
4. Show the display screen

3. Activity diagram for upload product

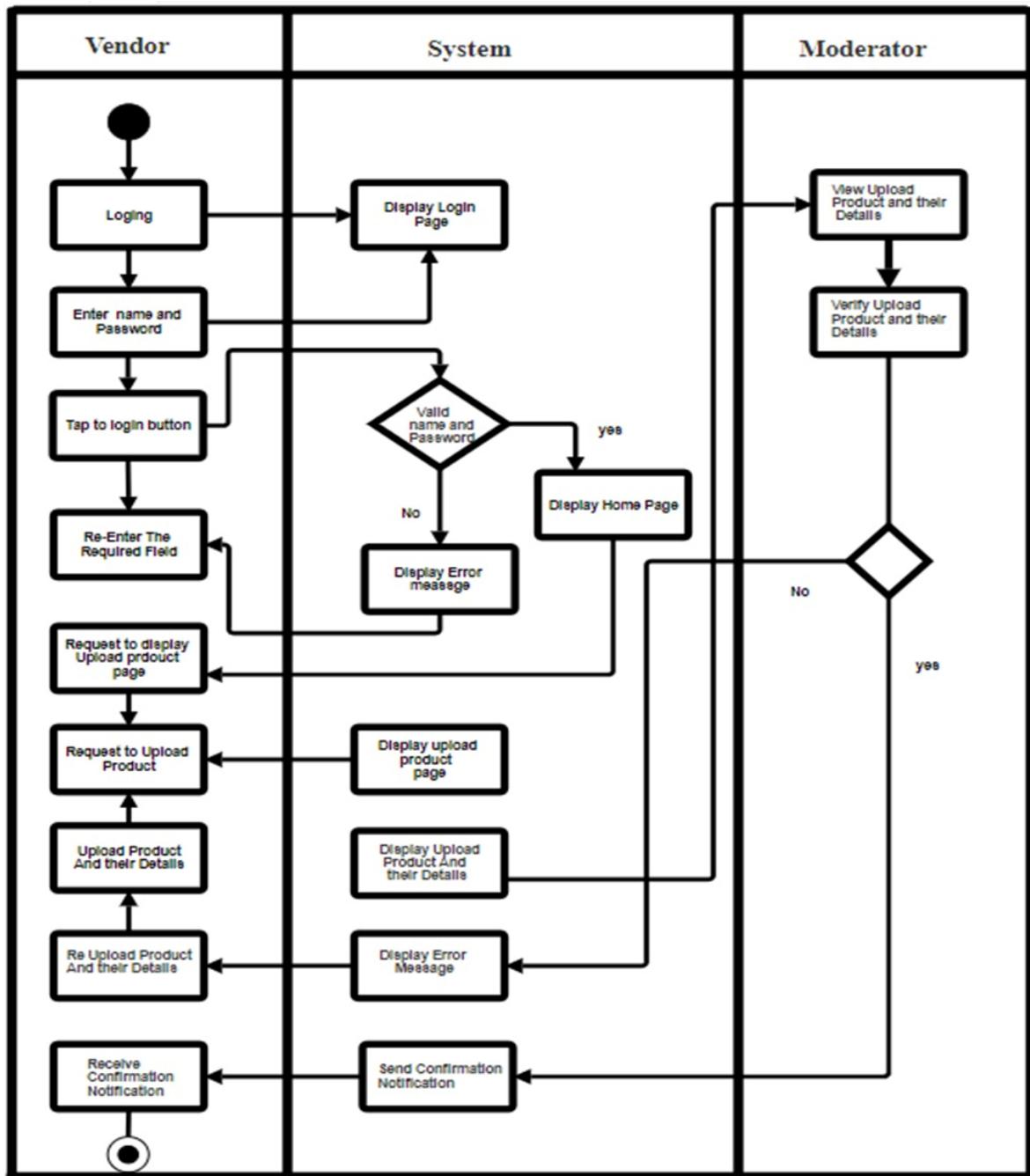


Figure 3-8 Activity Diagram For upload product

In the figure 3-8 the functionality to upload the products by the vendors

4. Login to display the screen
5. Request to upload the products
6. Upload the products picture
7. Write the details of the products

3.2.3 SEQUENCE DIAGRAM

1. Sequence diagram for rental sequence

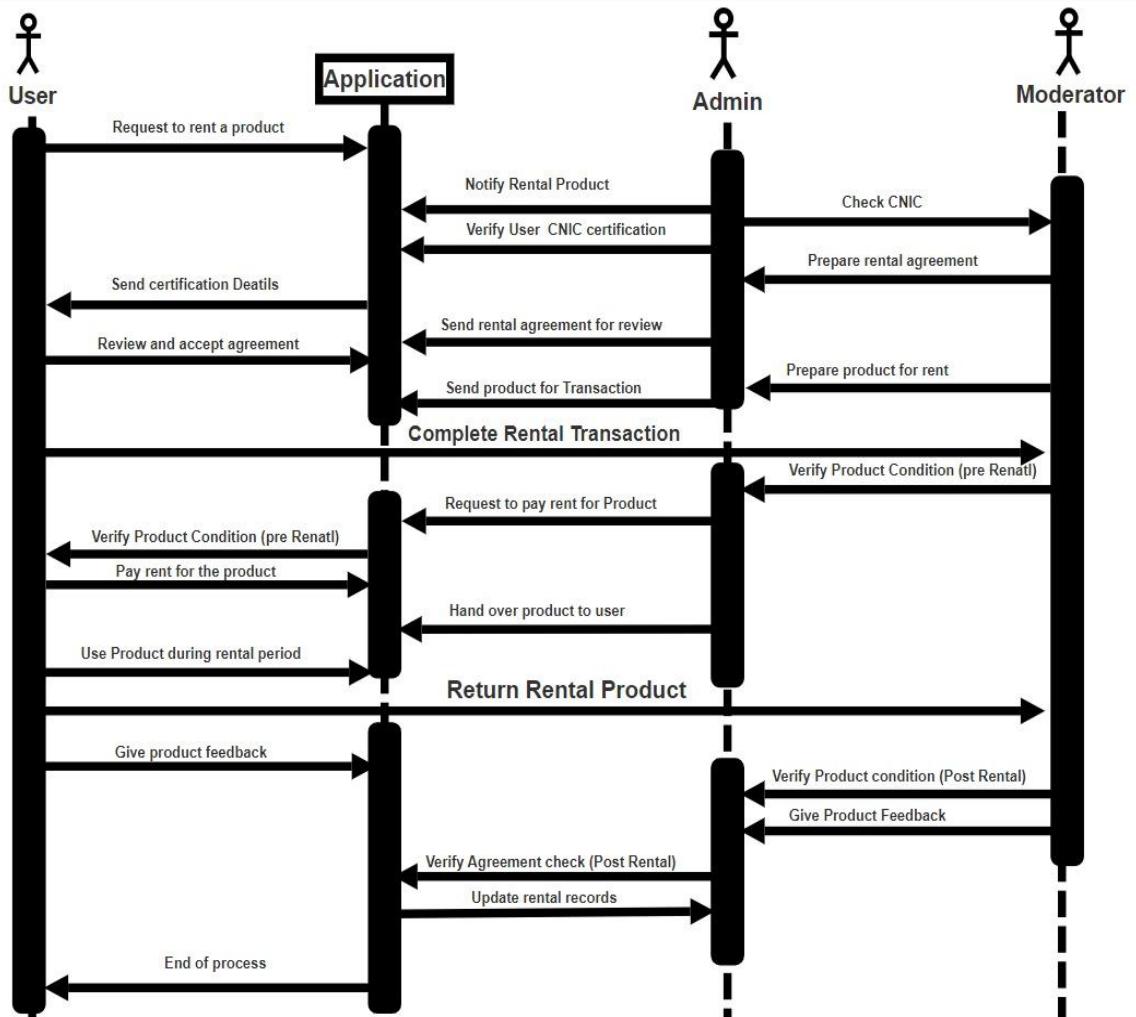


Figure 3-9 Rental Sequence Diagram

The above figure 3-8 define the role and function of the user and vendor to define the mechanism

1. The User Request to the Rent the Produce Sequence diagram in first sequence
2. The vendor want provide the Rental product admin accept the request
3. The vendor Set Rental Product Detail and the Admin saved the Rental product detail
4. The Moderator verify the User (vendor) set the Term and Condition

5. Sequence diagram for add products

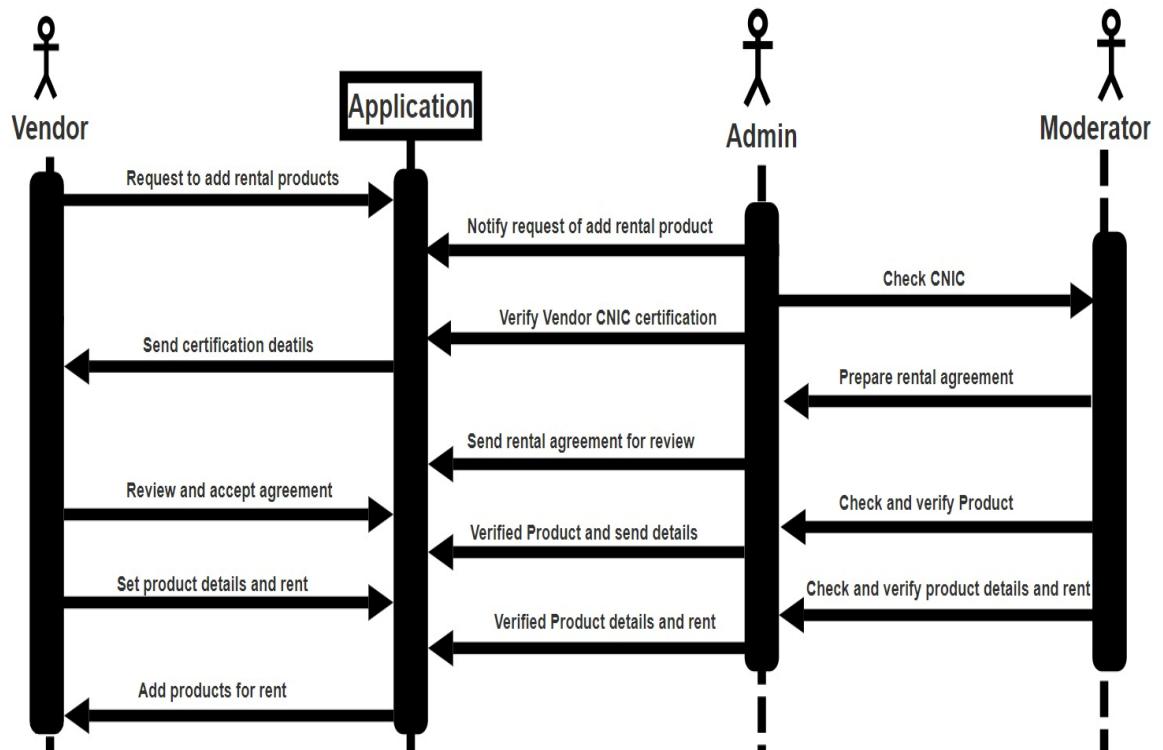


Figure 3-10 Add Product Sequence Diagram

The figure 3-9 shows the vendor to Add the products in a sequence form is define as:

1. The vendor request to the rental application to give access and add the products
2. The Application will check the vendors agreement
3. Then will verify the products category
4. Add the products for rent

