

ACKNOWLEDGEMENT

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. A special gratitude we give to our Internal Guide Prof. Rajul Suthar whose contribution in simulating suggestions and encouragement, helped us to coordinate our project especially in writing this report. We express our sincere gratitude to our H.O.D. Dr.Ashishkumar Patel for his support in our project completion. He provided us with immense knowledge. We are also thankful to our Principal Dr. Gargi Rajpara for providing us with great facilities for our project. We are also grateful to all the faculties for rendering great knowledge. This our great opportunity to complete this project under the guidance of all the faculties.

SATA MIHIR HIREN (22BECE30414)
PATEL LAXIT HAreshkumar (22BECE30296)
PATEL MAAN ROHITBHAI (22BECE30297)
PATEL RISHI VINODCHANDRA (22BECE30332)

ABSTRACT

The House Rental Management System is a comprehensive platform that streamlines the process of managing residential and commercial rental properties “**where renters and landlords connect with ease**”. It enables property owners to list their available properties, track rent payments, manage tenant details, and handle maintenance requests. The system also allows tenants to view available properties, sign lease agreements, make payments online, and submit maintenance issues. By automating key tasks such as payment reminders, contract renewals, and property inspections, the system reduces administrative burden and enhances operational efficiency for both landlords and property managers. Additionally, it provides a secure and transparent environment for tenants to interact with landlords. This system is designed to optimize property management, improve tenant satisfaction, and ensure timely financial transactions for all parties involved.

TABLE OF CONTENTS

NO.	NAME	PAGE NO.
1	Acknowledgement	i
2	Abstract	ii
3	Table of Contents	iii
4	List of Figures	vi
5	List of Tables	vii

NO.	NAME	PAGE NO.
1	Introduction	1
	1.1 Introduction	2
	1.2 Aim and Objective of the work	2
	1.2.1 Aims of the House Rental Management System	2
	1.2.2 Objectives of the House Rental Management System	3
	1.3 Brief Literature Review	4
	1.4 Problem definition	5
	1.5 Plan of their work	5
2	Technology And Literature Review	7
	2.1 Tools and Technology	8
	2.1.1 Node.js with Express.js	8
	2.1.2 HTML, CSS, and JavaScript	8
	2.1.3 MongoDB Database	8
	2.2 Project Planning	9
	2.2.1 Project Development Approach	9
	2.2.2 Milestones and Deliverables	10
	2.2.3 Group Dependencies	11

	2.3 Project Scheduling	12
	2.4 Estimation	12
	2.4.1 Effort Estimation	12
	2.4.2 Cost Estimation	13
3	System Requirement of Study	14
	3.1 User Characteristic	15
	3.2 Software and Hardware Requirement	16
	3.2.1 Software Requirements	16
	3.2.2 Hardware Requirements	17
	3.2.3 Functional Requirements	17
	3.2.4 Non-Functional Requirements	18
	3.3 Assumptions and Dependencies	18
	3.3.1 Assumptions	18
	3.3.2 Dependencies	19
4	System Diagrams	21
	4.1 Study of Current System	22
	4.2 Modules and Functionality of Purposed System	22
	4.3 Feasibility Study	23
	4.3.1 Technical Feasibility	24
	4.3.2 Operational Feasibility	24
	4.3.3 Economic Feasibility	25
	4.4 Requirements Validation	25
	4.5 Class Diagram	27
	4.6 Use Case Diagram	28
	4.7 Sequence Diagram	30
5	System Design	34
	5.1 Data Dictionary	35
	5.2 Entity Relation Diagram	37
	5.3 Data Flow Diagram	38

	5.3.1 Zero Level Diagram [Context Level]	38
	5.3.2 First Level Diagram	39
	5.3.3 Second Level Diagram	40
	5.4 Activity Diagram	41
	5.5 State Diagram	44
6	Result, Discussion and Conclusion	45
	6.1 Result	46
	6.2 Test Case	46
	6.2.1 Valid Test Case	46
	6.2.2 Non-Valid Test Case	47
	6.3 Snapshots	48
	6.3.1 Home Page	48
	6.3.2 Details and Review	50
	6.3.3 Create New Listing	51
	6.3.4 Error Detection	52
	6.4 Discussion	53
	6.5 Conclusion	55
7	References	57
	7.1 References	58

LIST OF FIGURES

NO.	NAME	PAGE NO.
1	Gantt chart	12
2	Class Diagram	27
3	Use Case Diagram	28
4	Sequence Diagram (Admin)	31
5	Sequence Diagram (Tenant)	32
6	Sequence Diagram (Landlord)	33
7	ER Diagram	37
8	Data Flow Diagram [LEVEL – 0]	38
9	Data Flow Diagram [LEVEL – 1]	39
10	Data Flow Diagram [LEVEL – 2]	40
11	Activity Diagram	41
12	State Diagram	44
13	Valid Test Case	46
14	Non-Valid Test Case	47
15	Homepage	48
16	Details and Review	50
17	Create New Listing	51
18	Error Detection	52

LIST OF TABLE

NO.	NAME	PAGE NO.
1	Month 1: Milestones & Deliverables	10
2	Month 2-4: Milestones & Deliverables	10
3	Month 5 to 7: Milestones & Deliverables	10
4	Month 8: Milestones & Deliverables	11
5	COCOMO Parameters for Semi-Detached	13
6	Data Dictionary	35

1. INTRODUCTION

- **Introduction**
- **Aims and Objective of the work**
- **Brief Literature Review**
- **Problem definition**
- **Plan of their work**

1.1 Introduction

The **House Rental Management System** is an innovative digital platform designed to transform the way rental properties are managed. This system aims to improve the efficiency, accessibility, and transparency of property listings, tenant management, and lease agreements. By leveraging modern technology, the system automates various tasks, reducing manual errors and ensuring a seamless experience for both tenants and property managers. The primary goal of this system is to provide a user-friendly online platform where tenants can easily browse listings, register, submit rental applications, and track payment schedules. This convenience encourages more people to find suitable rental properties and fosters a quicker turnaround for property owners.

For property managers and landlords, the system offers comprehensive tools to manage tenant information, track rental payments, handle maintenance requests, and monitor lease agreements. This streamlines property management tasks, ensuring that rental properties are maintained effectively and that tenant relationships are managed professionally.

Transparency is a key feature of the House Rental Management System. The platform provides real-time updates on rental availability, lease status, and payment history, helping both tenants and landlords stay informed and aligned on rental terms. Additionally, the system generates detailed reports and analytics on tenant trends, payment history, and property occupancy rates, enabling property managers to make informed decisions and optimize property usage. By implementing this system, rental agencies and individual landlords can significantly improve service quality, ensure timely rent collection, and enhance the overall tenant experience.

1.2 Aims and Objective of the work

1.2.1 Aims of the House Rental Management System:

- **Simplify Property Management:** To streamline and simplify the management of rental properties, including listing, monitoring, and tracking rental payments, maintenance requests, and tenant interactions.
- **Enhance Efficiency:** To automate and optimize the administrative and operational tasks involved in renting out properties, reducing manual work for landlords and property managers.
- **Improve Communication:** To provide a seamless communication channel between landlords and tenants, enabling easy resolution of issues and inquiries.
- **Ensure Transparency:** To promote transparency between all parties by offering clear and easily accessible information about properties, terms, agreements, and payments.
- **Reduce Operational Costs:** To reduce operational overheads for landlords and property managers by minimizing the need for paperwork, phone calls, and face-to-face interactions.

- **Enhance Tenant Satisfaction:** To improve the overall tenant experience by offering a smooth, user-friendly platform for managing rent payments, maintenance requests, and lease agreements.

1.2.2 Objectives of the House Rental Management System:

- **User Registration and Profile Management:** Allow landlords and tenants to create personalized accounts and manage their details, such as contact information, lease terms, property details, and payment history.
- **Property Listing:** Enable landlords to list their properties with relevant details such as rent amount, property size, location, amenities, and images.
- **Property Search and Filtering:** Allow tenants to search for available rental properties by applying filters based on location, price range, property type, and other preferences.
- **Lease Agreement Generation:** Automatically generate rental agreements based on tenant selection and landlord terms, with options for digital signatures for easy execution.
- **Online Rent Payment:** Provide a secure and convenient platform for tenants to pay rent online, with features such as automated payment reminders, recurring payments, and payment tracking.
- **Maintenance Request Management:** Facilitate tenants in submitting maintenance and repair requests, and allow landlords to track, manage, and address these issues in a timely manner.
- **Tenant and Landlord Communication:** Provide a secure messaging system for direct communication between tenants and landlords, enabling smooth discussions about the property, payments, or issues.
- **Property Availability and Calendar Management:** Enable landlords to update the availability status of their properties in real time, while tenants can check for available dates and schedule viewings.
- **Feedback and Rating System:** Implement a feedback system where tenants can leave ratings and reviews about properties, and landlords can review tenants based on their behaviour and payment history.
- **Automated Notifications:** Send timely reminders to both landlords and tenants about upcoming rent due dates, lease renewals, maintenance schedules, and other important events.
- **Data Security and Privacy:** Ensure that all user data (personal, payment, and property-related) is stored securely, with encryption and data protection measures in place.
- **Reporting and Analytics:** Provide landlords with detailed reports and analytics related to income, rent collection, property performance, and tenant interactions, enabling better decision-making.

1.3 Brief Literature Review

House rental management systems (HRMS) have emerged as essential tools for streamlining property management tasks. These systems facilitate various functions such as property listings, rent collection, lease agreements, maintenance requests, and communication between landlords and tenants, all of which enhance the efficiency of property management.

Key Features of HRMS platforms typically include features like:

- **Property Listings:** Landlords can create detailed property listings with descriptions, images, and pricing, which tenants can search through based on preferences.
- **Rent Collection and Tracking:** Automated systems allow for online rent payments, reminders, and financial tracking.
- **Lease Management:** Digital lease agreements, document storage, and lease renewal reminders help simplify the lease lifecycle.
- **Maintenance Requests:** Tenants can submit maintenance requests directly through the system, which landlords can track and resolve promptly.

Benefits

- **Efficiency:** Automation reduces manual effort, allowing property managers to focus on other critical tasks.
- **Transparency:** Clear documentation of payments, communications, and lease terms ensures transparency.
- **Improved Communication:** Direct messaging and automated notifications streamline landlord-tenant interaction.

Challenges

- **Adoption Barriers:** Older or less tech-savvy landlords may be reluctant to adopt HRMS platforms.
- **Data Security:** The storage of sensitive tenant and financial data requires robust security measures to prevent breaches.
- **Technical Issues:** System reliability and downtime could disrupt property management operations.

Future Trends

- **AI and Machine Learning:** Future HRMS platforms may use AI to predict rental pricing, optimize property recommendations, and automate tenant screening.
- **Blockchain Integration:** Blockchain could offer enhanced security and transparency through tamper-proof contracts and records.

1.4 Problem definition

In short, the core problem a house rental management system solves is the **inefficiency and complexity of traditional, manual methods of managing rental properties.**

This statement highlights the following key problems:

- **Inefficiencies:** Manual processes are inherently inefficient.
- **Disparate Systems:** Using multiple, unconnected tools (spreadsheets, paper files, etc.) creates data silos and hinders effective management.
- **Data Management Issues:** Manual record-keeping is prone to errors, loss, and difficulty in retrieval.
- **Communication Challenges:** Lack of a centralized communication platform leads to misunderstandings and delays.
- **Financial Tracking Difficulties:** Manual financial tracking increases the risk of errors and makes it difficult to generate accurate reports.
- **Accessibility Limitations:** Traditional methods often lack convenient access for both landlords and tenants.

1.5 Plan of their work

A work plan for developing a house rental management system would typically involve the following phases:

1. Requirements Gathering and Analysis:

- **Stakeholder Identification:** Defined key user groups (landlords, tenants, property managers).
- **Requirements Elicitation:** Conducted interviews, surveys, and workshops to gather user needs.
- **Requirements Documentation:** Created detailed documentation of functional and non-functional requirements.
- **Scope Definition:** Clearly defined system boundaries and features to be included.

2. System Design:

- **System Architecture Design:** Defined the overall structure and components (e.g., client-server, web-based).
- **Database Design:** Developed a relational database schema to store and manage data efficiently.
- **UI/UX Design:** Created wireframes, mock-ups, and prototypes for intuitive user interfaces.
- **Security Considerations:** Incorporated security measures into the design to protect sensitive data.

3. Development:

- **Development Methodology:** Selected an appropriate development methodology (e.g., Agile, Waterfall).
- **Technology Stack:** Defined the programming languages, frameworks, and tools to be used.
- **Module Development:** Developed core modules (e.g., tenant management, property management, rent collection, reporting).
- **Testing and Quality Assurance:** Implemented rigorous testing procedures (unit, integration, user acceptance testing).

