



## **Project and Professionalism (6CS007)**

### **Artefact Design**

### **GharBhada: Rental Management Platform**

**Full Name:** Krishma Shrestha

**Student Number:** 2431161

**Course:** Project and Professionalism (6CS007)

**University Email Address:** K.Shrestha32@wlv.ac.uk

**Supervisor:** Siddhartha Thapa

**Reader:** Durga Pokharel

**Date of Submission:** 12/26/2025

## **Abstract**

GharBhada is a complete web-based rental management solution for digitizing and improving all aspects of rental management in Nepal. By combining property search, advanced searching tools, booking requests, digital agreements with electronic signatures, automatic payments for all rent and utilities (including payments for metered water, electricity and waste collection), tracking complaints, and setting up role specific dashboards, it solves problems of fraud, inefficiencies, lack of transparency and accountability that have historically plagued the rental market in Nepal. Within GharBhada you will find three roles; tenant, owner (landlord) and administrator. GharBhada was developed using the latest technology stack based on fullstack JavaScript (React.js, Node.js with Express, MySQL) as well as incorporating local payment gateways (eSewa and Khalti), and establishing a trust/safety scoring system. GharBhada employs Kanban as a method of managing and facilitating the agile, iterative development process. GharBhada will allow for increased accountability, fewer disputes, a more streamlined, secure and transparent rental experience for the Nepali community.

## **Acknowledgment**

I would like to express my deepest gratitude to my Supervisor, Mr. Siddhartha Thapa, and my Reader, Mr. Durga Pokharel, for their invaluable guidance, constructive feedback, and continuous support throughout the development of this Final Year Project on GharBhada. Their expert insights, thoughtful suggestions, and encouragement have significantly shaped my ideas and approach, enabling me to overcome challenges and refine my work effectively.

I am also grateful to the University of Wolverhampton for providing an excellent academic environment, access to essential resources, online databases, and learning platforms that have greatly facilitated my research and project implementation.

Finally, I extend my sincere appreciation to all those who contributed directly or indirectly to this endeavour, whose support helped me overcome challenges and refine the artefacts, designs, and test cases presented in this report.

Yours sincerely,

Krishma Shrestha

12/26/2025

## Table of Contents

1. Introduction.....	1
1.1. Purpose.....	1
1.2 Subsystems:.....	1
1.2.1. User Management System .....	1
1.2.2. Property Listing and Search System .....	1
1.2.3. Booking and Notification System.....	2
1.2.4. Agreement and Contract Management System.....	2
1.2.5. Payment and Billing System.....	2
1.2.6. Complaint and Maintenance System .....	2
1.2.7. Admin Dashboard System.....	2
1.3. Functional Decomposition Diagram (FDD) .....	3
2. Overall Description.....	4
2.1. Overview of Product .....	4
2.2. Features/Functions of Product .....	4
2.3. Stakeholders.....	4
2.4. Environment for Operation .....	5
2.5. Design & Implementation Constraints.....	5
2.6. Assumptions & Dependencies .....	5
3. System Requirements.....	6
3. 1. User Management Subsystem (UM).....	6
3.2. Property Management Subsystem (PRM) - Property Listing and Search .....	12
3.3. Booking and Notification Subsystem (BN) .....	17
3.4. Agreement and Contract Management Subsystem (ACM) .....	21
3.5. Payment and Billing Subsystem (PMS).....	26
3.6. Complaint and Maintenance Subsystem (CM) - Included for Completeness.....	31
3.7. Admin Dashboard Subsystem .....	36
4. System Design (Whole System Level) .....	41
5. Entity Relationship Diagram (ERD).....	45
6. Data Dictionary for GharBhada Rental Management Platform.....	46
7. Test Cases.....	60
8. Wireframes .....	61

## Table of figures

Figure 1: Functional Decomposition Diagram (FDD) of GharBhada .....	3
Figure 2: Use-Case Diagram for UMS.....	8
Figure 3: Activity Diagrams for UMS .....	9
Figure 4: Sequence Diagrams for UMS .....	10
Figure 5: Class Diagram for UMS .....	11
Figure 6: Use-Case Diagram for PRM.....	14
Figure 7: Activity Diagrams for PRM .....	15
Figure 8: Sequence Diagrams for PRM .....	16
Figure 9: Class Diagram for PRM .....	17
Figure 10: Use-Case Diagram for BN.....	18
Figure 11: Activity Diagrams for BN .....	19
Figure 12: Sequence Diagrams for BN .....	20
Figure 13: Class Diagram for BN .....	20
Figure 14: Use-Case Diagram for ACM .....	22
Figure 15: Activity Diagrams for ACM .....	23
Figure 16: Sequence Diagrams for ACM.....	24
Figure 17: Class Diagram for ACM .....	25
Figure 18: Use Case Diagram for PMS.....	27
Figure 19: Activity Diagrams for PMS.....	28
Figure 20: Sequence Diagrams for PMS.....	29
Figure 21: Class Diagram for PMS .....	30
Figure 22: Use Case Diagram for CM .....	32
Figure 23: Activity Diagram for CM .....	33
Figure 24: Sequence Diagrams for CM .....	34
Figure 25: Class Diagrams for CM .....	35
Figure 26: Use Case Diagram for AD .....	37
Figure 27: Activity Diagrams for AD .....	38
Figure 28: Sequence Diagrams for AD .....	39
Figure 29: Class Diagram for AD .....	40
Figure 30: Activity Diagram for the Whole System .....	41
Figure 31: Use-Case Diagram for the Whole System.....	42
Figure 32: Sequence Diagram for the Whole System .....	43
Figure 33: Class Diagram for the Whole System.....	44
Figure 34: Entity Relationship Diagram (ERD).....	45
Figure 35: Test Cases of GharBhada.....	60

# **1. Introduction**

## **1.1. Purpose**

The GharBhada platform is a web-based rental management solution designed specifically for Nepal's housing market. This Software Requirements Specification document describes the functional, non-functional, and usability requirements for the product and has been organised into subsystems as defined by the Functional Decomposition Diagram (FDD). We will adopt an Agile Kanban approach and develop the product incrementally by concentrating on a single subsystem at any one time and incorporating feedback through iterative refinement.