3.2.4 ER-DIAGRAM

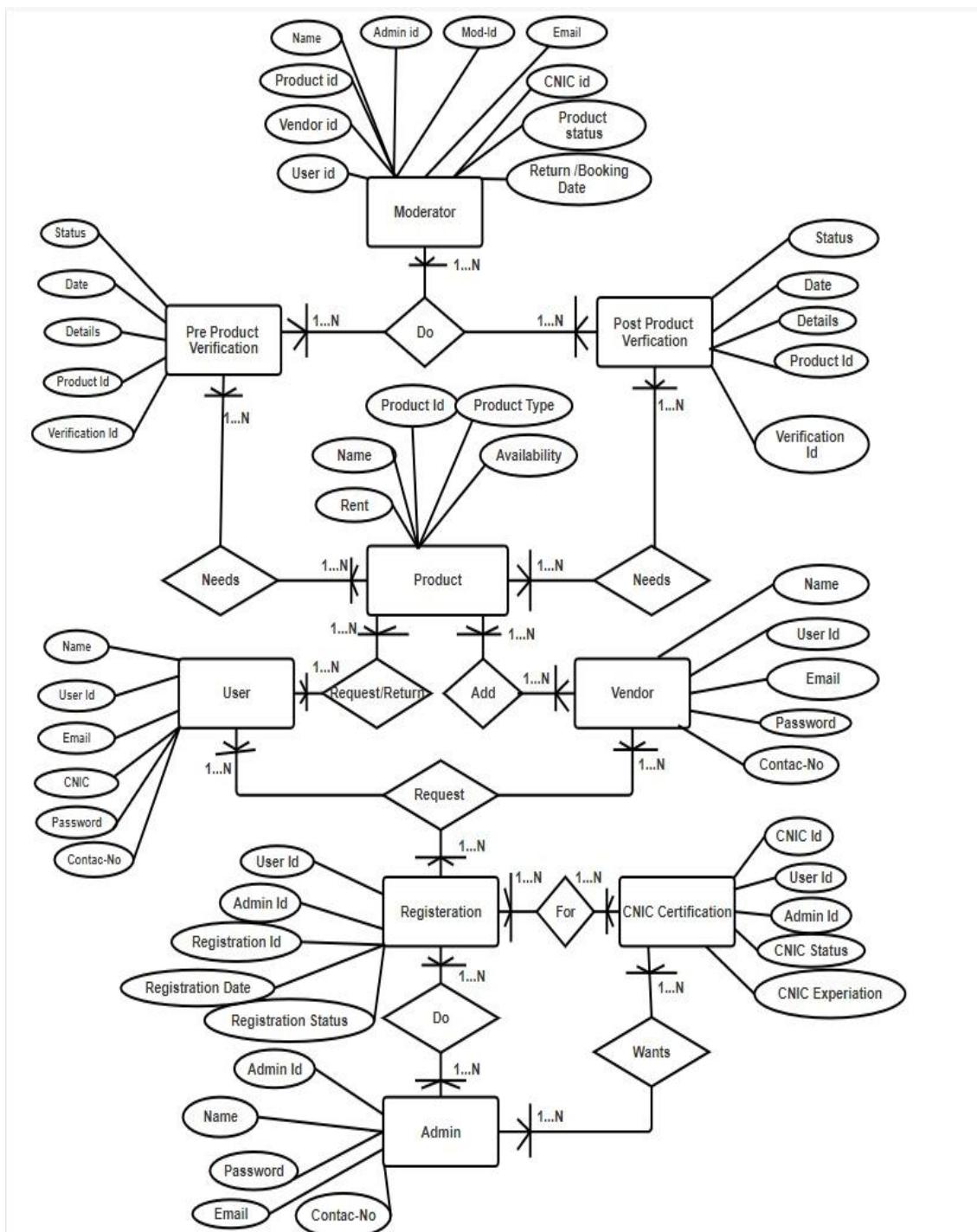


Figure 3-11 ER Diagram

The above figure 3-10 show the entity relation with each other's given below:

1. There are the 9 Entities associated with Our Application utilization
2. The pre and post product verification are handled by the moderator
3. The user and Vendor Entity have one many relations with the Product

3.3 NON-FUNCTIONAL REQUIREMENTS

1. **Usability:** The application should have an intuitive and user-friendly interface, allowing users to easily navigate, search for items, and complete rental transactions.
2. **Reliability:** The platform should be available and accessible to users with minimal downtime, ensuring a reliable rental experience.
3. **Scalability:** The application should be designed to handle a growing number of users and rental listings, scaling efficiently as the platform expands.
4. **Compatibility:** The application should be compatible with various devices, operating systems, and web browsers to ensure broad accessibility for users.
5. **Data Privacy:** The platform should comply with data protection regulations and ensure that user data is handled securely and confidentially

3.3.1 PERFORMANCE REQUIREMENTS

1. The system should handle a specified number of simultaneous users.
2. Response time for critical actions should be within acceptable limits.
3. Balance the outgoing and incoming traffic rapidly.
4. Action on Component perform the rapidly.
5. Notification, transaction, add product get product like functionality like work the efficiently.

3.3.2 SECURITY REQUIREMENTS

1. Implement secure user authentication and data encryption.
2. Authorize person can allow to access the credential.
3. Provide strong security features.
4. Ensure secure payment transactions.

3.4 SUMMARY

Chapter 3 of the RentMent application project outlines the research and system methodology employed in its development. The chapter emphasizes the use of agile methodology for its flexibility and continuous improvement approach. Various tools such as Jira, Smart Draw, Figma, and Android Studio were utilized to support the agile process. Software and hardware requirements, including operating systems, documentation tools, and server infrastructure, were identified. System constraints such as hardware limitations and platform compatibility were considered. External interface requirements, including hardware, software, and communication interfaces, were outlined to ensure seamless interaction with the application. Functional requirements, including user registration, product management, and authentication, were detailed. Use case diagrams, activity/flow chart diagrams, and sequence diagrams were provided to illustrate the application's functionality and workflow. Non-functional requirements, such as usability, reliability, and security, were also addressed. Overall, Chapter 3 sets the groundwork for the RentMent application development, providing a comprehensive understanding of its research, methodology, and requirements.

CHAPTER 4

SYSTEM IMPLEMENTATION

In this chapter, we present the detailed technical framework and architecture of the Rentment Mobile application. This includes an in-depth exploration of the system's architecture, the selection of technologies, the design and development of the database, and the mathematical models that underpin the application's functionality. Through this comprehensive analysis, we aim to demonstrate how the integration of React Native, Node.js, Express.js, and MongoDB contributes to creating a robust, scalable, and efficient solution for the rental market.

4.1 SYSTEM DEVELOPMENT PROCESS

For the development process of Rentment Application we can combine SDLC and Agile together, this combined approach ensures that the Rentment application is developed efficiently while remaining flexible enough to adapt to user needs, changes in the market, and technological advancements, the combination of **SDLC** and **Agile** ensures that the Rentment application development follows a clear, organized path while remaining adaptable to changes and improvements.



Figure 4-1 Agile SDLC

4.1.1 SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

SDLC provides the roadmap for development, ensuring that the project moves through defined phases (planning, designing, implementation, testing, deployment, and maintenance). SDLC provides a structured and systematic approach to software development. It is broken down into several phases that ensure the project is planned, executed, and completed efficiently. The main phases of the SDLC are:

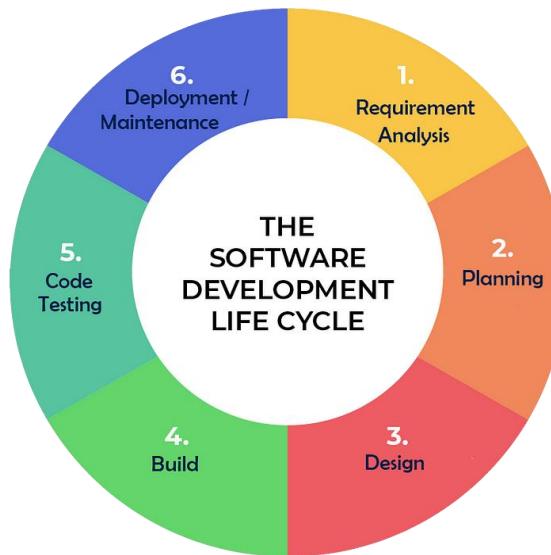


Figure 4-2 Software Development Life cycle

1. Requirement Analysis

1. **Goal:** Understand the business needs of the Rentment application.
2. **Actions:** Gather and document all requirements, including user registration, product rental, NIC verification, pre/post-product verification, and other critical functionalities.
3. **Outcome:** A Software Requirements Specification (SRS) document.

2. System Design

1. **Goal:** Create a blueprint for the Rentment system.
2. **Actions:** Design the architecture of the system, including the database structure (ERD diagrams), application logic, and user interface. Technologies such as the MERN stack (MongoDB, Express, React Native, and Node.js) are selected.
3. **Outcome:** Design documents like database schema, architecture diagrams, and interface layouts.

3. Implementation

1. **Goal:** Convert design specifications into code.
2. **Actions:** Frontend development (React Native) for user interactions and backend development (Node.js) for business logic and database management.
3. **Outcome:** Fully coded modules for user registration, product management, verification systems, and other features.

4. Testing

1. **Goal:** Identify and fix defects in the Rentment application.
2. **Actions:** Perform various tests such as unit testing, integration testing, and system testing. Testing tools like Selenium can be used.
3. **Outcome:** A bug-free, functional system.

5. Deployment

1. **Goal:** Deploy the system into a live environment.
2. **Actions:** Use Docker for containerized deployment and Firebase for backend infrastructure. Ensure compatibility across devices.
3. **Outcome:** A live and accessible Rentment mobile application.

6. Maintenance

1. **Goal:** Keep the application updated and functioning correctly.
2. **Actions:** Monitor system performance, resolve any issues, and introduce updates or new features.
3. **Outcome:** An up-to-date, smoothly functioning system.

4.2 Agile Methodology in SDLC

While the SDLC provides a structured framework, **Agile methodology** is used to introduce flexibility and iterative development. The Agile approach allows for continuous improvement, feedback, and responsiveness to changing requirements during the SDLC. Here's how Agile is incorporated into each phase:

1. Sprint Planning:

1. The project is divided into multiple **sprints**, each lasting 2-4 weeks.
2. The team collaborates with stakeholders to prioritize features and tasks, such as user registration, product management, or NIC verification.

2. Iterative Development:

1. Agile development emphasizes iterative cycles. At the start of each sprint, specific features are chosen for development.
2. For example, in one sprint, the user registration feature may be developed; in another, product verification is implemented.

3. Continuous Testing:

1. Testing is integrated into every sprint. Each module is tested as it's developed, ensuring that issues are caught early.
2. For example, after implementing NIC verification, it's tested immediately within the sprint.

4. User Feedback:

1. After each sprint, the Rentment system is demonstrated to stakeholders. Feedback is gathered and used to refine the system in subsequent sprints.
2. This ensures that the application meets user needs and expectations, adjusting the flow or features if needed.

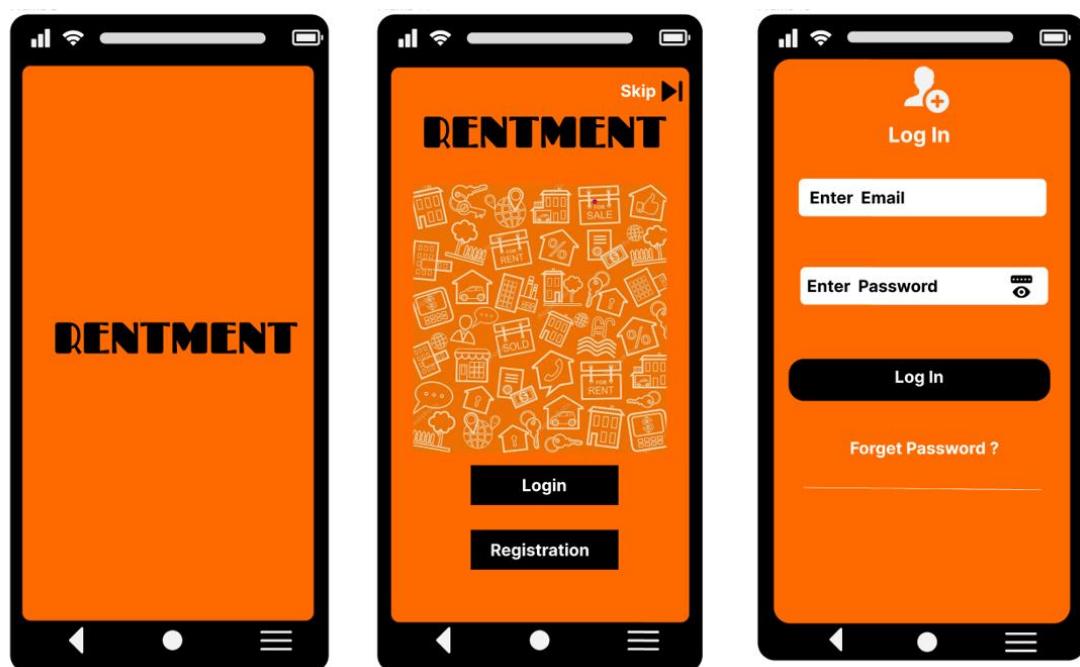
5. Deployment and Continuous Delivery:

1. Agile promotes frequent deployments, sometimes after every sprint.
2. Docker allows for easy and quick deployment of new features or updates without impacting the system's stability.