4. Implementation:

- **Deployment Strategy:** Defined the deployment environment (e.g., cloud-based, on-premise).
- **Data Migration:** Planned and executed the migration of existing data to the new system.
- **User Training:** Developed training materials and conducted training sessions for users.
- **Go-Live and Launch:** Managed the system launch and provided initial support.

5. Maintenance and Support:

- **Maintenance Plan:** Established a plan for ongoing maintenance, bug fixes, and updates.
- **Technical Support:** Provided a mechanism for users to report issues and receive technical assistance.
- **Performance Monitoring:** Implemented monitoring tools to track system performance and identify potential issues.
- **Feedback Collection and Iteration:** Established a process for collecting user feedback and incorporating improvements.

Additional Key Points for the Report:

- **Project Timeline:** Include a high-level timeline for each phase.
- **Resource Allocation:** Briefly mention the resources (personnel, budget) allocated to the project.
- **Risk Management:** Briefly discuss potential risks and mitigation strategies.

2.TECHNOLOGY AND LITERATURE REVIEW

- **Tools and technology**
- **Project planning**
- **Project scheduling**
- **Cost estimation**

2.1 Tools and Technology

2.1.1 Node.js with Express.js:

Node.js with Express.js is used to build the House Rental Management System (HRMS). This combination allows for fast and scalable web applications. Node.js handles multiple requests efficiently, while Express.js simplifies building and routing the web pages. In the context of HRMS, Node.js with Express.js can:

- Authenticate users like tenants, landlords, and admins.
- Manage property listings, rental agreements, tenant information, and payments.
- Integrate with third-party services like payment processors and property listing platforms.

Using this technology, HRMS can run smoothly, providing users with a fast and responsive experience for managing properties and rentals.

2.1.2 HTML, CSS, and JavaScript:

HTML, CSS, and JavaScript are essential for building the front-end of the House Rental Management System (HRMS). These technologies provide a responsive and interactive user interface. Key benefits include:

- Dynamic property updates: Property availability and rental status can be shown in real-time without page reloads.
- Efficient data management: Tenant information, payment details, and rental history can be updated instantly.
- Instant notifications: Alerts for lease expirations or rent reminders are displayed without disrupting the user experience.

Using HTML for structure, CSS for design, and JavaScript for interactive features, HRMS ensures a smooth and user-friendly experience.

2.1.3 MongoDB Database:

MongoDB serves as the central data storage solution for the House Rental Management System (HRMS), offering a flexible and scalable NoSQL environment to manage rental data. Key features of using MongoDB in this context include:

- **Data Management:** MongoDB allows for the storage of tenant information, property details, lease agreements, payment histories, and maintenance records in a flexible, document-oriented format, making it easy to store and retrieve related data.
- **Performance and Scalability:** MongoDB's horizontal scaling capabilities enable the HRMS to handle large volumes of data and traffic efficiently. It ensures the system can grow as the number of properties, tenants, and transactions increases.

- **Aggregation and Querying:** MongoDB's powerful aggregation framework and querying capabilities allow for complex data operations, such as generating rental reports, calculating late fees, or tracking payment histories, with high performance.

Additionally, MongoDB's **replication** and **sharding** features provide data redundancy and availability, ensuring that critical rental information remains accessible and recoverable even during system failures.

2.2 Project Planning

2.2.1 Project Development Approach

For the development of the House Rental Management System (HRMS), we have chosen the **Incremental Model**. This approach combines aspects of the waterfall model, enabling iterative development. Each phase is completed incrementally, allowing for continuous feedback and refinement as the project progresses.

The phases of development are as follows:

1. **Communication:** Initial discussions with stakeholders (landlords, tenants, property managers) to gather requirements and define system needs.
2. **Planning:** Creating a detailed development plan, allocating resources, and setting timelines for each phase.
3. **Modelling:** Designing the system architecture, user interfaces, and database structure for managing properties, tenants, payments, and notifications.
4. **Construction:** Developing and coding the core features such as property listings, rental agreements, user authentication, and payment processing.
5. **Deployment:** Launching the system, gathering feedback from users, and making necessary improvements based on their input.

This incremental approach is advantageous because it allows for adjustments based on user feedback after each phase, ensuring the system meets the evolving needs of users while avoiding the limitations of the waterfall model, where changes later in development can be difficult to implement.

2.2.2 Milestones and Deliverables

Month 1: Initial Planning & Design

Milestones	Deliverables
Gather stakeholder requirements for the House Rental Management System	Requirements Documentation
Define system features and functionalities for property listings, tenant management, and payment processing	Functional Specifications
Conduct interviews, surveys, and market research to refine system requirements	Requirements Gathering Report
Design initial wireframes for user interfaces (Tenant, Landlord, Admin)	UI Wireframes and Prototypes

Month 2-4: Core Development & System Architecture

Milestones	Deliverables
Finalize system architecture and technology stack (Node.js, MongoDB, HTML/CSS/JS)	System Architecture Document
Complete database schema design for properties, tenants, payments, etc.	Database Schema Document
Develop the back-end (server-side logic for rental management, user authentication)	Back-end Development Code
Implement front-end interfaces for Tenant and Landlord portals	Front-end Development Code (UI for Tenant & Landlord)
Integration of back-end and front-end systems for seamless functionality	Integrated Application (Back-end & Front-end)

Month 5-7: Module Development & Integration

Milestones	Deliverables
Develop Admin Panel for property and tenant management	Admin Panel Code and User Guide
Create Tenant Module for viewing properties, submitting rental requests, and payment history	Tenant Module Code and User Guide
Develop Landlord Module for property listing management and tenant interaction	Landlord Module Code and User Guide

Implement payment gateway for rent processing and tracking	Payment Module Code and User Guide
Perform initial integration and unit testing	Integration and Unit Testing Report

Month 8: Testing, Deployment, and Final Adjustments

Milestones	Deliverables
Conduct comprehensive system testing, including functional, integration, and performance testing	Comprehensive Testing Report (Test Cases & Results)
Perform user acceptance testing (UAT) with stakeholders (Landlords, Tenants, Admins)	User Acceptance Testing (UAT) Report
Address bugs, errors, and implement feedback from UAT testing	Bug Fixes and Updated Code
Finalize system deployment, including setting up server and database for live environment	Deployed HRMS System (Live Version)
Provide training materials for end-users (tenants, landlords, admins)	User Manuals and Training Documents

2.2.3 Group Dependencies

For the House Rental Management System (HRMS) project, our team works together in a cooperative way, with no single leader. We make decisions as a group, and everyone shares their ideas.

Key points of our team organization include:

- **Teamwork:** Everyone on the team is involved in discussions and decision-making, making sure all opinions are heard.
- **Problem-Solving Together:** When problems come up, we solve them as a team. We hold regular meetings where everyone can share their ideas and help find solutions.
- **Seeking Expert Advice:** We also consult with outside experts to get advice and make sure we're on the right track.

2.3 Project Scheduling

Project Scheduling for House Rental Management System (HRMS) involves organizing tasks and estimating time for project activities.

- **Gantt Chart:** A visual timeline that shows when each task will start and finish, who is responsible, and how tasks overlap. It helps track progress and ensures timely project completion.

This structured approach helps manage the project efficiently and meet deadlines for the HRMS project.



Fig 2.1 Gantt chart

2.4 Estimation

2.4.1 Effort Estimation

Effort Estimation for House Rental Management System (HRMS) involves determining the time and resources needed to complete the project. The Gantt chart helps outline each phase of the project, like design, development, testing, and deployment, ensuring tasks are completed on schedule. It estimates the labour required for each task but does not include manpower costs. This helps allocate resources efficiently and keep the project on track.

2.4.2 Cost Estimation

Cost estimation involves determining the overall project cost, which is influenced by various factors like project size and complexity. For this, we will use the COCOMO (Constructive Cost Model) heuristic technique. The project is categorized based on its complexity, which influences the effort and cost estimation.

COCOMO Complexity Categories:

- Organic
- Semi-Detached
- Embedded

Given the HRMS project will involve moderate complexity, such as integrating payment systems and managing multiple users, it is categorized as Semi-Detached.

COCOMO Parameters for Semi-Detached:

Software Project	AB	BB	CB	DB	Select
Organic	2.5	1.10	2.4	0.40	
Semi-Detached	3.5	1.15	2.6	0.30	✓
Embedded	4.0	1.25	2.8	0.28	

Example Calculation:

Assuming the estimated size of the **HRMS** is **8,500 KLOC (Thousands of Lines of Code)**, the cost estimation steps are as follows:

1. Effort (PM):

The effort required for the project is calculated as:

$$E=3.5 \times (8.500) 1.15 \approx 38.21 \text{ Person-Months (PM)}$$

2. Development Time (Tdev):

The development time is estimated as:

$$T_{dev}=2.6 \times (38.21) 0.30 \approx 9.12 \text{ months}$$

3. Cost Estimation: Assuming the monthly salary for a trainee developer is **10,000 Rupees**, the total cost for development would be:

$$\text{Total Cost} = 9.12 \times 10,000 \approx 91,200 \text{ Rupees}$$

3. SYSTEM REQUIREMENT OF STUDY

- **User Characteristic**
- **Software and Hardware Requirement**
- **Assumptions and Dependencies**

3.1 User Characteristic

Here are the key user characteristics of a House Rental Management System (HRMS):

1. Renters/Tenants:

Registration: Ability to create and manage personal profiles, including contact details, lease agreements, and payment preferences.

Eligibility Check: View the eligibility criteria for renting properties, including credit checks or rental history.

Lease History: Access past rental records, payment history, and upcoming lease renewal reminders.

Property Search: Ability to search for available properties based on preferences like price, location, size, and amenities.

Payment Management: Make rental payments, view invoices, and track payment status (e.g., paid, due, overdue).

2. Landlords/Property Owners:

Property Listings: Create, manage, and update property listings, including photos, rent prices, availability, and lease terms.

Tenant Management: View and manage tenant details, lease agreements, and communication history.

Rent Collection: Track rent payments, issue reminders for overdue rent, and generate payment receipts.

Maintenance Requests: Receive and manage maintenance requests from tenants, track their status, and assign tasks to service teams.

Reporting Tools: Generate reports related to rental income, tenant performance, property maintenance, and more.

3. Property Managers:

Tenant Request Management: Handle tenant issues, concerns, or requests (e.g., maintenance, complaints).

Lease Management: Oversee lease agreements, renewals, and terminations.

Property Inspections: Schedule and track property inspections, documenting property condition before and after tenancy.

Payment and Financial Tracking: Monitor rent payments, due dates, and financial transactions related to properties.

4. Administrators:

User Management: Control user access levels and permissions (for landlords, tenants, property managers).

System Configuration: Manage system settings, including property types, payment gateways, and automated reminders.

Compliance Management: Ensure the system complies with rental laws, regulations, and safety standards.

Audit Trails: Monitor system activity for security, compliance, and data integrity purposes.

5. Regulatory Authorities:

Compliance Monitoring: Access reports and data to ensure that rental practices comply with housing regulations, fair rental practices, and safety standards.

Data Analysis: Review overall system performance, tenant-landlord disputes, market trends, and rental prices to ensure proper regulation.

6. Real Estate Agents:

Property Listings Management: List properties on behalf of landlords, manage promotional content, and facilitate viewings.

Tenant Screening: Assist in screening potential tenants, including conducting background checks, credit scoring, and reference verification.

Lease Negotiations: Assist in negotiations between tenants and landlords, helping with the drafting of lease agreements.

3.2 Software and Hardware Requirement

3.2.1 Software Requirements

Client Requirements:

- Operating System: Windows, macOS, or Linux.
- Web Browser: Any modern browser (Chrome, Firefox, Safari, Edge).
- Mobile Support: Responsive web design for mobile and tablet access.

Developer Requirements:

- Operating System: Windows Server or Linux Server.
- Database Server: MongoDB, PostgreSQL or MySQL
- Technologies:
 - Backend: Node.js with Express.js (or ASP.NET with C#).
 - Frontend: HTML, CSS, JS

3.2.2 Hardware Requirements

Client :-

- Smart phone , Personal Computer
- RAM:- 4 GB
- Internet Connectivity
- Minimum Storage Space

Developer :-

- Computer :- minimum intel i5
- RAM : - 8 GB
- OS :- windows 10 or later
- Storage :- SSD 256 GB
- Internet connectivity
- External storage for data back-up and recovery

3.2.3 Functional Requirements

- **Tenant Management:** Register and maintain tenant details (contact information, lease agreements, payment history).
- **Property Listings Management:** Allow landlords to list and update property details, including rent prices, availability, and amenities.
- **Lease Agreement Management:** Record and manage lease agreements between landlords and tenants, including start/end dates and payment terms.
- **Payment Processing:** Manage and track rental payments, generate invoices, and send reminders for overdue payments.
- **Maintenance Request Handling:** Allow tenants to submit maintenance requests, and for property managers to track and resolve them.