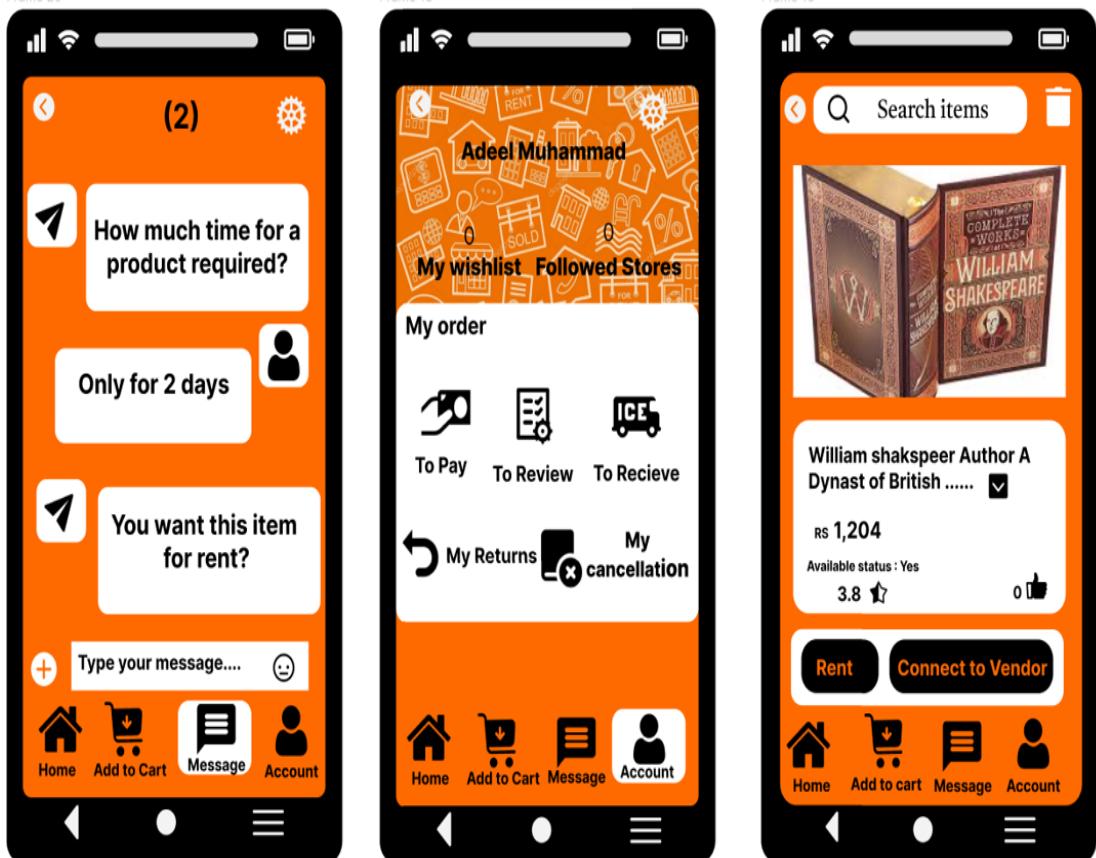
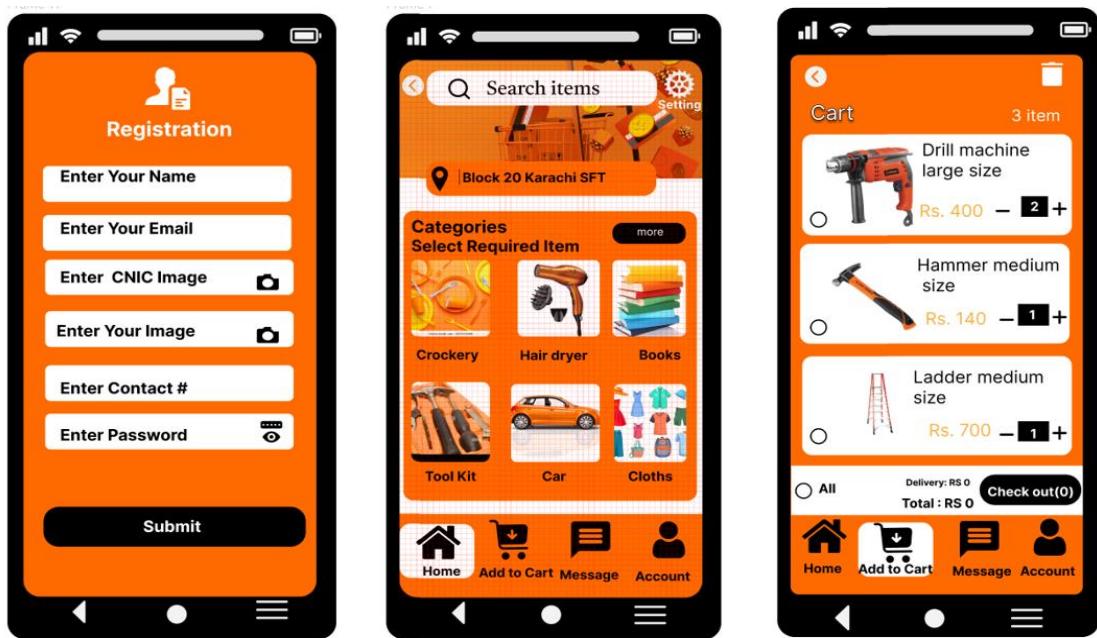
4.2 SYSTEM SIMULATION

System simulation is an essential part of the Rentment Mobile application's development process, allowing us to model, analyze, and predict the behavior of the system in a controlled environment. By simulating various components of the application, we can identify potential issues, optimize performance, and ensure that the system meets the required standards before deployment.

4.2.1 Making Simulation Model of RentMent Application:



RENTMENT APPLICATION



RENTMENT APPLICATION

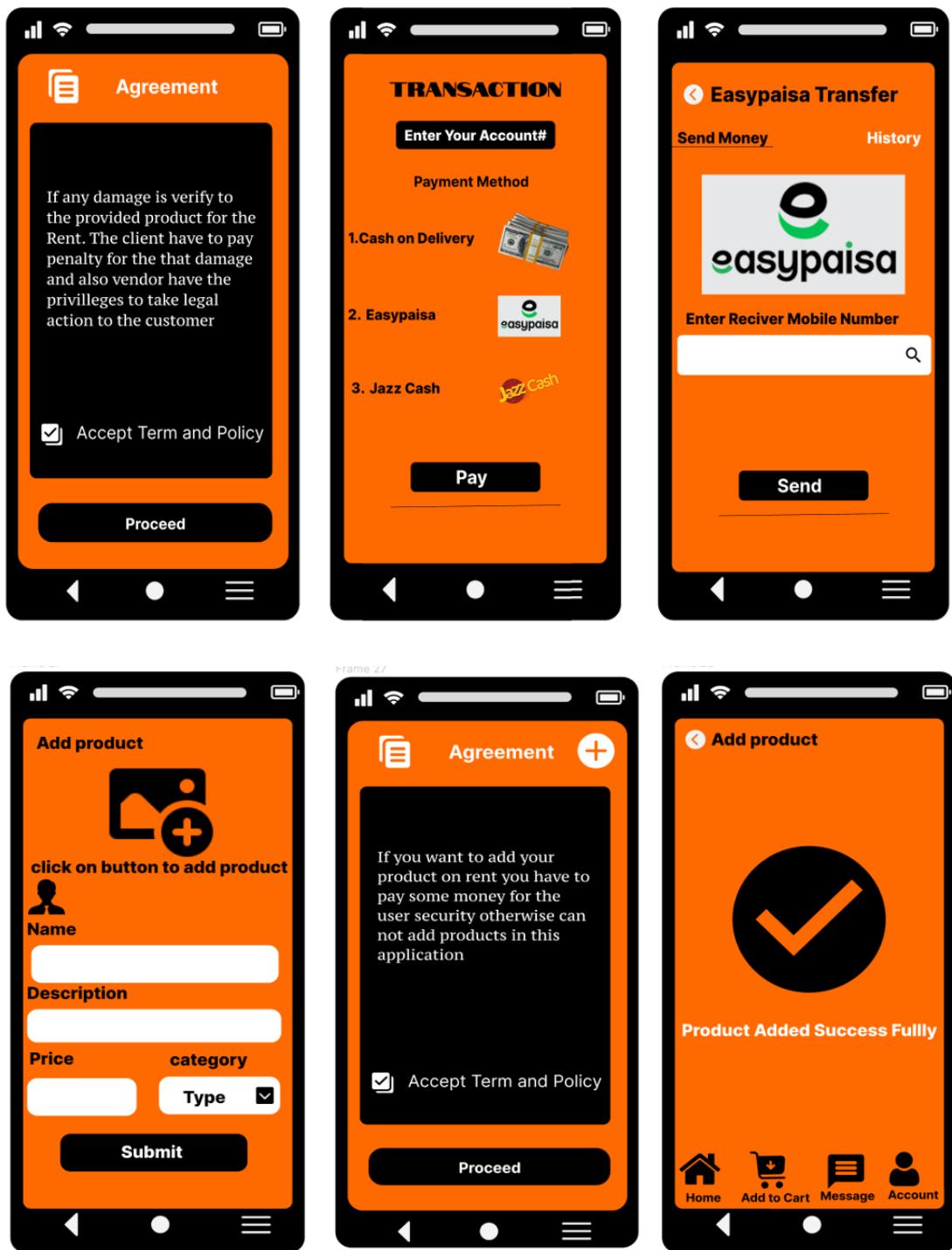


Figure 4-3 Simulation Model

4.2.2 Defining the scenarios of RentMent Simulation model:

This model define various scenarios that are considered in the simulation model for the Rentment Mobile application. These scenarios typically include:

RENTMENT APPLICATION

1. **User Interaction Scenarios:** Simulating how users interact with the application, including navigation, data entry, and response times to ensure a smooth user experience.
2. **Load Testing Scenarios:** Evaluating the system's performance under different load conditions, such as varying numbers of concurrent users, to identify potential bottlenecks.
3. **Error Handling Scenarios:** Testing how the system responds to errors or unexpected inputs, ensuring that it can gracefully handle issues without crashing.
4. **Integration Scenarios:** Assessing how well the application integrates with other systems or services, such as payment gateways or third-party APIs.
5. **Performance**
5. **Optimization Scenarios:** Identifying areas where the application can be optimized for better performance, such as reducing response times or improving resource usage. These scenarios help in thoroughly testing the application and ensuring it meets the required standards before deployment

4.3. SYSTEM ARCHITECTURE

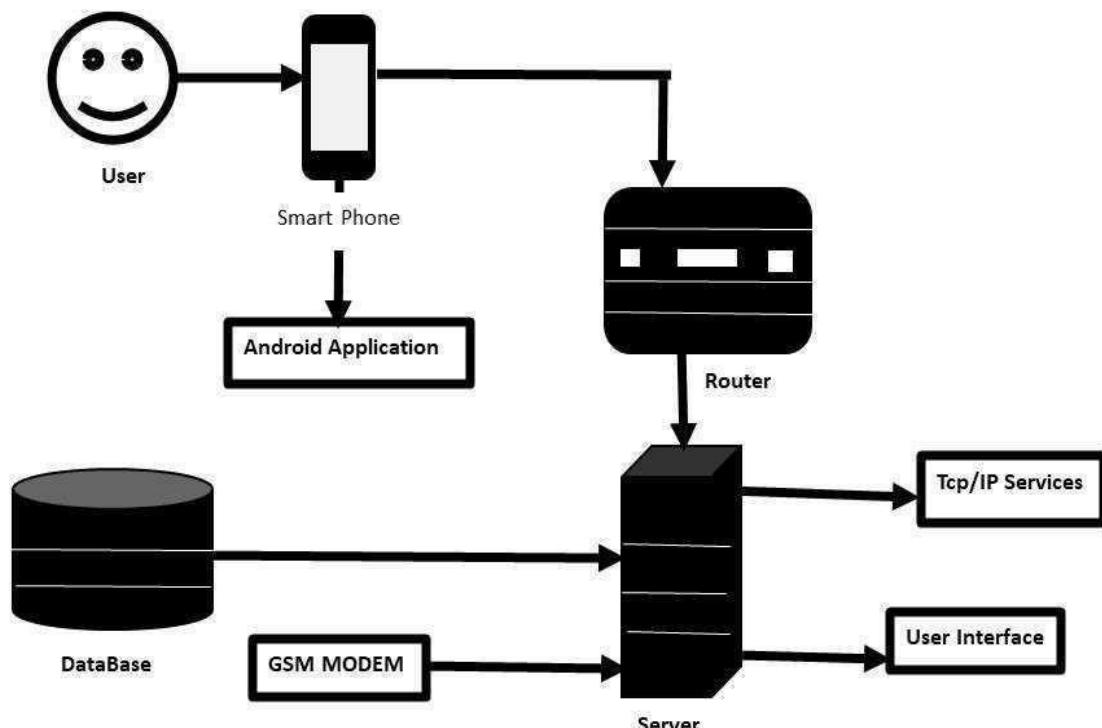


Figure 4-4 System Architecture

RENTMENT APPLICATION

The above Figure 4.3-1 represents a system architecture that likely involves a communication system using a smartphone and a server infrastructure. Here's a breakdown of the components:

1. **User:** The user interacts with the system via a smartphone.
2. **Smartphone:** The smartphone is running an Android application. This application is a core interface between the user and the backend system.
3. **Android Application:** The app on the smartphone facilitates communication between the user and the server.
4. **Router:** The router is involved in handling the network communication, likely directing traffic between the smartphone and the server. It may involve routing data packets via the internet or a local network.
5. **Server:** The server is the backbone of this system, connecting to multiple services and databases. It handles communication, processes data, and sends information back to the Android application.
6. **Database:** The database stores persistent data that the server can query. This could be user data, message logs, or other forms of structured data.
7. **GSM Modem:** The GSM modem is connected to the server and is responsible for sending or receiving data through a mobile network, possibly for sending SMS or other wireless communications.
8. **TCP/IP Services:** These services facilitate communication over the internet or an intranet using the TCP/IP protocol stack, possibly for online services like data exchange or real-time communication.
9. **User Interface:** This component could represent either a front-end interface that the user interacts with via the Android application or an admin/management interface on the server side.

This architecture outlines a system where a user interacts with an Android application on a smartphone. The application communicates with a server through a router, which processes requests and interacts with a database and GSM modem to provide services like data storage and mobile communication.

4.4 ALGORITHMS/ API/ PACKAGES USED

4.4.1 Algorithms

1. **Authentication Algorithms:** For secure login and signup processes, algorithms like bcrypt or Argon2 used for hashing passwords.
2. **Search Algorithms:** Algorithms for searching products, such as binary search or more complex algorithms like Elasticsearch for full-text search capabilities.
3. **Recommendation Algorithms:** Algorithms that suggest products to users based on their browsing history or preferences, such as collaborative filtering or content-based filtering.

4.4.2 APIs

1. **Authentication API:** RESTful APIs for user authentication (e.g., OAuth, JWT) to manage user sessions securely.
2. **Product Management API:** APIs to handle CRUD (Create, Read, Update, Delete) operations for products, allowing vendors to manage their listings.
3. **Chat API:** Real-time communication APIs (e.g., Firebase) to facilitate chat functionality between users and vendors.

4.4.3 Packages

1. **Frontend Frameworks:** Libraries like React, Angular, or Vue.js for building the user interface.
2. **Backend Frameworks:** Frameworks like Node.js with Express, Django, or Flask for server-side development.
3. **Database Packages:** ORM (Object-Relational Mapping) packages like Sequelize (for Node.js) or SQLAlchemy (for Python) to interact with databases.
4. **State Management:** Libraries like Redux or MobX for managing application state in frontend frameworks.

4.4 MATHEMATICAL MODEL

4.4.1 Data Modeling

Entity-Relationship Model (ERM): This model can be used to define the relationships between different entities in the application, such as users, products, and orders. It helps in structuring the MongoDB database effectively.

4.4.2 User Interaction and Role Management

Set Theory: Users and their roles can be represented as sets. For example, let U be the set of users and R be the set of roles. The relationships can be analyzed using set operations to determine user permissions and access levels.

4.4.3 Search Functionality

Vector Space Model: Products can be represented as vectors in a multi-dimensional space based on features (e.g., price, category, ratings). This model can be used to implement search algorithms that find similar products based on user queries.

4.5 DATABASE DESIGN AND DEVELOPMENT

4.5.1 Database Design

The Rentment Mobile application requires a robust and flexible database design to manage the diverse and complex data associated with property rentals. MongoDB, a NoSQL database, was chosen for its scalability, flexibility, and ease of integration with the chosen technology stack. This section covers the design and development process of the database, including data modeling, schema design, and optimization techniques.

4.5.1.1 Database Selection: MongoDB

1. MongoDB was selected due to its document-oriented nature, which aligns well with the requirements of the Rentment application. The schema-less

RENTMENT APPLICATION

structure of MongoDB allows for dynamic and flexible data modeling, accommodating the varying data structures of property listings, user information, and rental agreements.

2. MongoDB's ability to scale horizontally and handle large volumes of unstructured data makes it an ideal choice for a growing application like Rentment, where data types and volume can vary significantly over time.

4.5.2 Database Development

The screenshot shows the MongoDB Compass interface with the 'rental app' connection selected. The 'test' database is open, containing three collections: 'categories', 'chats', and 'contracts'. The 'categories' collection is currently selected. Each collection provides storage statistics and document counts. Below the collections, the 'test' database is expanded to show its sub-collections: 'admin', 'local', and 'categories'. The 'categories' sub-collection is highlighted. On the right, the detailed view for the 'categories' collection shows three documents with their respective IDs, names ('Clothing.', 'Electronics', 'Tools'), image URLs, and version numbers ('__v').

Collection	Storage size	Documents	Avg. document size	Indexes	Total index size
categories	20.48 kB	3	155.00 B	1	36.86 kB
chats	4.10 kB	0	0 B	2	8.19 kB
contracts	20.48 kB	4	227.00 B	1	36.86 kB

Document	Fields
1	<pre>_id: ObjectId('668249a9e78f24199c248b4b') name : "Clothing." image : "https://res.cloudinary.com/ddi9ikzpq/image/upload/v1719815072/upload/v... __v : 0</pre>
2	<pre>_id: ObjectId('668249e3e78f24199c248b55') name : "Electronics" image : "https://res.cloudinary.com/ddi9ikzpq/image/upload/v1721027422/upload/c... __v : 0</pre>
3	<pre>_id: ObjectId('6694cbf2309debf5fc6f267b') name : "Tools" image : "https://res.cloudinary.com/ddi9ikzpq/image/upload/v1721027568/upload/f... __v : 0</pre>

RENTMENT APPLICATION

local

- test**
 - categories
 - chats
 - contracts
 - favorites
 - feedbacks
 - messages
 - products
 - subcategories**
 - users

```
_id: ObjectId('66824b18e78f24199c248b7d')
name : "Laptop's"
category : ObjectId('668249e3e78f24199c248b55')
image : "https://res.cloudinary.com/ddi9ikzpq/image/upload/v1719814927/upload/w..."
__v : 0
```

```
_id: ObjectId('66824c263545d791a43a7563')
name : "Camera's"
category : ObjectId('668249e3e78f24199c248b55')
image : "https://res.cloudinary.com/ddi9ikzpq/image/upload/v1719815193/upload/k..."
__v : 0
```

```
_id: ObjectId('6694c93f309deb5fc6f2612')
name : "Men's clothing "
category : ObjectId('668249a9e78f24199c248b4b')
image : "https://res.cloudinary.com/ddi9ikzpq/image/upload/v1721026869/upload/x..."
__v : 0
```


local

- test**
 - categories
 - chats
 - contracts**
 - favorites
 - feedbacks
 - messages
 - products
 - subcategories
 - users

```
_id: ObjectId('66958e0f572c283940e90050')
user : ObjectId('6682494c872bee02fd6edc4d')
contractNo : "0003"
vendor : ObjectId('668438ed1e323100c798af60')
product : ObjectId('66843b9253c3be3d0902a242')
status : "pending"
amount : 258
isEnd : false
startDate : 2024-07-16T00:00:00.000+00:00
endDate : 2024-07-17T00:00:00.000+00:00
createdAt : 2024-07-15T21:01:03.940+00:00
updatedAt : 2024-07-15T21:01:03.940+00:00
__v : 0
```

```
_id: ObjectId('6698afb5b599c96e0daf8051')
user : ObjectId('66819d3458b3569787d9cf63')
contractNo : "0004"
vendor : ObjectId('668d259bf18cdceb72b644d')
product : ObjectId('6698ae7eb599c96e0daf7fde')
status : "inprogress"
amount : 14000
```


local

- test**
 - categories
 - chats
 - contracts
 - favorites
 - feedbacks
 - messages
 - products
 - subcategories
 - users**

```
_id: ObjectId('668248d9872bee02fd6edc38')
name : "Adeel Muhammad"
role : "vendor"
email : "adeelniaz123@gmail.com"
password : "$2b$10$T02cw10q49Mu1nbuQnf4POzwRF7Ld.Mg9k/80pz828xiP/yPxI9tq"
image : "https://res.cloudinary.com/ddi9ikzpq/image/upload/v1719814353/upload/t..."
cnic : "https://res.cloudinary.com/ddi9ikzpq/image/upload/v1719814317/upload/i..."
number : 3121054031
latitude : ""
longitude : ""
status : "approve"
isBlock : false
createdAt : 2024-07-01T06:12:41.527+00:00
__v : 0
```

```
_id: ObjectId('6682494c872bee02fd6edc4d')
name : "Farah"
role : "user"
email : "farahriazemail@gmail.com"
password : "$2b$10$jWkzPvST6qUL.dHQnyBRXeXzUlcmas3pi5iKrhM3dsN2SmQSGbzXK"
image : "https://res.cloudinary.com/ddi9ikzpq/image/upload/v1719814440/upload/u..."
```

RENTMENT APPLICATION

The screenshot shows the MongoDB Compass interface. On the left, there's a tree view of the database structure under the 'local' database:

- test
 - categories
 - chats
 - contracts
 - favorites
 - feedbacks
 - messages
 - products
 - subcategories
 - users** (highlighted in green)
 - ...

On the right, two user documents are displayed in JSON format:

```
_id: ObjectId('668d259bf18cddceb72b644d')
name : "IrfanKhan"
role : "vendor"
email : "irfan@gmail.com"
password : "$2b$10$Z/j44pXLayNw8sAKyaCoufoBaQuGxy7ZpnvYSDUz4GpsklWqwJ/y"
image : "https://res.cloudinary.com/ddi9ikzpq/image/upload/v1720526205/upload/w..."
cnic : "https://res.cloudinary.com/ddi9ikzpq/image/upload/v1720526220/upload/t..."
number : 3472938524
latitude : "24.9312656"
longitude : "66.9747178"
status : "approve"
isBlock : false
createdAt : 2024-07-09T11:57:15.601+00:00
__v : 0

_id: ObjectId('66af1685628a39ecb06c5c0a')
name : "Adeel"
role : "user"
email : "adeel@gmail.com"
password : "$2b$10$fiifjBwVVfoX7qAt60sU8uj5pOBfo7wphHeerBiId.oFLLESalX4YK"
image : "https://res.cloudinary.com/ddi9ikzpq/image/upload/v1722750517/upload/d..."
```

4.5 SUMMARY

Chapter 4 detailed the technical foundation of the Rentment Mobile application, covering its system architecture, database design, and key mathematical models. The application uses React Native for the front-end, Node.js with Express.js for the back-end, and MongoDB for database management, ensuring a responsive and scalable system. The chapter discussed how various algorithms, APIs, and packages were integrated to enhance functionality and user experience.