- **Request Handling for New Properties:** Allow tenants to submit property requests based on their preferences, and for landlords to respond or update availability.
- **Real-Time Availability Tracking:** Monitor and update real-time availability of rental properties.
- **Reporting:** Generate reports on tenant payment history, property occupancy, maintenance requests, and overall rental performance.

3.2.4 Non-Functional Requirements

- **Performance:** The system should handle multiple simultaneous users (tenants, landlords, property managers) without significant delays, ensuring fast response times for property searches, payment processing, and data retrieval.
- **Reliability:** The system must be dependable, with minimal downtime. Backup and recovery mechanisms should be in place to prevent data loss (e.g., tenant details, property listings, payment records).
- **Scalability:** The system should be able to scale as the number of tenants, landlords, and properties increases without affecting system performance or response times.
- **Security:** Sensitive tenant and landlord information should be protected through encryption, secure authentication methods and strict access controls to prevent unauthorized access.
- **Usability:** The system should have an intuitive, user-friendly interface, allowing tenants to search for properties, landlords to manage listings, and property managers to process maintenance requests and payments without requiring extensive training.
- **Maintainability:** The system should be designed with modular architecture, clear documentation, and easy-to-maintain code to allow for quick troubleshooting and easy integration of future enhancements or updates.
- **Availability:** Ensure high system availability, especially during peak times (e.g., when tenants are looking for properties or making payments). Downtime should be minimized, and failover measures should be in place to ensure continuous operation.

3.3 Assumptions and Dependencies

3.3.1 Assumptions

- **Internet & Device Access:** Tenants, landlords, and property managers must have access to a reliable internet connection and a compatible device (smartphone, tablet, or computer) to use the online house rental management system.
- **Basic Digital Skills:** All users, including tenants, landlords, and property managers, should possess basic digital skills to effectively navigate the system. This includes tasks like creating accounts, browsing property listings, managing rental agreements, and making payments.
- **Booking and Viewing Properties:** Tenants must schedule property viewings in advance to ensure the availability of the property manager or landlord. The system

should allow tenants to select convenient time slots for viewings based on the landlord's or property manager's availability.

- **Staff Availability:** Property managers and landlords must regularly update their availability in the system to reflect when they can accommodate property viewings, handle inquiries, and finalize lease agreements. This ensures tenants can select times when they are available.
- **Eligibility for Renting:** Tenants may need to meet specific criteria (such as a credit check or rental history) and provide necessary documents (e.g., proof of income, references) during the application process to be eligible for renting a property.
- **Valid Identification:** Tenants will need to present a valid form of identification (e.g., government-issued ID) when signing lease agreements to verify their identity and eligibility for renting the property.
- **Data Security:** The system must implement strong data protection measures to secure personal and financial information of tenants, landlords, and property managers. This includes complying with relevant data protection laws to prevent unauthorized access and misuse of sensitive data.

3.3.2 Dependencies

- **Government IT Infrastructure:** The system will rely on existing government IT infrastructure, including servers and security protocols, to ensure reliable operation and data protection. This will help ensure compliance with local regulations and ensure smooth functioning.
- **Interoperability with Property Management Systems:** The house rental management system must integrate with other property management platforms and databases to verify tenant and landlord identities, track property availability, and manage rental agreements in real-time for better resource management.
- **Cloud Storage and Services:** The system may utilize cloud services for data storage, backup, and scalability. This will enable flexible data management, ensure data accessibility, and accommodate a growing user base as more properties and tenants are managed on the platform.
- **Staff Availability:** The efficiency of the system will depend on property managers and landlords regularly updating their availability and managing rental listings. This ensures tenants can book property viewings or schedule appointments at times when staff are available to assist them.
- **Notification Services:** The system requires integration with external notification services (SMS, email, or push notifications) to communicate with tenants, landlords, and property managers about viewing confirmations, rental payments, contract updates, and other important notifications.
- **Identity Verification Services:** The system will depend on external identity verification services to confirm the identities of tenants and landlords during the lease signing process (e.g., government-issued IDs, biometric verification, or national ID systems).
- **Tenant Recruitment and Awareness Programs:** Successful operation of the system will depend on ongoing tenant recruitment initiatives and marketing efforts to attract

potential tenants and property owners, ensuring the platform remains active and maintains a steady demand for rental properties.

4. SYSTEM DIAGRAMS

- **Study of Current System**
- **Modules and Functionality of Proposed System**
- **Feasibility Study**
- **Requirement Validation**
- **Class Diagram**
- **Use Case Diagram**
- **Sequence Diagram**

4.1 Study of Current System

The current house rental management system uses various technologies to manage rental properties, tenant information, and lease agreements. Modern systems automate tasks like property listings, rent collection, and maintenance requests to reduce errors and make the process more efficient. Features such as online payments, digital contract signing, and automated reminders have made things easier for both tenants and landlords.

However, there are still challenges. Some landlords and property managers continue to use manual methods, which can be slow and error-prone. Even automated systems can face issues, such as not working well with other tools like payment systems or local databases, which can lead to data gaps. Additionally, during events like the COVID-19 pandemic, there were disruptions in rent collection, vacant properties, and virtual property viewings. These issues show the need for ongoing improvements to make the system more reliable and adaptable for everyone involved.

4.2 Modules and Functionality of Proposed System

1. Tenant Management Module Functionality:

- Register new tenants and landlords.
- Track tenant history (rental history, payment records, and lease agreements).
- Set eligibility criteria for renting properties (income, background check, etc.).
- Schedule property viewings or meetings.
- Notify tenants about upcoming rent due dates, property viewings, or lease renewals (SMS/email alerts).

2. Property Management Functionality:

- Manage property listings (property type, location, price, availability).
- Real-time updates of property availability.
- Categorize properties by type, location, price range, and amenities.
- Issue alerts for upcoming maintenance or inspections.
- Handle property maintenance requests (repairs, cleaning, etc.).

3. Rental Request Management Functionality:

- Manage rental requests from potential tenants.
- Verify requests based on property availability and tenant eligibility.
- Track rental agreement fulfilment (lease start date, tenant details, landlord).
- Maintain records of signed lease agreements (property type, rent amount, lease terms).

4. Tenant Screening and Background Check Module Functionality:

- Track background check results (credit score, rental history, criminal background).
- Record tenant eligibility based on screening results.
- Automate rejection of unsuitable tenants.

- Maintain compliance with local rental laws and regulations.

5. User Management and Access Control Functionality:

- Admin and staff management (roles and permissions).
- Secure login for various user roles (admin, tenant, landlord, property manager).
- Restrict access to sensitive information based on user roles.
- Audit trails to track user actions in the system.

6. Notifications and Alerts Functionality:

- Send SMS/email reminders for upcoming rent payments.
- Issue alerts for low property availability or approaching lease expiry.
- Notify tenants about new properties or viewing opportunities.
- Emergency alerts for critical maintenance issues or urgent property matters.

4.3 Feasibility Study

- A Feasibility Study for a House Rental Management System (HRMS) is essential to determine whether the proposed system is practical, cost-effective, and beneficial for all stakeholders involved, such as tenants, landlords, property managers, and real estate companies. This study evaluates various aspects of the project, including technical, economic, operational, and legal factors, as well as additional considerations unique to property management. Below is a comprehensive framework to guide you through conducting a feasibility study for the HRMS:

The feasibility study consists of the following three aspects:

1. Technical Feasibility:

- Assess the system's technical requirements, including software, hardware, and security measures.
- Ensure it is compatible with various devices and can handle multiple users and transactions.

2. Operational Feasibility:

- Evaluate if the system is user-friendly and helps streamline tasks like property management, rent collection, and tenant screening.
- Ensure it can scale as the business grows.

3. Economic Feasibility:

- Compare the system's development and maintenance costs with potential benefits (e.g., improved efficiency, revenue generation).
- Assess if the project is financially viable and if it will provide a return on investment (ROI).

4.3.1 Technical Feasibility:

Front-End and Back-End Technologies:

Front-End: The application will use HTML, CSS, and JavaScript for building an intuitive user interface, ensuring that tenants and landlords can easily navigate the system.

Back-End: The back-end will be supported by technologies like MongoDB for database management, ensuring efficient storage and retrieval of property listings, tenant details, and lease information.

Cross-Browser Compatibility:

- The system will be developed using web technologies that ensure compatibility with popular browsers such as Chrome, Firefox, Safari, Opera, and Internet Explorer. This ensures users can access the platform from different devices and browsers.

Database Integration:

- MySQL will be used for managing the database, ensuring a secure, scalable, and reliable solution for storing and retrieving property details, rent payments, tenant records, and lease agreements.

Performance and Usability:

- The system will be optimized for fast performance, ensuring quick responses to user queries, such as property search and payment processing.
- The platform will be designed to be user-friendly and responsive, ensuring that both tenants and landlords have an easy, seamless experience. It will work efficiently across devices and provide reliable access to all functionalities.

4.3.2 Operational Feasibility:

The main goal of assessing **Operational Feasibility** is to ensure that the **House Rental Management System (HRMS)** will function effectively and meet the needs of its users after development and installation.

Outcomes of Operational Feasibility:

- The system will manage and track rental properties, tenant details, rental payments, and lease agreements, ensuring accurate records are maintained.
- Tenants and landlords can access the system remotely via email or mobile applications, providing real-time updates, property availability, payment notifications, and maintenance requests.
- The HRMS is designed to enhance operational efficiency, simplify property management tasks, streamline the rental process, and improve communication between tenants, landlords, and property managers.

Thus, this **House Rental Management System** is operationally feasible and will support seamless operations post-launch.

4.3.3 Economic Feasibility:

The goal of evaluating **Economic Feasibility** is to determine whether the financial investment in the **House Rental Management System (HRMS)** is justified by the expected benefits and aligns with the organization's financial capabilities.

Outcomes of Economic Feasibility:

- The HRMS will be developed as a web application, accessible via the internet, reducing the need for physical infrastructure and ensuring broad accessibility.
- Technologies such as HTML, CSS, and MongoDB will be used for the front end and back end, which are cost-effective and widely supported.
- The system's benefits—such as streamlining property management, automating rent collection, and improving tenant-landlord communication—will lead to significant cost savings and operational efficiency.
- By reducing manual effort, improving tenant retention, and streamlining administrative tasks, the system will generate value that outweighs the initial development and maintenance costs, making it a financially sound investment.

4.4 Requirements Validation

Basic Validation from User Side:

1. User Authentication:

- Users (e.g., tenants, landlords, property managers) must enter a valid username and password to access the system.
- If the username and/or password fields are left blank, the user will be denied access and prompted to complete the required fields.

2. Property Listing Form Submission:

- When landlords list a property for rent, all required fields (such as property details, price, location, and availability) must be filled out before submission.
- If any required field is left blank, the system will prevent the form from being submitted and prompt the user to fill in the missing information.

3. Error Messaging:

- If a user enters incorrect data (e.g., an invalid email address or mismatched rent prices), the system will generate an error message indicating the specific issue.
- Error messages will be clear and actionable, guiding users on how to correct the errors (e.g., "Please enter a valid email address").

4. Database Interaction:

- After successful validation of the username and password, users will be allowed to view or edit their information in the database, such as property listings, tenant details, and rent payment history.

5. Maintenance Request Submission:

- When tenants submit maintenance requests, all required fields (e.g., issue description, urgency level) must be filled.
- If any mandatory field is left blank, the system will block the request submission and provide feedback to the user to complete the required fields.

4.5 Class Diagram

A class diagram for a house rental management system visually represents the system's structure by showing classes, their attributes (data), and their relationships.