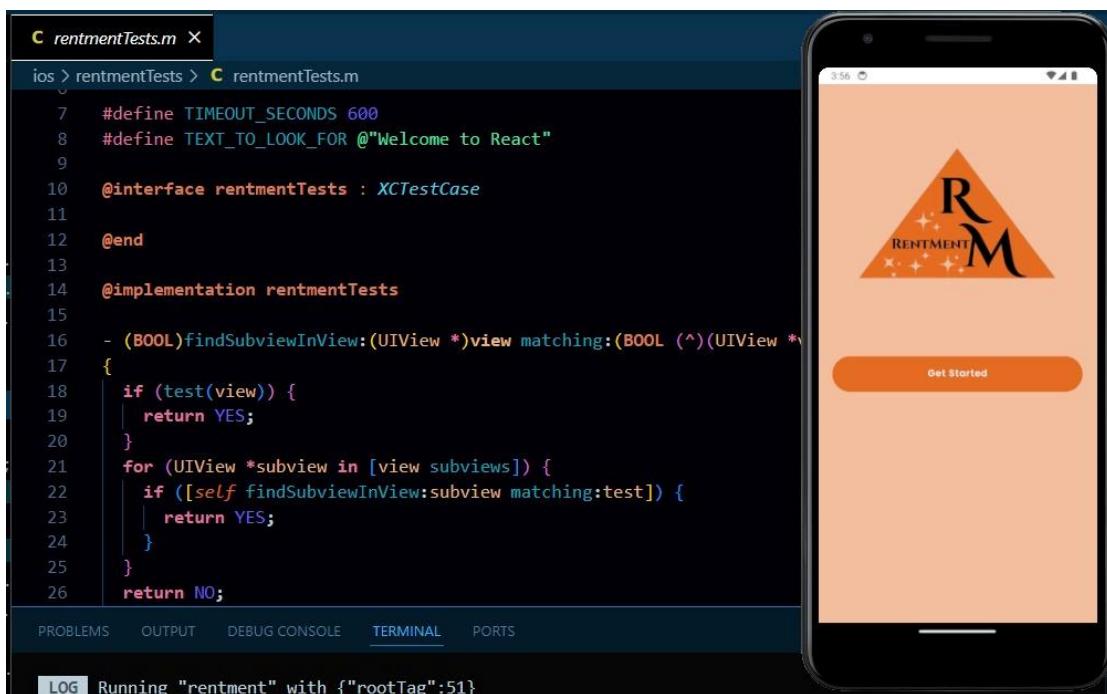
CHAPTER 5

RESULTS AND DISCUSSION

In this chapter, we present an overview of the various screens available in the application, detailing their functionalities and user interactions. Overall, this chapter serves as a guide to understanding the user interface and functionalities available to both users and vendors within the application.

5.1 RESULTS OF USER SCREENS

5.1.1 Splash Screen



This is the initial screen that users see when they open the application. It serves as a loading screen while the app initializes.

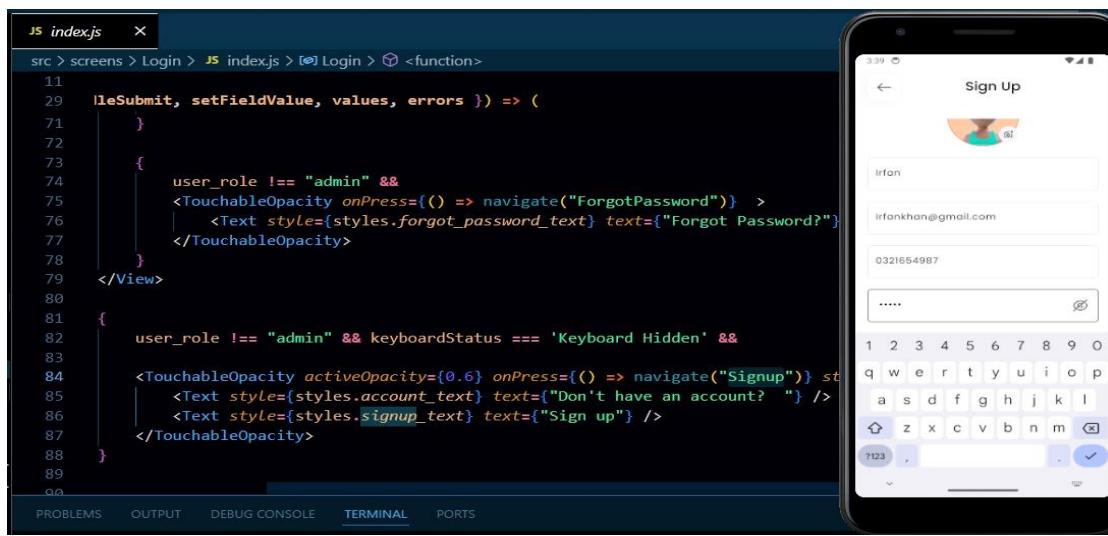
RENTMENT APPLICATION

5.1.2 Select Role Screen



In this screen, users can select their role by clicking on the appropriate user button. This helps the application tailor the experience based on whether the user is a regular user or a vendor.

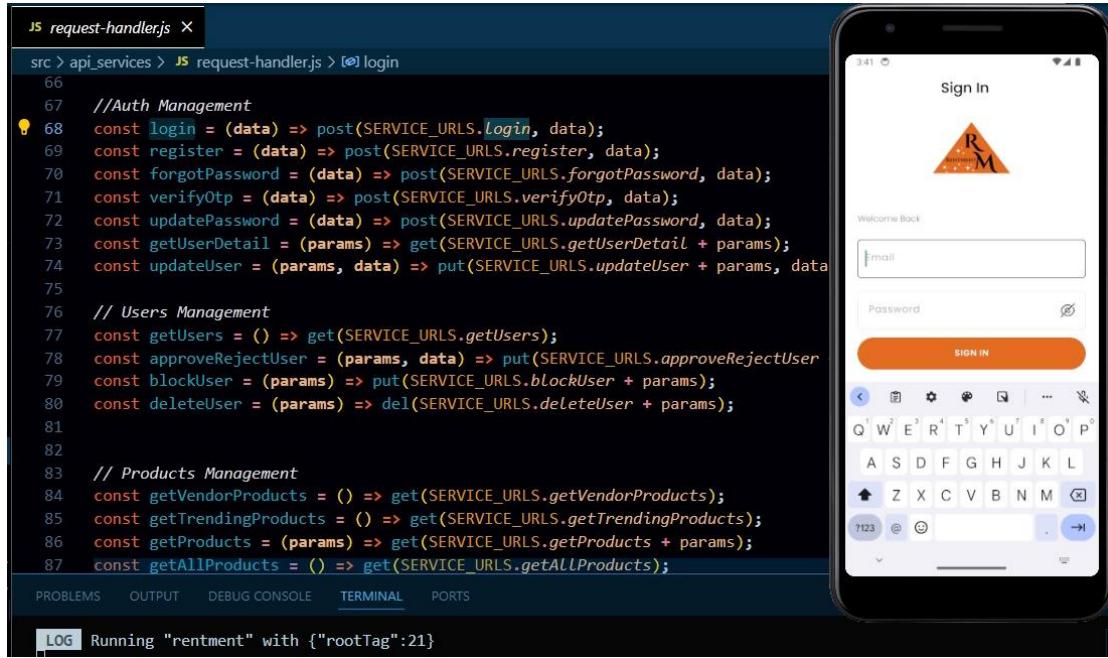
5.1.3 Signup Screen



Users/Vendors can request to sign up by providing their personal details, uploading an image, and entering their CNIC (Computerized National Identity Card) number. This screen is essential for new users/Vendors to create an account.

RENTMENT APPLICATION

5.1.4 Sign In Screen



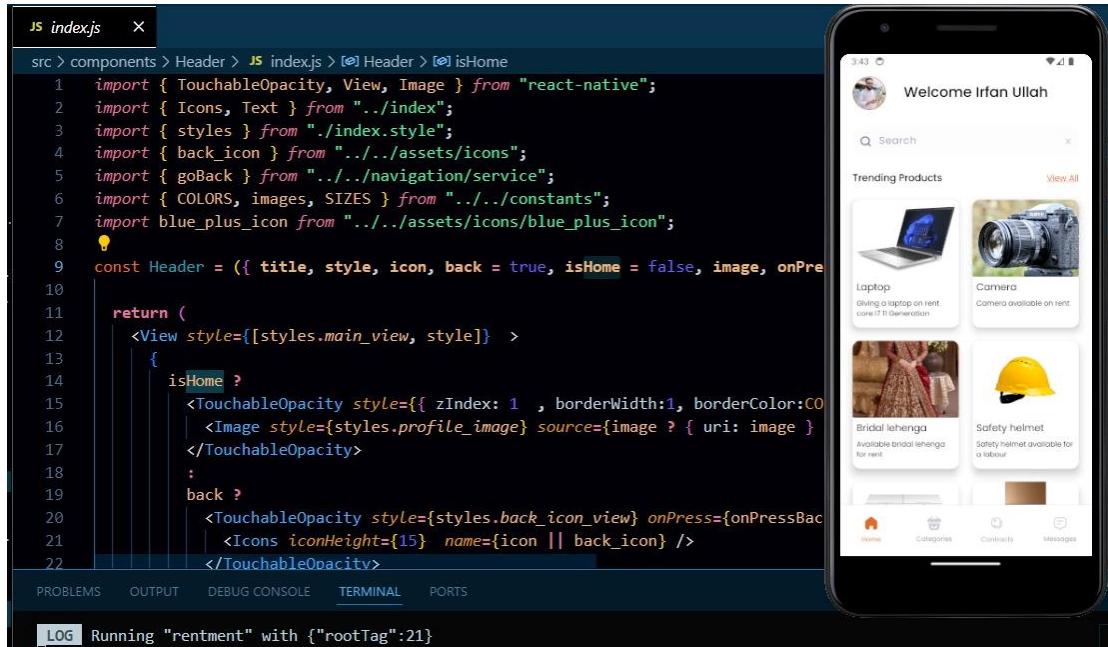
The screenshot shows the React Native development interface. On the left, the code editor displays `request-handler.js` with various API endpoints for user and product management. On the right, a mobile phone screen shows the 'Sign In' screen of the RENTMENT application. The screen features a logo with 'R' and 'M', a 'Welcome Back' message, email and password input fields, and a large orange 'SIGN IN' button.

```
JS request-handler.js x
src > api_services > JS request-handler.js > [o] login
66
67 //Auth Management
68 const login = (data) => post(SERVICE_URLS.Login, data);
69 const register = (data) => post(SERVICE_URLS.register, data);
70 const forgotPassword = (data) => post(SERVICE_URLS.forgotPassword, data);
71 const verifyOtp = (data) => post(SERVICE_URLS.verifyOtp, data);
72 const updatePassword = (data) => post(SERVICE_URLS.updatePassword, data);
73 const getUserDetail = (params) => get(SERVICE_URLS.getUserDetail + params);
74 const updateUser = (params, data) => put(SERVICE_URLS.updateUser + params, data);
75
76 // Users Management
77 const getUsers = () => get(SERVICE_URLS.getUsers);
78 const approveRejectUser = (params, data) => put(SERVICE_URLS.approveRejectUser + params, data);
79 const blockUser = (params) => put(SERVICE_URLS.blockUser + params);
80 const deleteUser = (params) => del(SERVICE_URLS.deleteUser + params);
81
82
83 // Products Management
84 const getVendorProducts = () => get(SERVICE_URLS.getVendorProducts);
85 const getTrendingProducts = () => get(SERVICE_URLS.getTrendingProducts);
86 const getProducts = (params) => get(SERVICE_URLS.getProducts + params);
87 const getAllProducts = () => get(SERVICE_URLS.getAllProducts);

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
LOG Running "rentment" with {"rootTag":21}
```

This screen allows users to log into the application by entering their email and password. It is crucial for returning users to access their accounts.

5.1.5 Home Screen



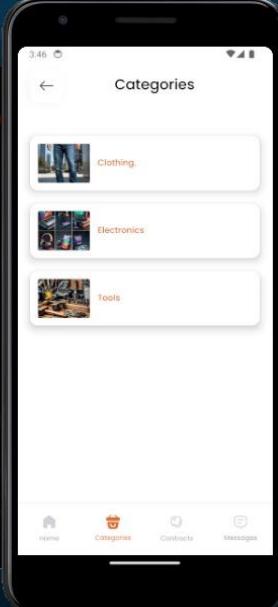
The screenshot shows the React Native development interface. On the left, the code editor displays `index.js` which defines the `Header` component. On the right, a mobile phone screen shows the 'Home' screen of the RENTMENT application. The screen displays a 'Welcome' message for 'Irfan Ullah', a search bar, and a section for 'Trending Products' featuring items like a laptop, camera, bridal lehenga, and safety helmet.

```
JS index.js x
src > components > Header > JS index.js > [o] Header > [o] isHome
1 import { Touchableopacity, View, Image } from "react-native";
2 import { Icons, Text } from "../index";
3 import { styles } from "./index.style";
4 import { back_icon } from "../../assets/icons";
5 import { COLORS, images, SIZES } from "../../constants";
6 import blue_plus_icon from "../../assets/icons/blue_plus_icon";
7
8 const Header = ({ title, style, icon, back = true, isHome = false, image, onPressBack }) =>
9
10   return (
11     <View style={[styles.main_view, style]} >
12       {
13         isHome ?
14           <Touchableopacity style={{ zIndex: 1, borderwidth: 1, bordercolor: CO
15             <Image style={styles.profile_image} source={image ? { uri: image } : null} />
16           </Touchableopacity>
17         :
18           back ?
19             <Touchableopacity style={styles.back_icon_view} onPress={onPressBack}>
20               <Icons iconheight={15} name={icon || back_icon} />
21             </Touchableopacity>
22       }
23     </View>
24   );
25
26 export default Header;

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
LOG Running "rentment" with {"rootTag":21}
```

For users, this screen displays trending products and allows them to browse through various offerings.

5.1.6 Categories Screen



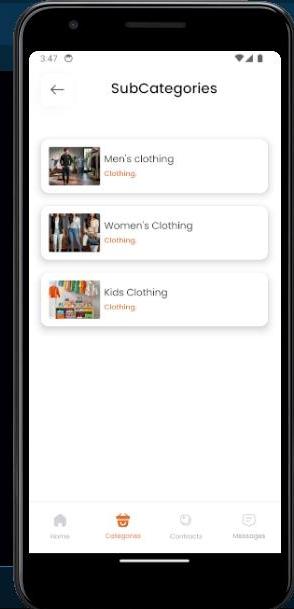
```
JS CategoryStack.js X
src > navigation > JS CategoryStack.js > [o] CategoryStack
1 import React from "react";
2 import { createStackNavigator } from "@react-navigation/stack";
3 import { SubCategories, Categories, Products, ProductDetail, AddCategory, AddSubCategory } from "../screens";
4 import AddCategory from "../screens/AddCategory";
5
6 const CategoryStack = () => {
7   const Stack = createStackNavigator();
8
9   return (
10     <>
11       <Stack.Navigator
12         initialRouteName="Categories"
13         screenOptions={{ headerShown: false }}>
14         >
15           <Stack.Screen name="Categories" component={Categories} />
16           <Stack.Screen name="SubCategories" component={SubCategories} />
17           <Stack.Screen name="AddCategory" component={AddCategory} />
18           <Stack.Screen name="AddSubCategory" component={AddSubCategory} />
19           <Stack.Screen name="Products" component={Products} />
20           <Stack.Screen name="ProductDetail" component={ProductDetail} />
21           <Stack.Screen name="ViewProfile" component={ViewProfile} />
22         </Stack.Navigator>

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

This screen presents all the main product categories available in the application, allowing users to navigate to specific types of products.

5.1.7 Sub Categories Screen



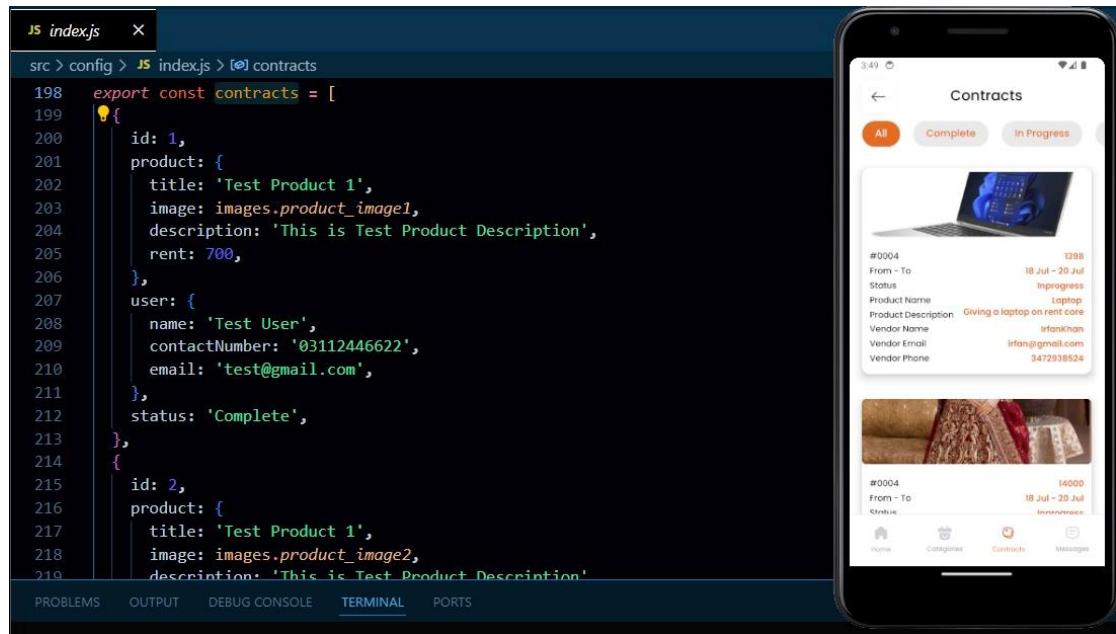
```
JS index.js X
src > config > JS index.js > [o] subcategories
156 export const subcategories = [
157   {
158     id: 1,
159     category: {
160       name: 'Test Category',
161     },
162     name: 'Test Sub Category 1',
163     image: images.category_image,
164   },
165   {
166     id: 2,
167     category: {
168       name: 'Test Category',
169     },
170     name: 'Test Sub Category 2',
171     image: images.category_image,
172   },
173   {
174     id: 3,
175     category: {
176       name: 'Test Category',
177     },
178   }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

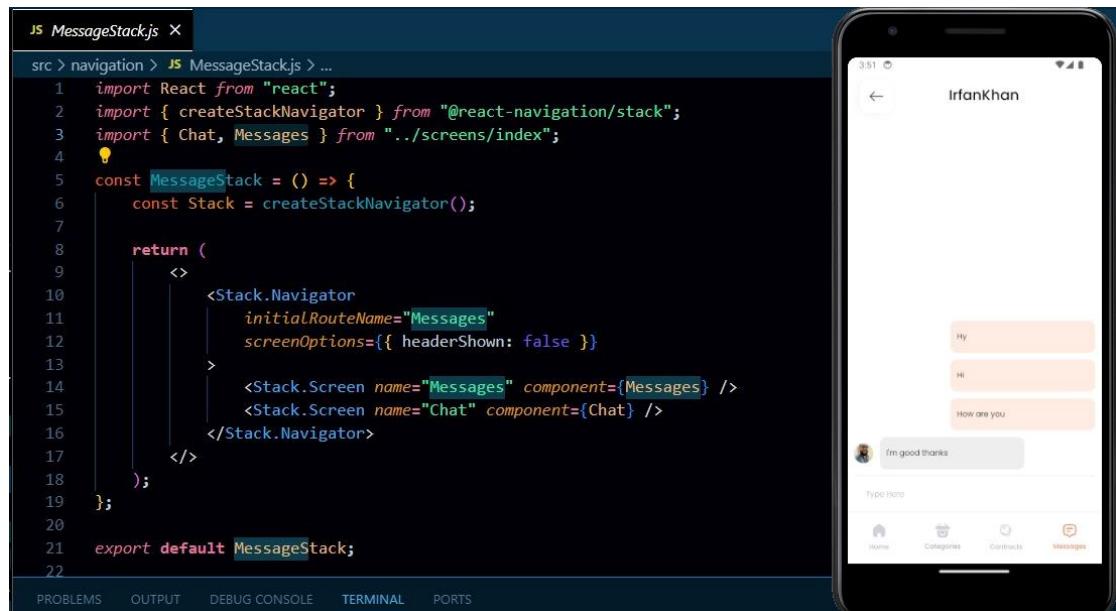
Here, users can view the subcategories under each main category, helping them to refine their search for products.

5.1.8 Contracts Screen



This screen allows users to view all their contracts with vendors, including both completed and pending contracts. It helps users keep track of their agreements.

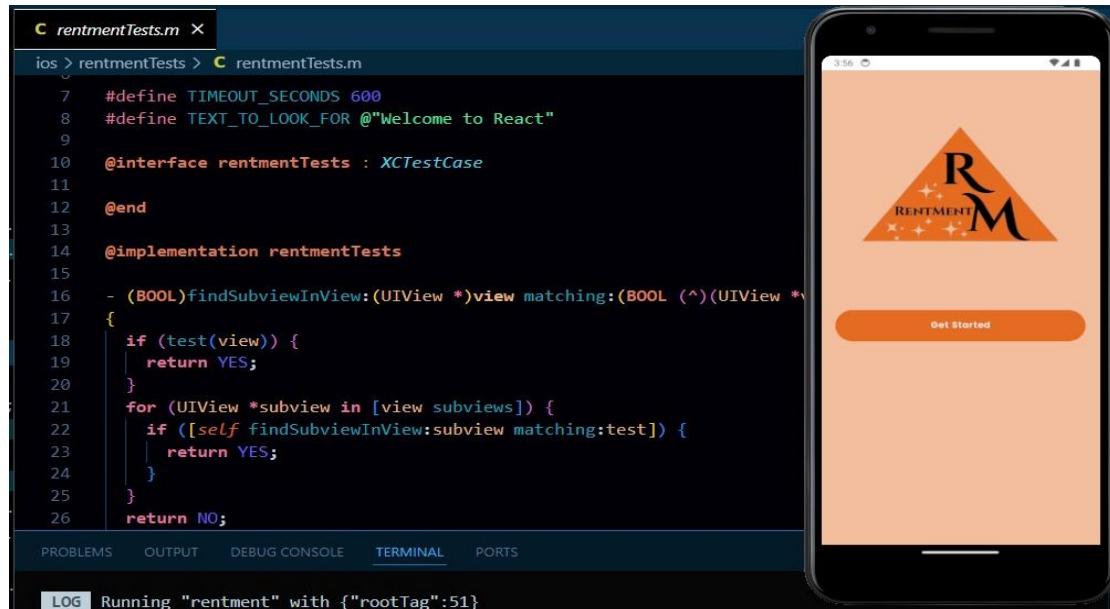
5.1.9 Chat Screen



In this screen, users can communicate directly with vendors, facilitating inquiries and discussions about products or services.