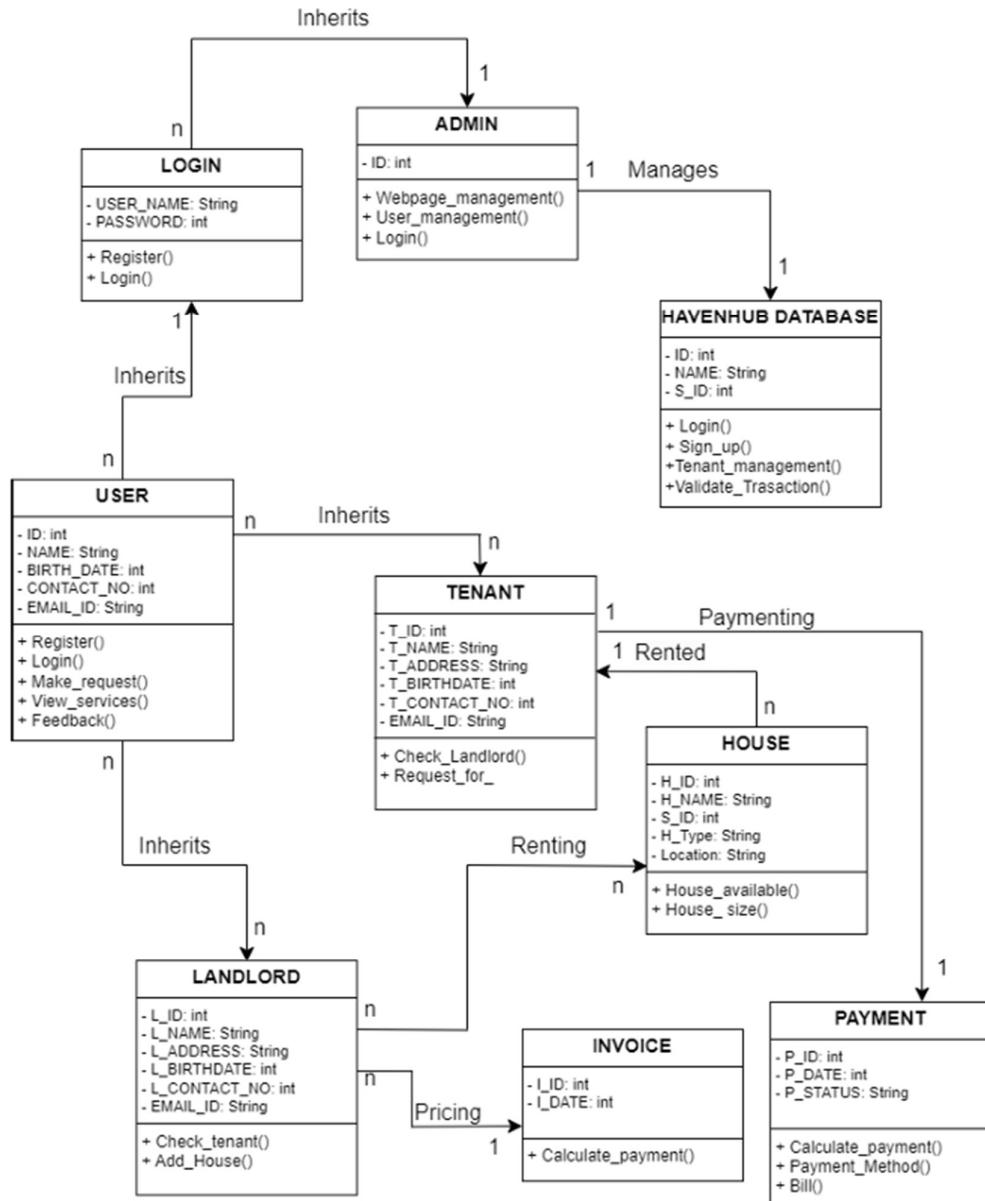


Fig 4.1 Class Diagram

4.6 Use Case Diagram

A use case diagram for a house rental management system illustrates the interactions between users (actors) and the system. It shows what the system does from the user's perspective, without detailing how it does it.

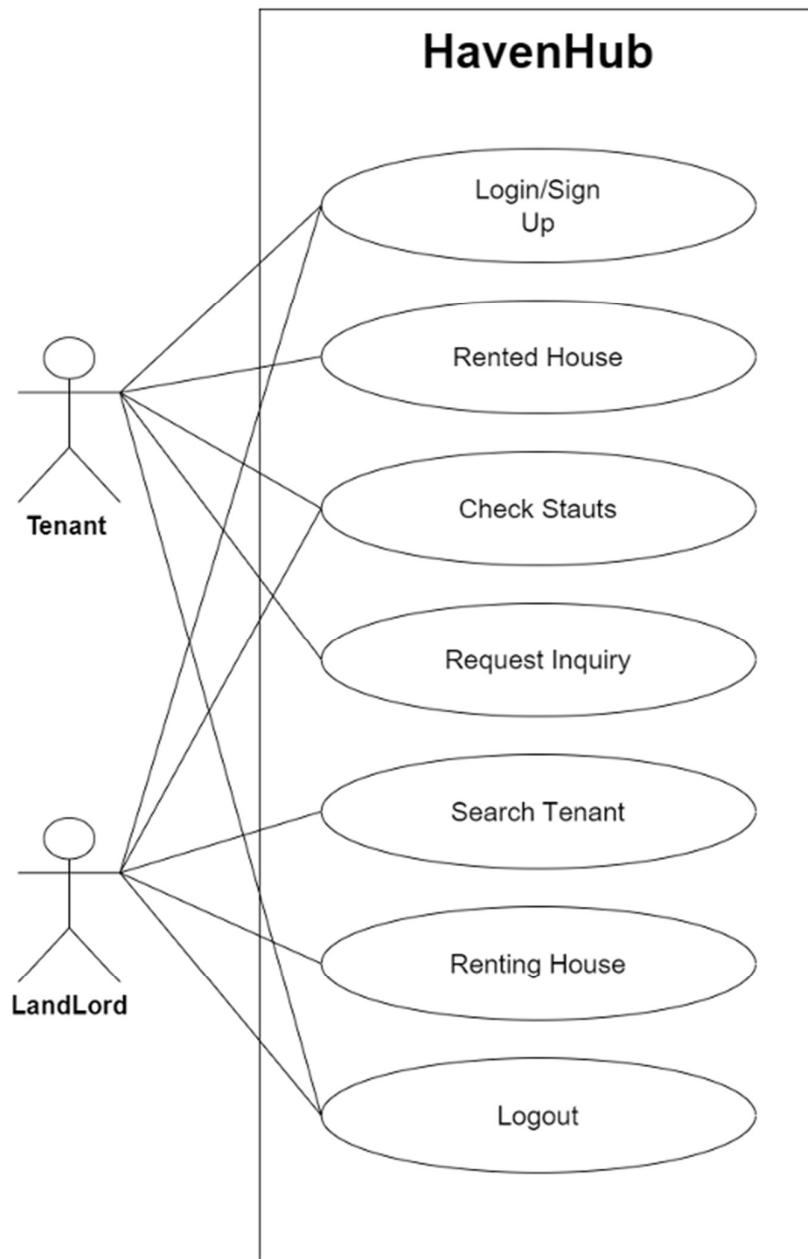


Fig 4.2 Use Case Diagram

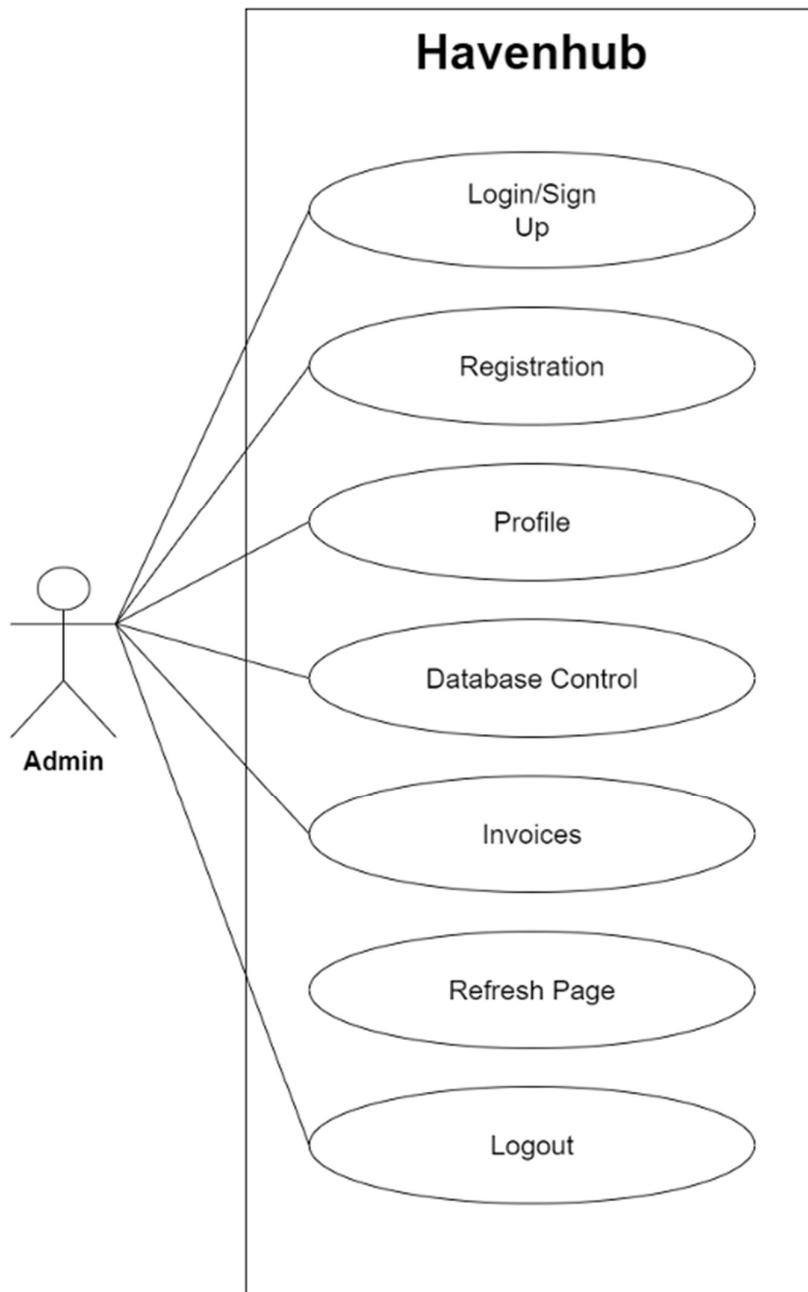


Fig 4.3 Use Case Diagram

4.7 Sequence Diagram

A sequence diagram in a house rental management system illustrates the interactions between objects (instances of classes) in a time-ordered sequence. It focuses on the flow of messages and the order in which events occur for a specific use case.

Sequence diagrams are useful for understanding the dynamic behaviour of the system and how objects interact to accomplish specific tasks. They are particularly valuable for developers during the design and implementation phases. In a report, they provide a visual representation of how different parts of the system work together.

In a typical interaction where a tenant schedules a property viewing, the sequence begins with the tenant requesting to schedule an appointment. The system checks available time slots in the database and returns this information to the tenant. After the tenant selects a time slot and confirms their choice, the system updates the database and sends a confirmation notification back to the tenant.

This summary encapsulates the key elements of the sequence diagram, emphasizing its role in enhancing understanding of user-system interactions within the HRMS.

Admin :

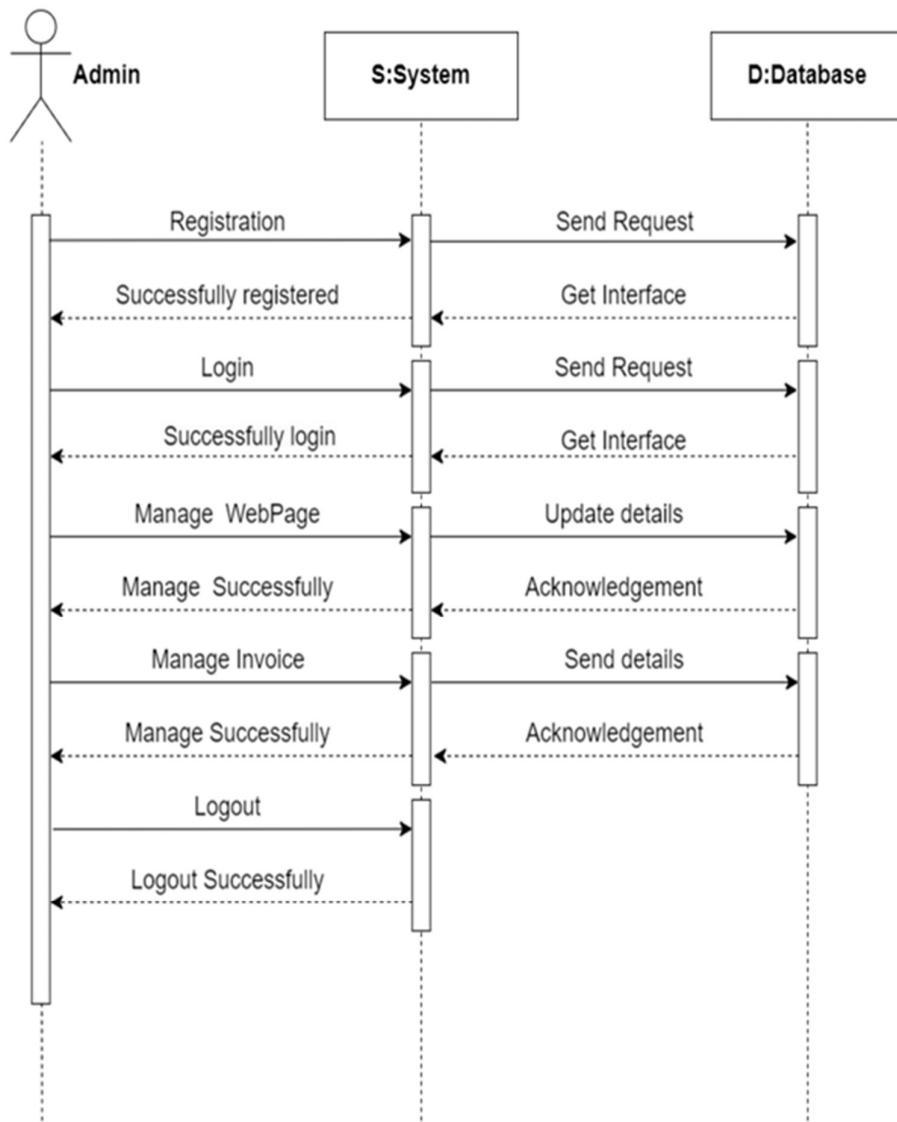


Fig 4.4 Sequence Diagram (Admin)

Tenant :

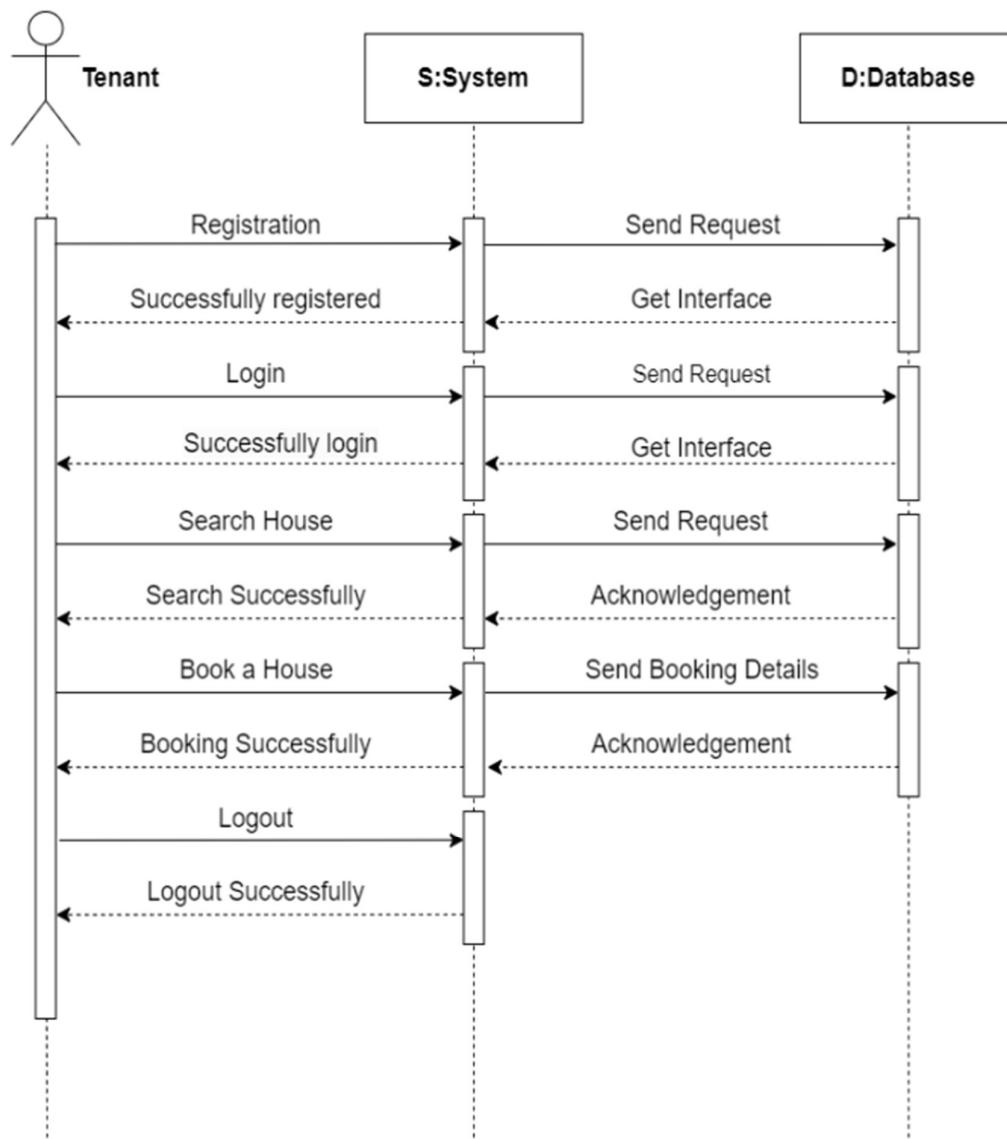


Fig 4.5 Sequence Diagram (Tenant)

LandLord :

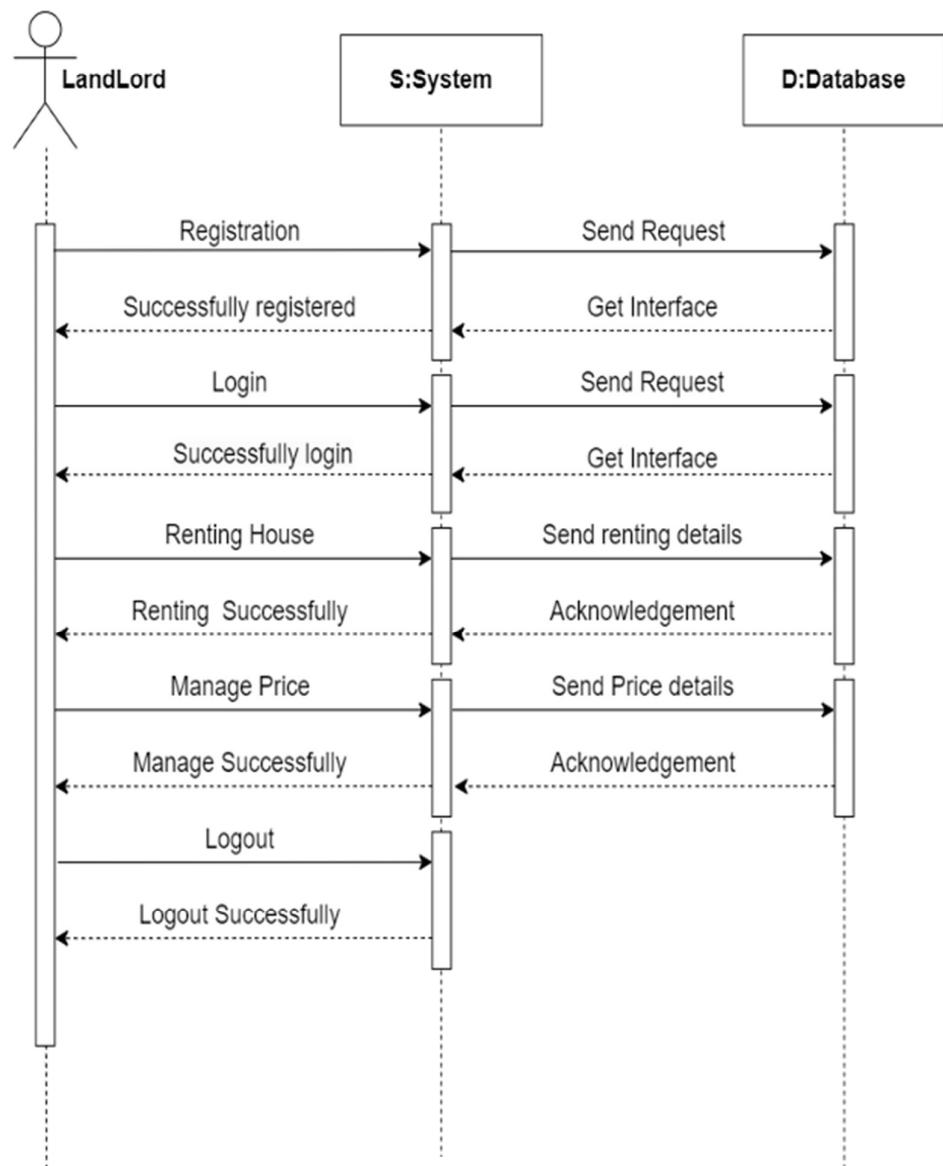


Fig 4.6 Sequence Diagram (LandLord)

5. SYSTEM DESIGN

- **Data Dictionary**
- **Entity Relation Diagram**
- **Data Flow Diagram**
- **Activity Diagram**
- **State Diagram**

5.1 Data Dictionary

A **data dictionary** for a **House rental management system** is a central repository of information about the data used in the system. It defines the meaning, data type, format, and other characteristics of each data element. This ensures consistency and clarity across the system.

A **data dictionary** for a **House Rental Management System** helps ensure that all stakeholders (developers, administrators, and users) understand how data is structured, stored, and accessed. It contains detailed descriptions of each data element, table, and their relationships, ensuring consistency and clarity in how data is used across the system. By providing this centralized reference, the system can be more easily managed, updated, and scaled as needed.

Table Name : Admin

ADMIN			
ColumnName	DataType	Size	Constraint
Admin_id	Int()	-	Primary Key
AdminName	Varchar()	100	-
Contact No.	Int()	-	-
Email_id	Varchar()	100	Primary Key

Table Name : Registration

REGISTRATION			
ColumnName	DataType	Size	Constraint
Reg_id	Int()	-	Primary Key
User_id	Varchar()	100	Foreign Key
Contact No.	Int()	-	-
Email_ID	Varchar()	100	Primary Key
Pin-code	Int()	-	-
Password	Varchar()	10	-

Table Name : User

USER			
ColumnName	DataType	Size	Constraint
UserID	Int()	-	Primary Key
UserName	Varchar()	100	-
Contact No.	Int()	-	-
Email_id	Varchar()	100	Primary Key
Password	Int()	10	-

Fig 5.1 Data Dictionary

Table Name : Tenant

TENANT			
ColumnName	DataType	Size	Constraint
Tenant_id	Int()	-	Primary Key
T_Name	Varchar()	120	-
Contact No.	Int()	-	-
Reg_id	Int()	100	Foreign Key
DOB	Int()	-	-

Table Name : LandLord

LANDLORD			
ColumnName	DataType	Size	Constraint
L_id	Int()	-	Primary Key
L_Name	Varchar()	120	-
Contact No.	Int()	-	-
DOB	Int()	-	-
Price	Int()	-	-

Table Name : Feedback

FEEDBACK			
ColumnName	DataType	Size	Constraint
F_id	Int()	-	Primary Key
Name	Varchar()	100	-
Feedback	Int()	100	-