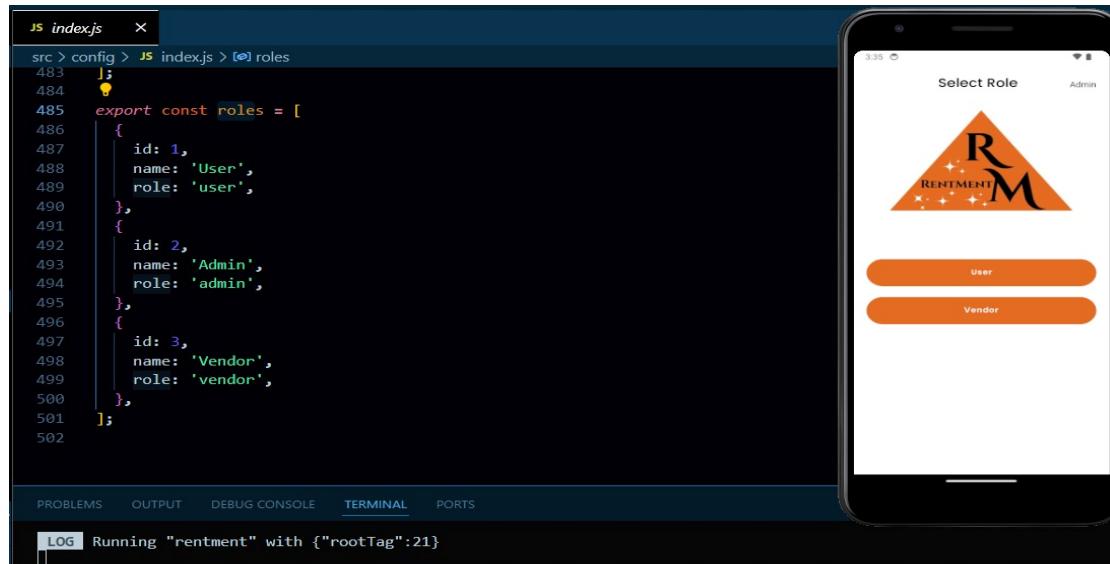
5.2 RESULTS OF VENDOR SCREENS

5.2.1 Splash Screen



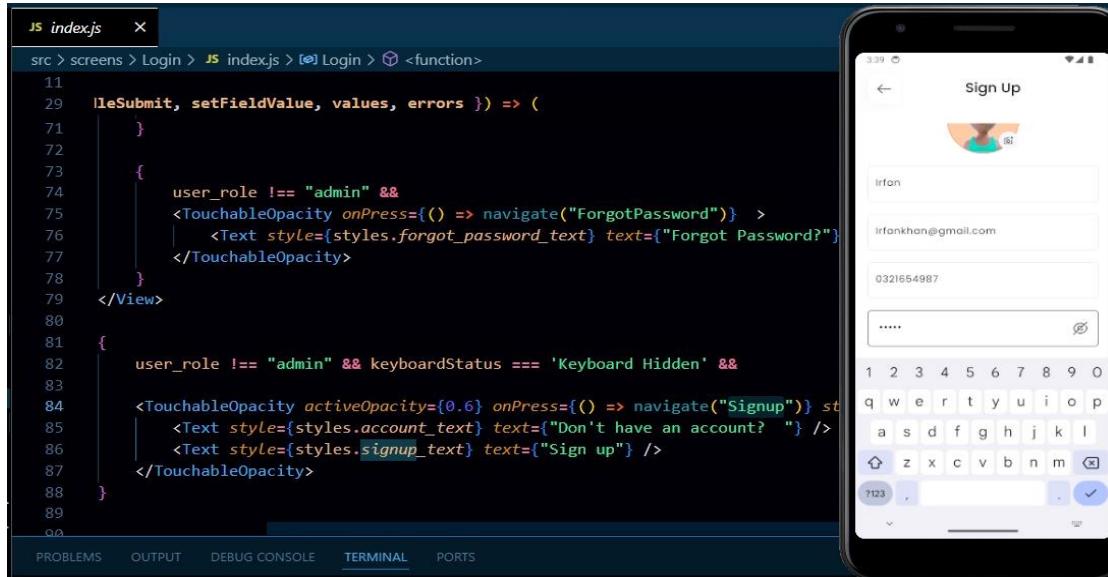
This is the initial screen that users see when they open the application. It serves as a loading screen while the app initializes.

5.2.2 Select Role Screen



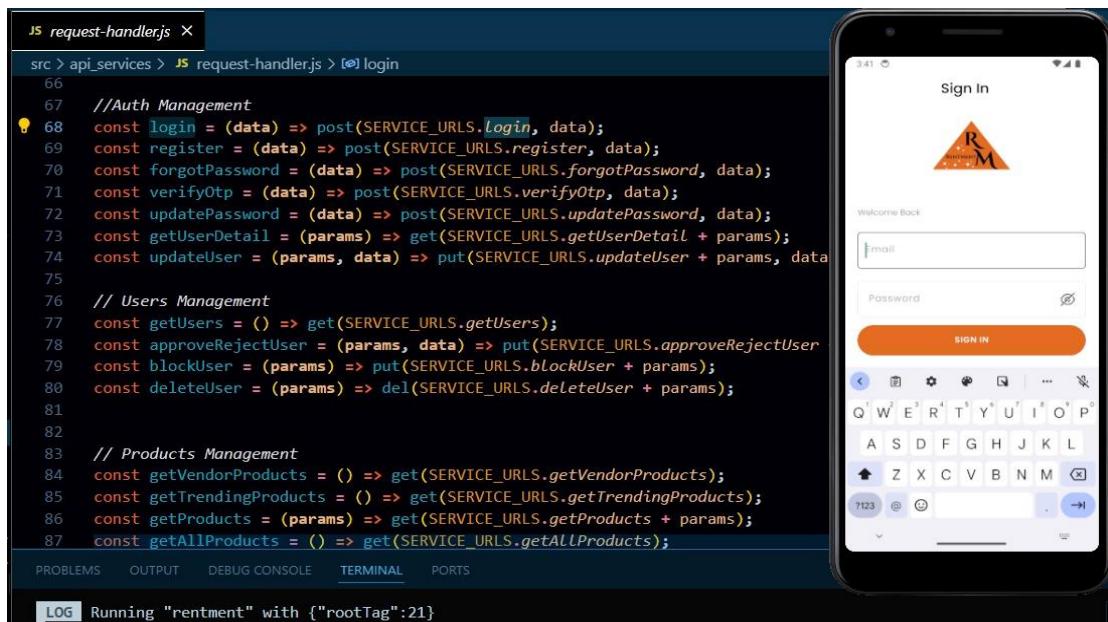
In this screen, users can select their role by clicking on the appropriate user button. This helps the application tailor the experience based on whether the user is a regular user or a vendor.

5.2.3 Signup Screen



Users/Vendors can request to sign up by providing their personal details, uploading an image, and entering their CNIC (Computerized National Identity Card) number. This screen is essential for new users/Vendors to create an account.

5.2.4 Sign In Screen



This screen allows users to log into the application by entering their email and password. It is crucial for returning users to access their accounts.

RENTMENT APPLICATION

5.2.5 Home Screen



The screenshot shows the 'Product Detail' screen. At the top, there's a navigation bar with a back arrow and the text 'Product Detail'. Below it is a large image of a silver laptop. To the right of the image, the text 'Laptop' is displayed, followed by a description: 'Giving a laptop on rent core i7 11 Generation' and 'Electronics / Laptop's'. Below this, a section titled 'Vendor Detail' shows information: Name (Intention), Email (intention@gmail.com), and Contact Number (9479898524). At the bottom right is an orange button labeled 'Edit Product'. At the very bottom of the screen, there are four navigation icons: Home, My Products, Contracts, and Messages.

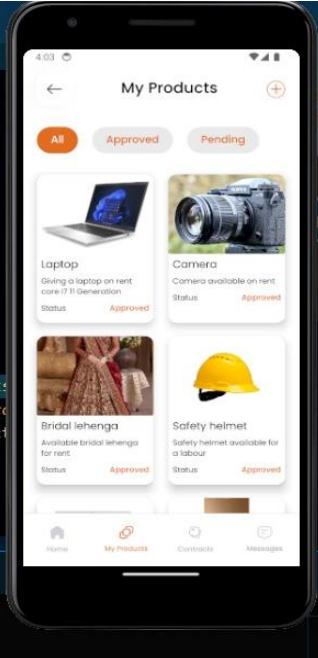
```
js index.js
src > screens > ProductDetail > js index.js > [o] ProductDetail >  <function>
  7 import { Button } from '../components/index'
  8 import ProductDetailFunctional from './index.function'
  9 import AgreementModal from './components/AgreementModal'
 10 import { navigate } from '../../../../../navigation/service'
 11
 12 const ProductDetail = ({ route }) => {
 13
 14   const { detail } = route.params;
 15
 16   const {
 17     setShowModal,
 18     showModal,
 19     user_role,
 20     onApprove,
 21     update_loading,
 22     product_detail,
 23     add_contract_loading,
 24     onPressAdd,
 25   } = ProductDetailFunctional({ detail })
 26
 27   return (
    ...
  )
  
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

LOG Running "rentment" with {"rootTag":51}

For vendors, it shows all the products they have uploaded, providing a quick overview of their inventory.

5.2.4 Products Screen



The screenshot shows the 'My Products' screen. At the top, there's a navigation bar with a back arrow and the text 'My Products'. Below it is a filter bar with three buttons: 'All', 'Approved', and 'Pending'. The main area displays four product cards: 1. Laptop: 'Giving a laptop on rent core i7 11 Generation', Status: Approved. 2. Camera: 'Camera available on rent', Status: Approved. 3. Bridal lehenga: 'Available bridal lehenga for rent', Status: Approved. 4. Safety helmet: 'Safety helmet available for a labour', Status: Approved. At the bottom right is an orange '+' button. At the very bottom of the screen, there are four navigation icons: Home, My Products, Contracts, and Messages.

```
js MyProductStack.js
src > navigation > js MyProductStack.js > [o] MyProductStack
  1 import React from "react";
  2 import { createStackNavigator } from "@react-navigation/stack";
  3 import { ProductDetail, MyProducts, AddProduct } from "../screens";
  4
  5 const MyProductStack = () => {
  6   const Stack = createStackNavigator();
  7
  8   return (
  9     <>
 10       <Stack.Navigator
 11         initialRouteName="MyProducts"
 12         screenOptions={{ headerShown: false }}
 13       >
 14         <Stack.Screen name="MyProducts" component={MyProducts} />
 15         <Stack.Screen initialParams={()} name="AddProduct" component={AddProduct} />
 16         <Stack.Screen name="ProductDetail" component={ProductDetail} />
 17       </Stack.Navigator>
 18     </>
 19   );
 20 }
  
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

LOG Running "rentment" with {"rootTag":51}

In this screen vendors can see all their approved or pending products.

5.2.5 Contracts Screen

The screenshot shows the React Native development interface with the following components:

- Code Editor:** Displays the `index.function.js` file containing the logic for filtering contracts based on their status (complete, pending, inprogress). It also includes an `onRefresh` function.
- Terminal:** Shows the command `LOG Running "rentment" with {"rootTag":51}`.
- Simulator:** Shows the 'Contracts' screen on an iPhone X simulator. The screen has tabs for 'All', 'Complete', and 'In Progress'. It lists two contracts:
 - #0004: Bridal lehenga, 14000, From - To: 18 Jul - 20 Jul, Status: Inprogress.
 - #0004: Available bridal lehenga for, 1398, From - To: 18 Jul - 20 Jul, Status: Incomplete.

This screen allows vendors to view all their contracts with users, including both completed and pending contracts. It helps vendors keep track of their agreements.

5.2.6 Message Screen

The screenshot shows the React Native development interface with the following components:

- Code Editor:** Displays the `MessageStack.js` file, which uses `createStackNavigator` to define a stack navigator for 'Messages' and 'Chat' screens.
- Terminal:** Shows the command `LOG Running "rentment" with {"rootTag":51}`.
- Simulator:** Shows the 'Messages' screen on an iPhone X simulator. It displays a message from 'Irfan Ulah' with the text 'I'm good thanks'. The screen has tabs for 'Home', 'My Products', 'Contracts', and 'Messages'.

In this screen, vendors can communicate directly with users, facilitating inquiries and discussions about products or services.

RENTMENT APPLICATION

5.2.7 Add Product Screen

The screenshot shows the 'Edit Product' screen on a mobile device and the corresponding code in a VS Code editor. The mobile screen displays fields for Product Name (Laptop), Product Description (g a laptop on rent core i7 11 Generation), Product Rent Per Day (699), Product Category (Electronics), and Product Sub Category (Laptop's). Below the form are navigation icons for Home, My Products, Contracts, and Messages. The VS Code editor shows the 'index.js' file for the 'AddProduct' screen, which contains code for rendering a container, header, formik, and scroll view.

```
JS index.js x
src > screens > AddProduct > JS index.js > (e) AddProduct
14  const AddProduct = ({ route }) => {
41    return (
42      <Container>
43        <Header
44          title={data ? "Edit Product" : "Add Product"}
45        />
46        <Formik
47          validationSchema={product_validation_schema}
48          initialValues={
49            name: data?.name || "",
50            image: data?.image || "",
51            description: data?.description || "",
52            amount: data?.amount?.toString() || ""
53          }
54          onSubmit={(values) => onAddProduct(values)}
55        >
56          {({ handleSubmit, setFieldValue, values, errors }) => (
57            <>
58              <View style={styles.main_view} >
59                <ScrollView keyboardShouldPersistTaps="handled"
60                  >
```

In this screen vendor can add or edit their products.