Fig 5.2 Data Dictionary

5.2 Entity Relation Diagram

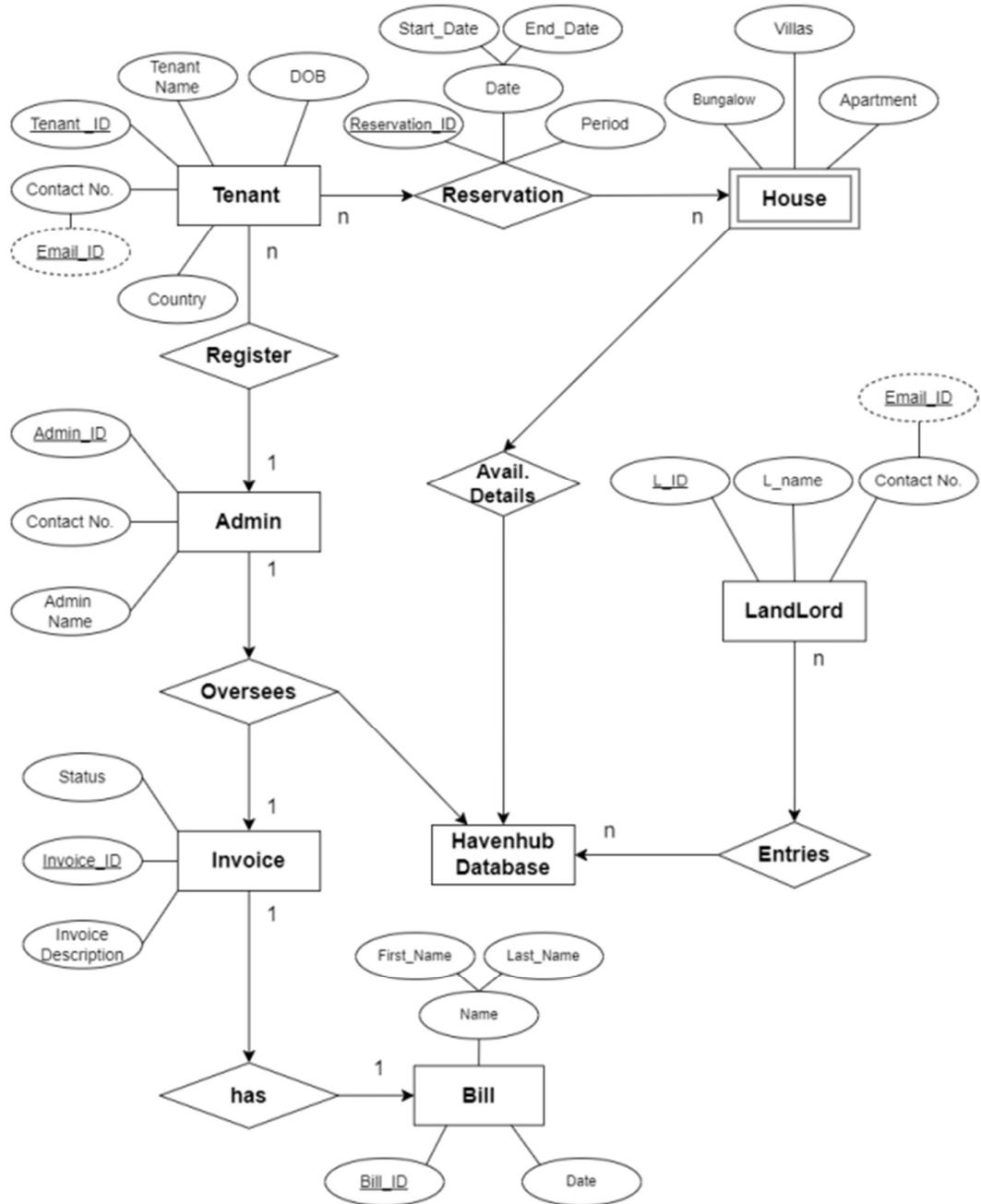


Fig 5.3 ER Diagram

5.3 Data Flow Diagram

5.3.1 Zero Level Diagram [Context Level]

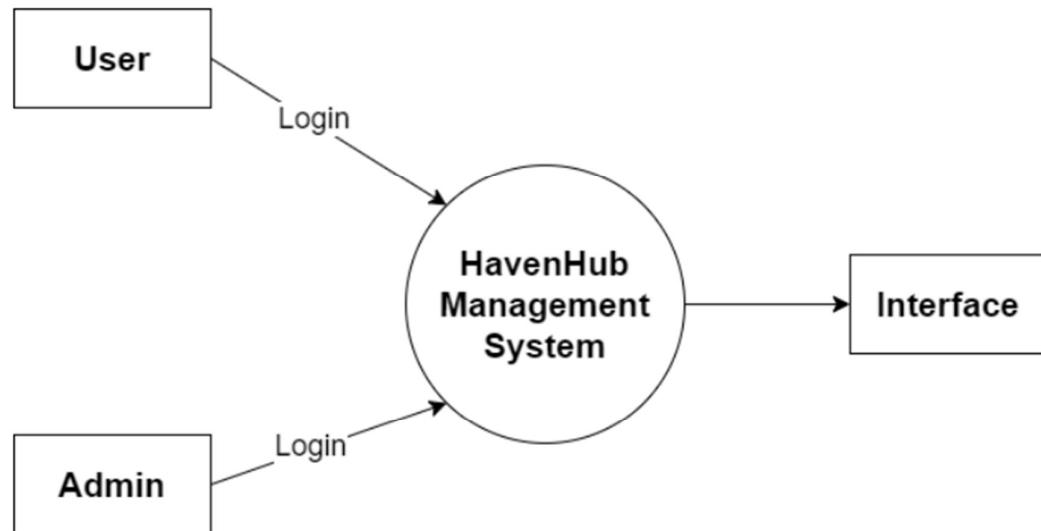


Fig 5.4 Data Flow Diagram [LEVEL – 0]

5.3.2 First Level Diagram

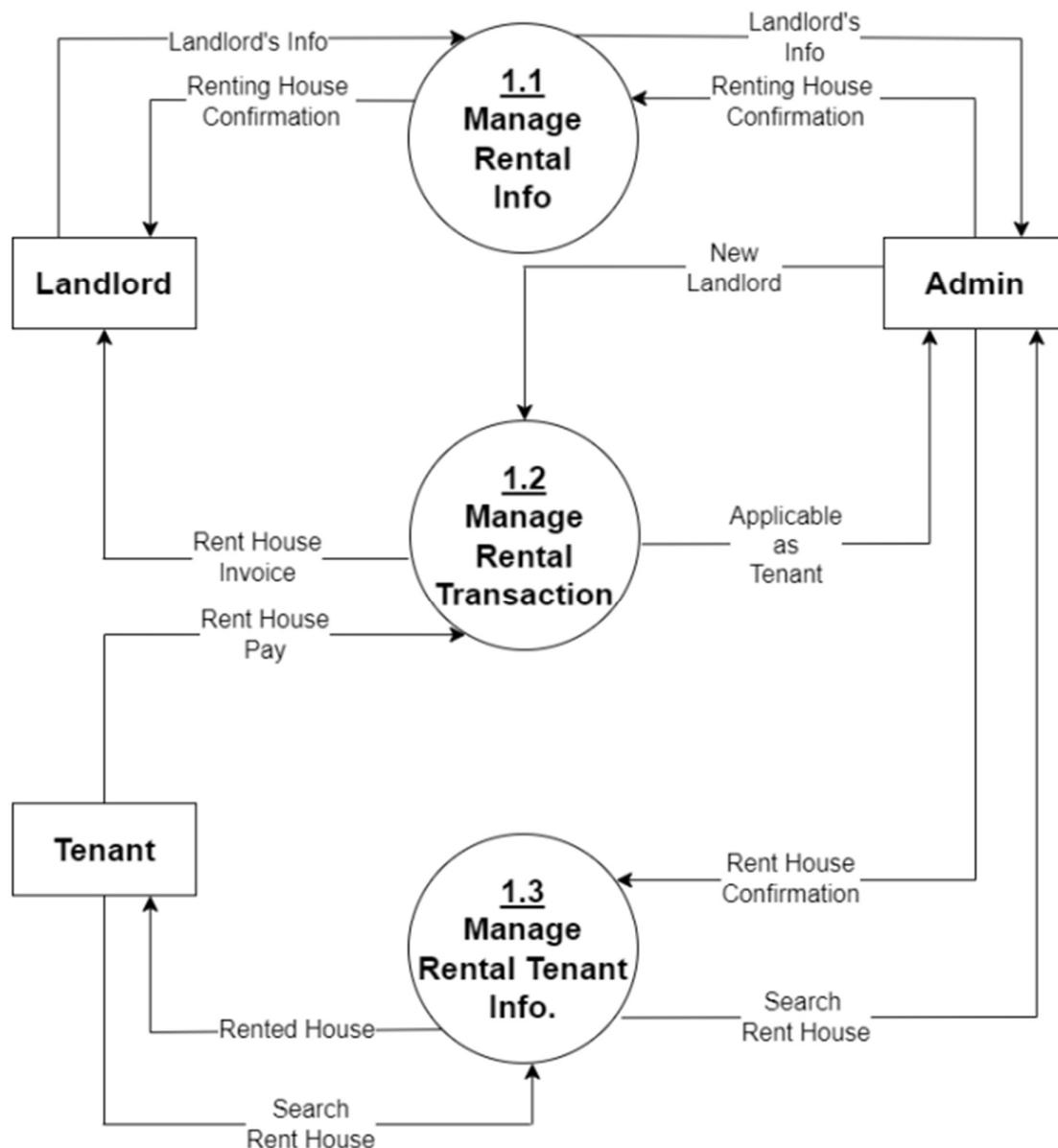


Fig 5.5 Data Flow Diagram [LEVEL – 1]

5.3.3 Second Level Diagram

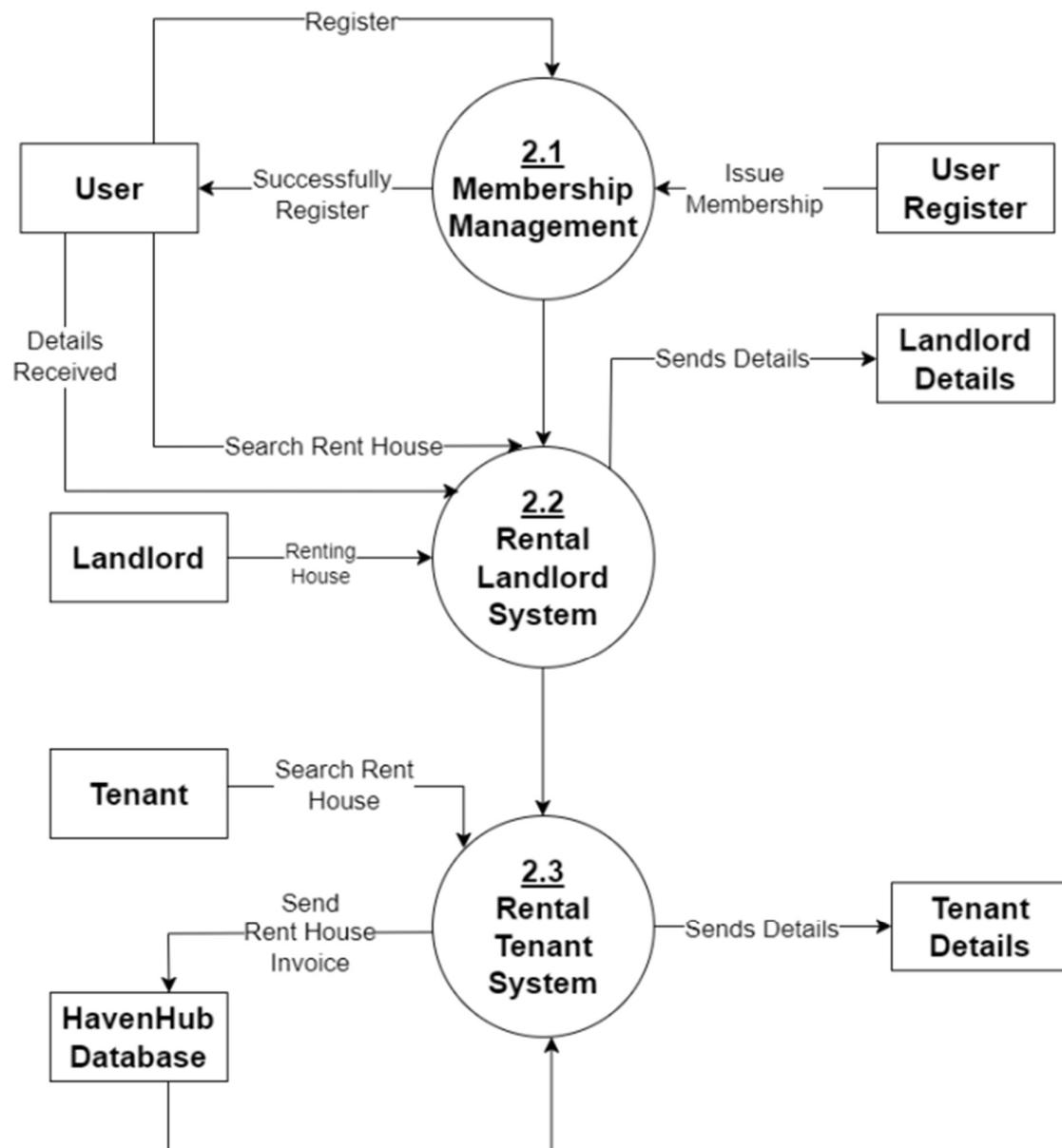


Fig 5.6 Data Flow Diagram [LEVEL – 2]

5.4 Activity Diagram

An Activity Diagram for a House Rental Management System is a visual representation of the workflow and processes involved in managing the rental of properties through an online platform. It illustrates the sequence of actions or activities performed by different users (such as tenants, landlords, and administrators) and how they interact with the system.

This activity diagram provides a high-level overview of the core activities involved in a house rental management system. It can be further refined and detailed to capture specific use cases and requirements.

In summary, the activity diagram captures the various tasks and decisions made by tenants, landlords, and administrators in the house rental system, and how each role interacts with the system to manage property rentals efficiently.

Tenant :

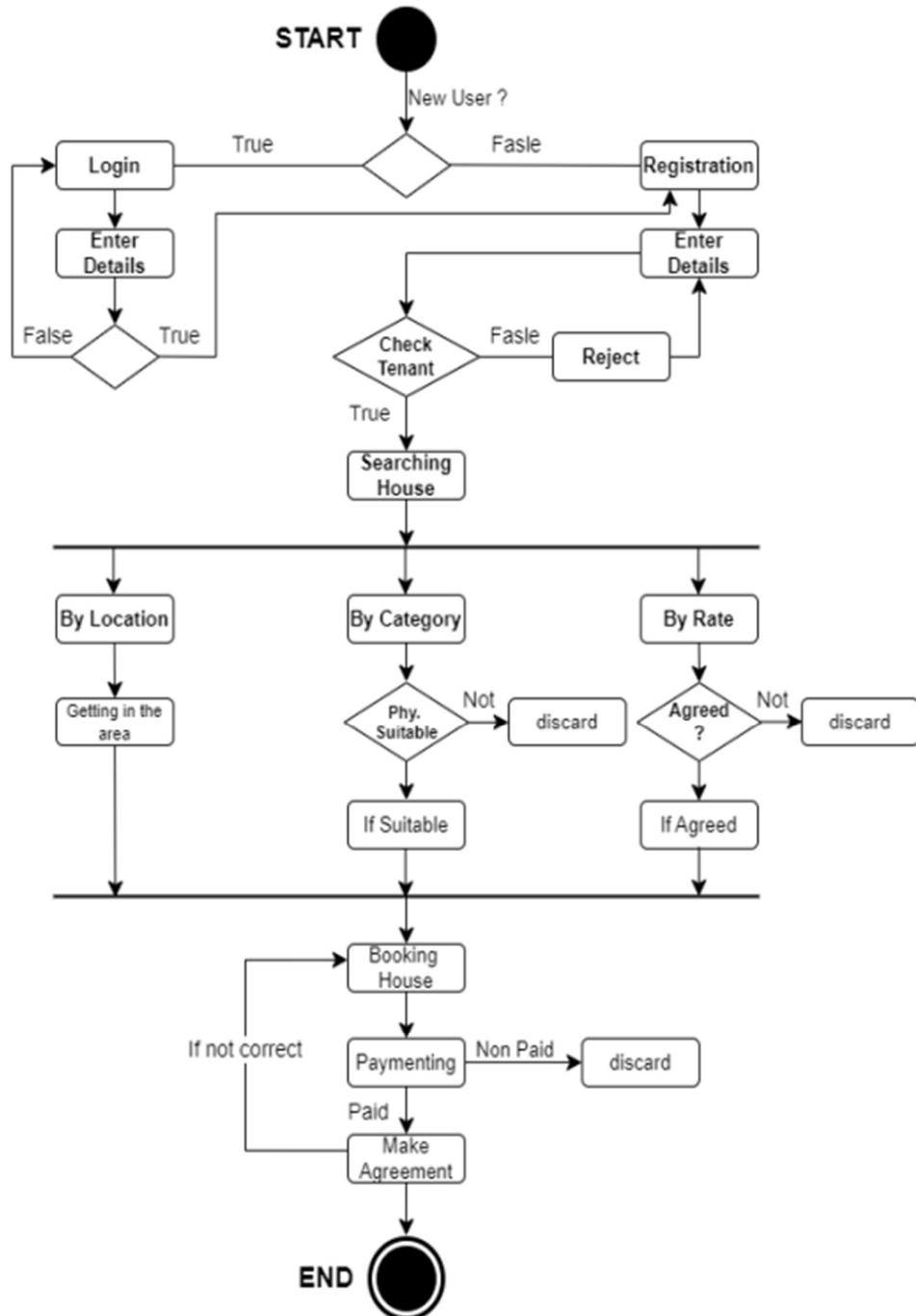


Fig 5.7 Activity Diagram (Tenant)

LandLord :

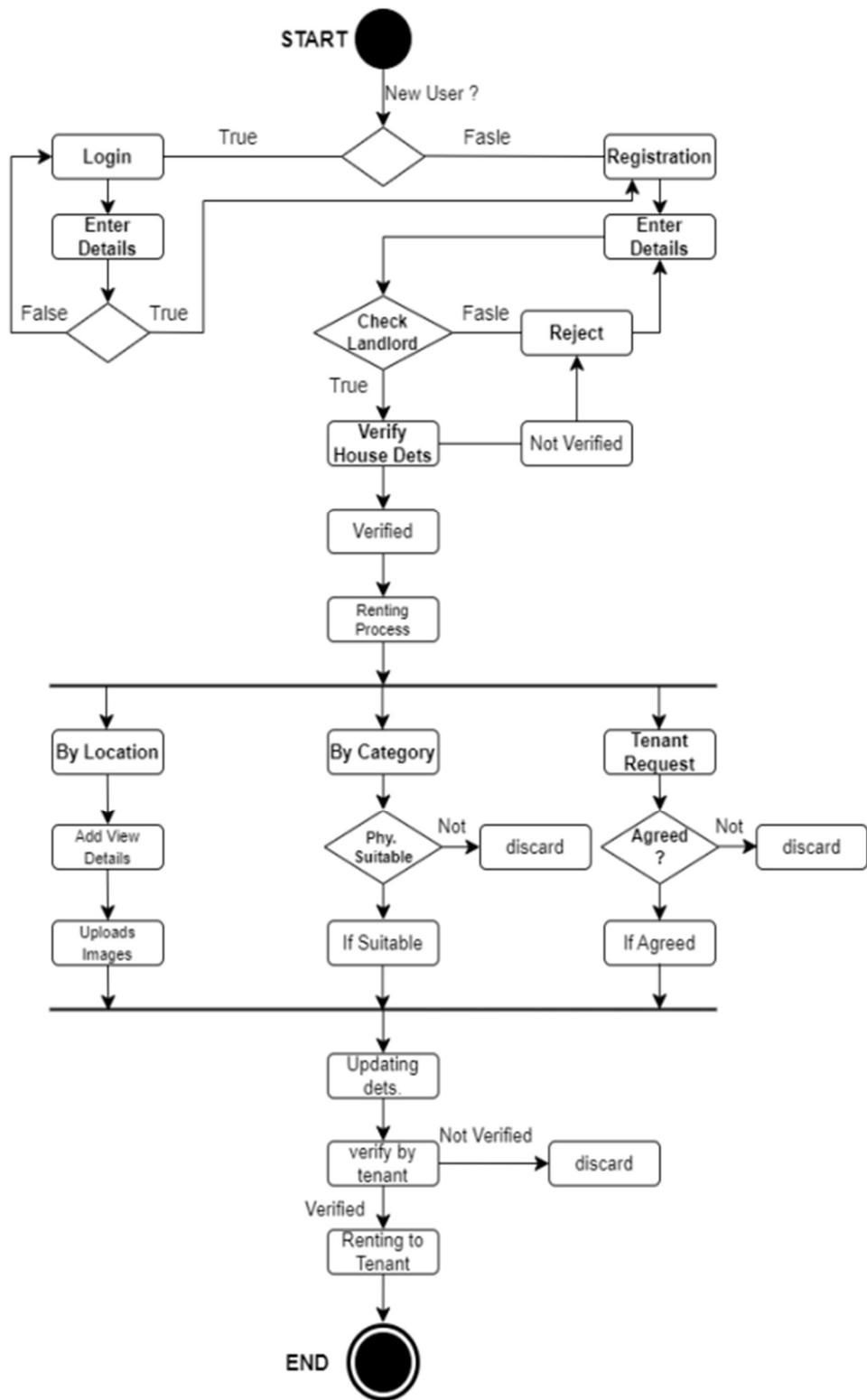


Fig 5.8 Activity Diagram (Landlord)

5.5 State Diagram

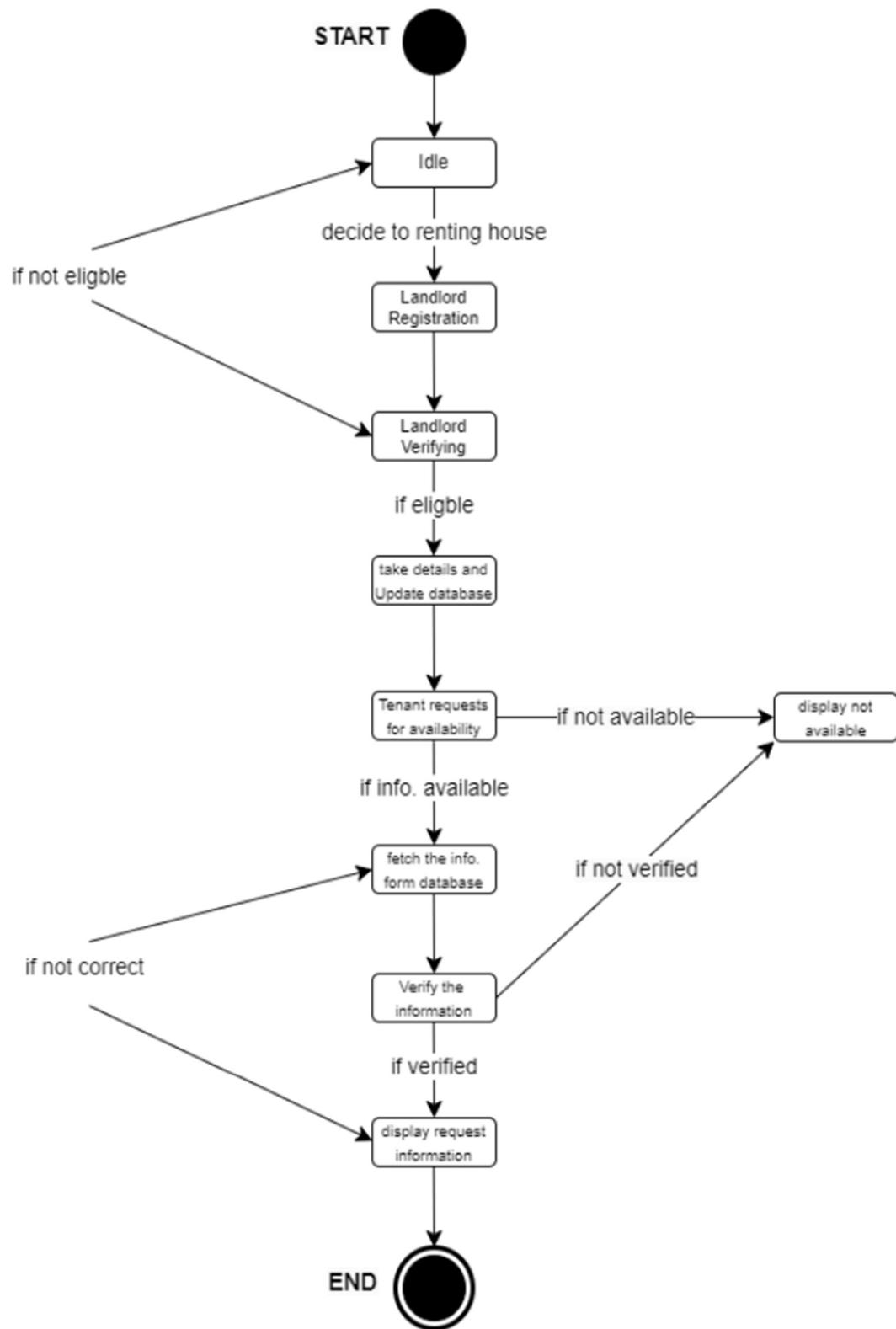


Fig 5.9 State Diagram

6.Result, Discussion and Conclusion

- **Result**
- **Test Case**
- **Snapshots**
- **Discussion**
- **Conclusion**

6.1 Result

Result for House Rental Management System (HRMS)

The House Rental Management System (HRMS) was successfully developed to address the challenges of managing rental properties, tenants, and payments. The system was designed to streamline processes for both landlords and tenants, ensuring efficient property management and smooth rent transactions.

Key results of the HRMS implementation include:

1. **Property Management:** Landlords can efficiently list, update, and manage properties, including details like rent, location, and availability. The system allows them to monitor property status in real-time.
2. **Tenant Management:** The system stores and organizes tenant information, lease agreements, and payment history. It allows easy tracking of tenants' rental status and lease durations.
3. **Payment Management:** Tenants can view payment details, make rent payments online, and receive notifications about due dates. The system allows landlords to track payments and manage arrears effectively.
4. **User Interface:** The system provides an intuitive interface for both landlords and tenants, simplifying their interactions with the system. The user experience is smooth, with easy access to the necessary information.
5. **Reporting and Analytics:** The HRMS includes various reports such as rent history, property occupancy status, and lease expiration, providing landlords with valuable insights into their property portfolio.
6. **Efficient Resource Management:** By automating key tasks such as rent reminders and payment tracking, the HRMS reduces manual effort and enhances operational efficiency.

The system was developed using modern technologies, ensuring scalability and ease of maintenance. It is expected to significantly improve the management of rental properties, offering a user-friendly and comprehensive solution to landlords and tenants alike.

6.2 Test Case

6.2.1 Valid Test Case

Test Case Name: Verify Successful Creation of a New Listing with Valid Input

Description: This test case ensures that a user can successfully create a new listing by entering valid details in all required fields. It checks the form validation logic, field acceptance criteria, and the submission process, verifying that the data is properly processed and saved.

Home All Listings Add new Listings

Create New Listing

Title

 ✓
Title looks good!