5.3 RESULTS OF ADMIN SCREENS

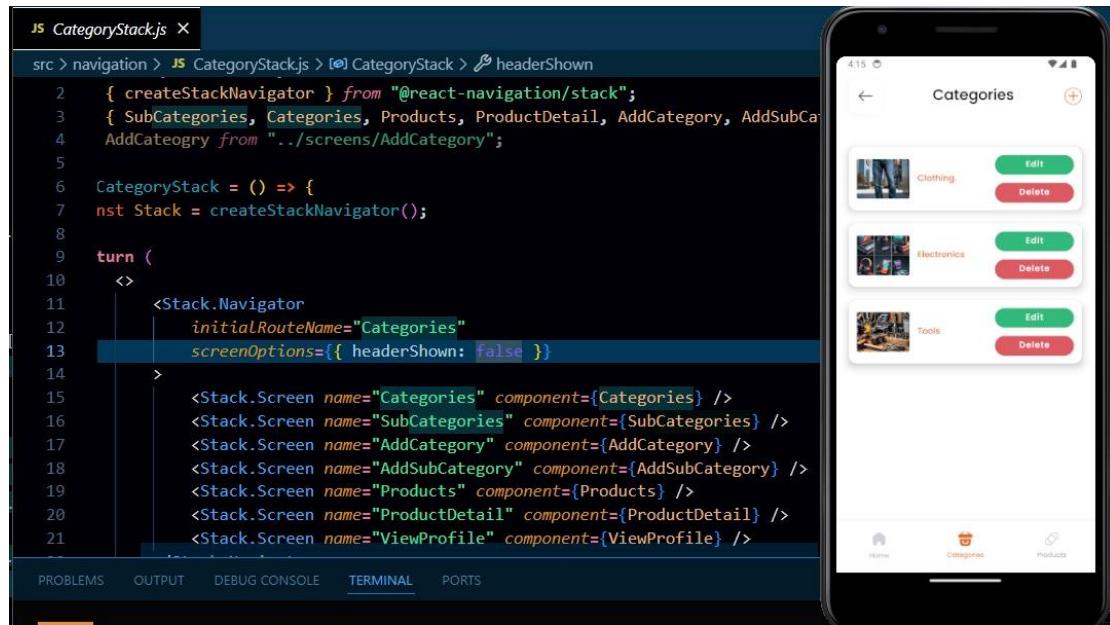
5.3.1 Home Screen

The screenshot shows the Admin Home screen on a mobile device and its corresponding code in VS Code. The mobile screen displays a welcome message 'Welcome RENTMENT Admin', a section for 'Approve/Reject Users' showing a user named 'Irfan Ullah' with status 'Approved', and a 'User Details' section for 'Aslam Muhammad' with status 'Approved'. Below these are navigation icons for Home, Categories, and Products. The VS Code editor shows the 'SingleUserView.js' file for the 'Home' component, which handles user approval/rejection logic.

```
JS SingleUserView.js x
src > screens > Home > components > JS SingleUserView.js > (e) SingleUserView
8  _userView = ({ item, onPressAccept, onPressReject, onPressBlock, onPressDelete }) => {
34    <Text
35      style={styles.number}
36      text={item?.email || "--"} />
37    <View style={{ height: SIZES.padding2 * 0.2 }} />
38    <Text
39      style={styles.number}
40      text={`User Role: ${capitalize(item?.role)}}` />
41  </View>
42  <View>
43    <Text
44      style={styles.number}
45      text={item?.status === 'approve' ? "Approved" : item?.status === 'pending' ? "Pending" :
46    </View>
47  </View>
48
49  'item?.status === "pending" && (
50    <View style={[styles.row, { justifyContent: "space-between" }]} >
51      <Button
52        style={styles.accept_btn}
```

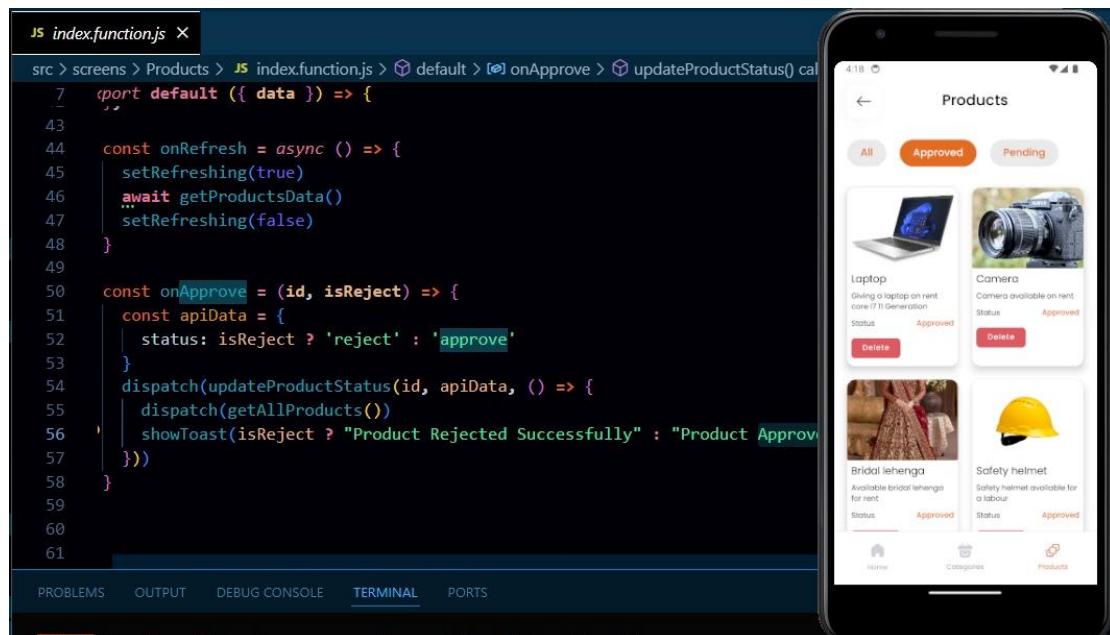
In this screen admin can accept or reject user and vendor request or delete or block user and vendor.

5.3.2 Categories Screen



In this screen admin can add categories or sub categories or edit, or delete categories or sub categories.

5.3.3 Products Screen



In this Screen admin can delete or approved product or see the pending products.

5.4 SUMMARY

This chapter provides a comprehensive overview of the various screens within the application, detailing their functionalities and the user interactions they facilitate. It is structured to cover three main user roles: users, vendors, and admins, each with distinct screens tailored to their needs.

Throughout the chapter, the significance of each screen in contributing to a seamless user experience is emphasized. Visual aids, such as screenshots and flowcharts, are suggested to enhance understanding. The chapter ultimately serves as a vital resource for comprehending the application's interface, ensuring that all users can effectively navigate and utilize the features available to them. Future enhancements and user feedback are also mentioned, indicating ongoing development and improvement of the application.

CHAPTER 6

CONCLUSION AND FUTURE DIRECTIONS

In this chapter we present potential directions and advancements that the RentMent application can take in the future, specifically focusing on the integration of AI technologies. This chapter outlines several areas where AI can be applied to enhance various aspects of the platform, such as recommendation systems, chatbot assistance, and image and object recognition, fraud detection and security, dynamic pricing, sentiment analysis, predictive maintenance, and expansion to voice and smart devices. In this chapter we can also define the conclusion of RentMent Application where we can know the need, scope and working of RentMent Application.

6.1 CONCLUSION

The conclusion of the RentMent application project is that it aims to revolutionize the rental industry in Pakistan by providing a centralized platform for renting and leasing various items or services. The project has conducted a comprehensive review of existing rental platforms and identified their strengths and weaknesses.

Based on this review, the RentMent application aims to address the shortcomings of existing platforms by incorporating comprehensive features, ensuring security, and providing a user-friendly experience. The project emphasizes the importance of online transactions, product availability, user and product verification, vendor details, damage policies, and secure transactions.

The RentMent application project strives to provide a reliable, secure, and user-friendly platform that meets the needs of both renters and vendors in Pakistan's rental industry. By leveraging AI technologies and addressing the gaps in existing platforms, RentMent aims to offer a comprehensive solution for renting and leasing, revolutionizing the way people access and share resources.

6.2 FUTURE DIRECTIONS

The future direction of the RentMent application project with AI can involve the integration of artificial intelligence technologies to enhance various aspects of the platform.

Here are some potential directions:

6.2.1 Intelligent Recommendation System

Implement an AI-powered recommendation system that analyzes user preferences, rental history, and other relevant data to provide personalized recommendations for items or services that users may be interested in. This can improve the user experience and increase engagement on the platform.

6.2.2 Chatbot Assistance

Develop a chatbot feature that utilizes natural language processing (NLP) and machine learning techniques to provide automated customer support and assistance. The chatbot can help users with inquiries, provide guidance on rental processes, and address common issues, improving overall user satisfaction and reducing the need for manual support.

6.2.3 Image and Object Recognition

Utilize computer vision technology to enable image and object recognition capabilities. This can allow users to upload images of items they want to rent, and the system can automatically categorize and tag them, making it easier for other users to find and browse relevant listings.

6.2.4 Fraud Detection and Security

Implement AI algorithms to detect and prevent fraudulent activities on the platform. This can involve analyzing user behavior patterns, transaction history, and other relevant data to identify suspicious activities and protect both renters and vendors from potential scams.

6.2.5 Dynamic Pricing

Utilize AI algorithms to optimize rental pricing based on various factors such as demand, availability, and market trends. This can help users find competitive prices and assist vendors in maximizing their rental earnings.

6.2.6 Sentiment Analysis

Apply sentiment techniques to analyze user reviews and feedback. This can provide valuable insights into the quality of rental experiences, identify areas for improvement, and help maintain a high standard of service on the platform.

6.2.7 Predictive Maintenance

Implement predictive maintenance algorithms to monitor the condition of rental items and predict maintenance or repair needs. This can help vendors proactively address any issues, ensuring that rented items are in good working condition and reducing downtime for users.

6.2.8 Expansion to Voice and Smart Devices

Explore integration with voice assistants and smart devices to enable users to interact with the RentMent application through voice commands or smart home devices. This can enhance convenience and accessibility for users, allowing them to browse, rent, and manage their rentals hands-free.

REFERENCES

- (1) G. W. W. Tiba and A. H. Rangkuti, "Development of Mobile-Based Information Letter of Rukun Tetangga And Rukun Warga With Mern Stack Method," 2022 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS), Jakarta, Indonesia, 2022, June.
- (2) Aleksandar Simovic; Slobodan Stojanovic, Serverless Applications with Node.js: Using AWS Lambda and Claudia.js , Manning, 2019.
- (3) Shiv Kumar Goel. (2019). A Comparative Study of Agile & Devops Methodology. International Journal for Research in Applied Science and Engineering Technology, 7(6), 2019
- (4) Bhavyaa, Mayuri Gupta, & Vaishali. (2021). Comprehensive Study of MERN Stack - Architecture, Popularity and Future Scope. International Journal of Scientific Research in Computer Science, Engineering and Information Technology 4 th (2021)
- (5) S. P. Uniyal, K. Joshi, V. K. Singh, A. Aggarwal, G. Chhabra and A. Kumar, "Comparative Analysis of App Size Variations between React Native and Apache Cordova Powered Android Applications," 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Trichy, India, 2023,
- (6) N. Govil, M. Saurakhia, P. Agnihotri, S. Shukla and S. Agarwal, "Analyzing the Behaviour of Applying Agile Methodologies & DevOps Culture in e-Commerce Web Application," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), Tirunelveli, India, 2020, pp. 899-902,
- (7) T. Capris, P. Melo, N. M. Garcia, I. M. Pires and E. Zdravevski, "Comparison of SQL and NoSQL databases with different workloads: MongoDB vs MySQL evaluation," 2022 International Conference on Data Analytics for Business and Industry (ICDABI), Sakhir, Bahrain, 2022, pp. 214-218, doi.

RENTMENT APPLICATION

(8) *K. Kishore, S. Khare, V. Uniyal and S. Verma, "Performance and stability Comparison of React and Flutter: Cross-platform Application Development," 2022 International Conference on Cyber Resilience (ICCR), Dubai, United Arab Emirates, 2022, pp. 1-4, doi: 10.1109.*