Description

 ✓
6

Image link

Price Country

1650 India

Location

 ✓
4

Add



© HavenHub Private Limited

[Privacy](#) [Terms](#)

Fig 6.1 Valid Test Case

6.2.2 Non-Valid Test Case

Test Case Name: Verify Error Handling for Invalid Price Input

Description: This test case ensures that the application correctly identifies and rejects invalid input in the "Price" field. It checks that proper validation logic is implemented and that the user receives a clear error message when an invalid price (e.g., "0000") is entered.

Home All Listings Add new Listings

Create New Listing

Title

 ✓
Title looks good!

Description

 ✓
6

Image link

Price Country

0000 India

Location

 ✓
4

Add



© HavenHub Private Limited

[Privacy](#) [Terms](#)

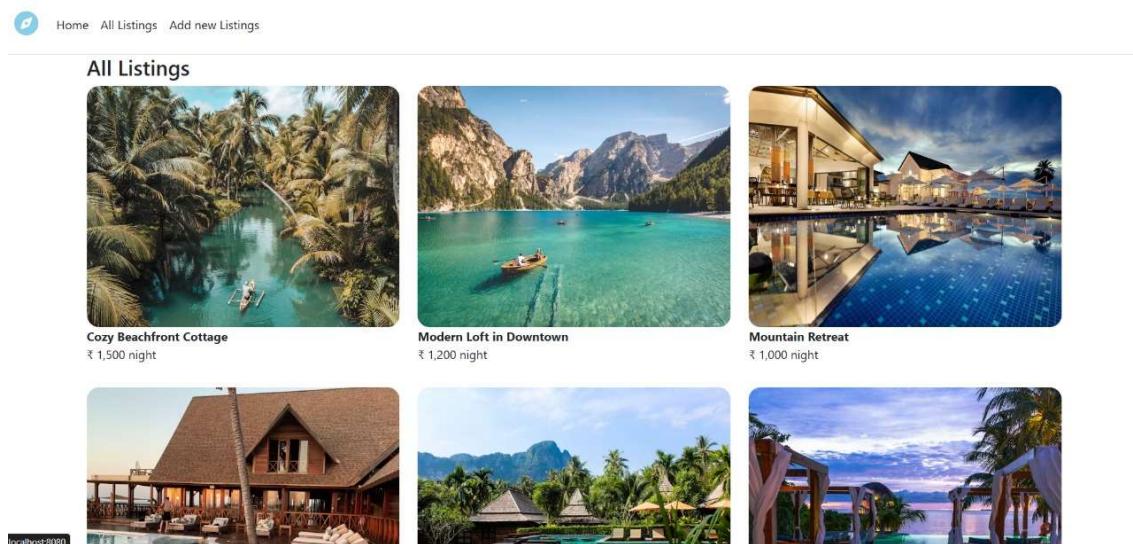


Fig 6.2 Non-Valid Test Case

6.3 Snapshots

6.3.1. HOMEPAGE

DESCRIPTION: This is the homepage of the House Rental Management System that can be accessed by anyone who wishes to learn more about the available properties for rent, rental policies, and other services offered by the system.





Charming Cottage in the Cotswolds
₹ 1,200 night



Historic Brownstone in Boston
₹ 2,200 night



Beachfront Bungalow in Bali
₹ 1,800 night



Fig 6.3 HOMEPAGE

6.3.2 Details and Review

Home All Listings Add new Listings

Mountain View Cabin in Banff



Enjoy breathtaking mountain views from this cozy cabin in the Canadian Rockies.
₹ 1,500
Banff
Canada

[Edit](#) [Delete](#)

Leave a Review

Rating

Comments

Home All Listings Add new Listings

₹ 1,200
New York City
United States

[Edit](#) [Delete](#)

Leave a Review

Rating

Comments

[submit](#)

© HavenHub Private Limited

[Privacy](#) [Terms](#)

Fig 6.4 Details and Review

6.3.3 Create New Listing

Home All Listings Add new Listings

Create New Listing

Title
 Title looks good!

Description
 ✓

Image link
 ✓

Price Country
 ✓

Location

Home All Listings Add new Listings

Create New Listing

Title
 Title looks good!

Description
 ✓

Image link
 ✓

Price Country

Location

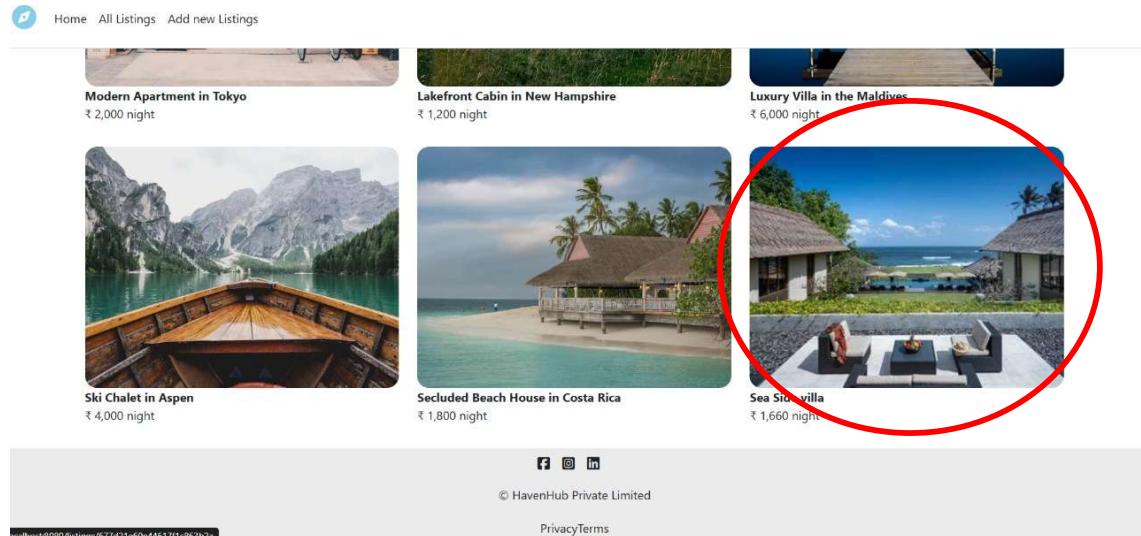


Fig 6.5 Create New Listing

6.3.4 Error Detection

Home All Listings Add new Listings

Create New Listing

Title
Sea Side villa
Title looks good!

Description
Full facing beach View & access to beach ✓

Image link
<https://th.bing.com/th/id/OIP.OHMaJgjk82-ksPAL0vJ9PQHaEJ?rs=1&pid=ImgDetMain> ✓

Price 0000 ⓘ	Country India
-----------------	------------------

Price should be valid

Location
goa.goa

Add



Fig 6.6 Error Detection

6.4 Discussion

A **House Rental Management System (HRMS)** is a software application or platform designed to facilitate and streamline the processes involved in renting out residential properties. It is commonly used by landlords, property managers, and tenants to manage rental properties, payments, maintenance requests, and other related activities.

The goal of such a system is to automate and simplify the administrative tasks related to property management, ensuring efficiency and improved communication between all stakeholders.

Key Features of a House Rental Management System

1. Tenant Management

- **Tenant Registration:** Allows tenants to register, update profiles, and submit necessary documents.
- **Lease Management:** Tracks lease start and end dates, renewal reminders, and lease terms.
- **Payment Tracking:** Manages rent payments, due dates, and sends reminders or invoices to tenants.
- **Tenant Communication:** A communication platform for landlords and tenants to exchange messages and updates.

2. Property Listings

- **Property Database:** A comprehensive list of available rental properties with details like size, amenities, pricing, and location.
- **Property Search:** Filters for potential tenants to search by criteria like price range, number of bedrooms, location, etc.
- **Property Viewing Management:** Option for tenants to schedule viewings and for landlords to track them.

3. Rent Collection and Payment

- **Online Payment Gateway:** Tenants can pay rent through the system using credit/debit cards, bank transfers, or other online payment methods.
- **Automated Reminders:** Sends payment reminders or overdue notices automatically to tenants.
- **Receipts and Invoices:** Automatically generates receipts and invoices for payments made.
- **Late Fee Calculation:** The system can automatically calculate and apply late fees if rent is not paid on time.

4. Maintenance Requests

- **Tenant Maintenance Requests:** Tenants can submit maintenance requests through the system, including descriptions, images, or videos of the issue.
- **Ticketing System:** Assigns tickets to maintenance staff or third-party vendors to resolve the issues.
- **Status Updates:** Tenants can track the status of their maintenance requests in real time.

5. Financial Management

- **Income and Expense Tracking:** The system keeps track of rent income, property expenses (repairs, utilities, etc.), and overall profitability.
- **Tax Reports:** Generates reports to help landlords with tax filing, including income reports and deductions.
- **Financial Dashboard:** Provides an overview of revenue, expenses, outstanding payments, and more in an easy-to-read dashboard.

6. Document Management

- **Lease Agreement Templates:** Pre-made templates for lease agreements that can be customized and signed digitally by both parties.
- **Important Documents:** Storage for documents such as ID proofs, lease agreements, property inspection reports, etc.

7. Security and Access Control

- **Role-based Access:** Different roles for users (landlord, property manager, tenant) to ensure data privacy and security.
- **Data Encryption:** Protection of sensitive information such as payment details, personal documents, and communication.

8. Reporting and Analytics

- **Financial Reports:** Income, expense, and profit reports for each property or across all properties.
- **Occupancy Rate Reports:** Tracks the occupancy rates of rental properties, helping landlords analyse trends and adjust pricing strategies.
- **Tenant Behaviour Analysis:** Insights into tenant payment history, request frequency, and lease renewal tendencies.

Benefits of Using a House Rental Management System

1. **Efficiency:** Automates many administrative tasks, reducing the amount of time and effort needed to manage properties.

2. **Reduced Errors:** Automated calculations for rent payments, late fees, and invoices reduce human error.
3. **Improved Communication:** Enables better communication between landlords, tenants, and service providers, leading to a more streamlined operation.
4. **Cost Savings:** The system reduces the need for paper-based documentation and manual tracking of payments and repairs.
5. **Better Financial Control:** Landlords can gain real-time insights into income, expenses, and financial performance.
6. **Convenience for Tenants:** Tenants benefit from online payment options, easy maintenance request submission, and direct communication with landlords.
7. **Compliance:** Helps landlords ensure they comply with local rental regulations and laws.

Types of Users

1. **Landlords:** Property owners who rent out residential units. They can use the system to track leases, manage payments, and communicate with tenants.
2. **Tenants:** Individuals or families renting properties. Tenants can use the system to pay rent, report maintenance issues, and manage lease terms.
3. **Property Managers:** Individuals or companies responsible for managing multiple rental properties on behalf of landlords. Property managers typically have access to broader system functionalities.
4. **Maintenance Staff/Vendors:** They are responsible for handling maintenance requests submitted by tenants.

Technical Aspects

- **Cloud-based or On-premises:** HRMS can be hosted in the cloud (more common) or on-premises. Cloud-based solutions tend to be more scalable and accessible from anywhere.
- **Mobile App Integration:** A mobile version of the system allows landlords, tenants, and property managers to manage tasks on the go.
- **API Integration:** The system can integrate with other platforms like accounting software, payment gateways, and third-party maintenance services.

Challenges to Consider

1. **Security:** Managing sensitive personal and financial data requires robust security measures, such as encryption and multi-factor authentication.
2. **User Adoption:** Both landlords and tenants may be resistant to using a new system, especially if they are not tech-savvy.
3. **Customization:** A one-size-fits-all solution may not meet the specific needs of every landlord or property manager, so some degree of customization may be needed.
4. **Technical Support:** Ensuring reliable support for users facing technical issues is crucial to maintaining system effectiveness.

6.5 Conclusion

A House Rental Management System significantly enhances the rental process by automating tasks, providing better transparency, and improving efficiency for all parties involved. Its

features support both operational tasks (like rent collection and maintenance) and strategic tasks (like financial reporting and tenant management), making it an essential tool for modern property management.

House Rental business has emerged with a new goodies compared to the past experience where every activity concerning House rental business is limited to a physical location only. Even though the physical location has not been totally eradicated; the nature of functions and how these functions are achieved has been reshaped by the power of internet. Nowadays, customers can reserve book/buy/sale House online, rent House online, and have the house contracted successfully without any sweat once the customer is a registered member of the House Rental Management System.

The web based House rental system has offered an advantage to both Tenants as well as Landlords to efficiently and effectively manage the business and satisfies customers' need at the click of a button.

7. REFERENCES

- **References**

7.1 References

Books Used:

Software Engineering - R.S. Pressman

PHP for Dummies

PHP Beginners Guide by McGraw-Hill Publication

JavaScript by McGraw-Hill Publication

References Used:

<http://www.resgroup.net/>

<http://www.alrealstate.solutions>

<https://unsplash.com/>

<https://www.npmjs.com/package/joi>

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

<https://mongoosejs.com/>

Wikipedia.org

www.w3schools.com

www.webcampzg.org