## **1.2 Subsystems:**

### **1.2.1. User Management System**

- This is the subsystem to provide registration and authentication as well as maintain profiles for users based upon their role (Tenant, Owner or Admin). In addition to these items, this subsystem will also provide the ability to approve all new user registrations, and to determine trust scores.

### **1.2.2. Property Listing and Search System**

- This subsystem will allow Owners to create property listings that are detailed including pictures, amenities and rental deposit amounts greater than NPR 5,000 prior to admin approval. In addition, Tenants are able to conduct complex searches, filter their search results, view properties using a map, create wish lists and receive AI recommendations.

### **1.2.3. Booking and Notification System**

- This subsystem will enable Tenants to submit requests for bookings, and have the request approved or rejected by the Owner. The system will automatically cancel competing requests and will send notifications for both email/SMS/in-app of response.

### **1.2.4. Agreement and Contract Management System**

- Tenants and Owners will receive customized agreements digitally (1 - 5 years in length or custom), will allow for negotiation and signing via e-signatures as well as have secure PDF download available.

### **1.2.5. Payment and Billing System**

- This subsystem provides automated payment management for the deposit, monthly rental and utilities (Metered Electricity/Water + Fixed Garbage) with eSewa/Khalti integration for partial payments, reminders and receipts.

### **1.2.6. Complaint and Maintenance System**

- This subsystem allows Tenants to submit complaints with attached documentation and allows for tracking of the Owner/Admin intervention resolution status and emailing notifications of status of all complaints submitted.

### **1.2.7. Admin Dashboard System**

- Administrators will be provided access to the Admin Dashboard System for monitoring purpose to provide oversight, verification of users/properties, dispute resolution, payment/complaint activity and analytics/reports of all of the above.

1.3. Functional Decomposition Diagram (FDD)

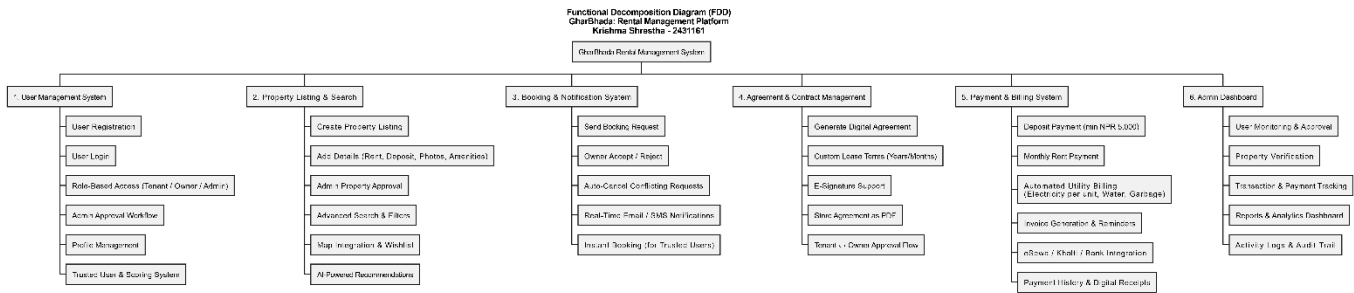


Figure 1: Functional Decomposition Diagram (FDD) of GharBhada



## **2. Overall Description**

### **2.1. Overview of Product**

GharBhada will create a cohesive rental market in Nepal by utilizing an all-inclusive digital rental management solution as opposed to only listing properties/rentals like current competitors (e.g., Lalpurja Nepal). The integration of a verification, agreement and payment system along with dispute management will increase trust and efficiency for both tenants and landlords.

### **2.2. Features/Functions of Product**

- Tenant: Create profile, search, reserve rental, sign agreement, pay rent/utilities; submit complaint about rental; view all transactions via dashboard.
- Landlord: List properties, manage/track reservations; receive payments; respond/handle tenant complaints.
- Administrator: Verify tenants/landlords; track/monitor all transactions; handle disputes; generate reports on performance data through analytics.
- System: Provide trust score to all users; notify users; generate usage reports via analytics.

### **2.3. Stakeholders**

- Tenants: Individuals looking to rent a property.
- Landlords: Individuals who rent their property.
- Administrators: Individuals who enforce standards for the platform and protect users.
- Guests: Unregistered individuals browsing rental listings.

## **2.4. Environment for Operation**

Frontend: React.js (responsive web application).

Backend: Node.js with Express.js framework.

Database: MySQL.

Gateway Payment Processor: eSewa, Khalti.

Notification System: Email (Nodemailer), SMS API.

Web Host: AWS/Heroku (scalable).

Browser Compatibility: Any web browser with internet access and the ability to render JavaScript code.

## **2.5. Design & Implementation Constraints**

Phase 1 of platform will be web-based only (no native mobile applications).

Entire system will be developed using JavaScript full-stack to maintain consistency.

All agreements will be generated from template agreements (not legal advice).

Kanban methodology will be followed in tandem with Jira project management for purpose tracking.

## **2.6. Assumptions & Dependencies**

Users will have stable internet connections and devices (all types).

All three payment gateway companies (eSewa and Khalti) will remain in business.

All users will be validated through administrator approval processes.

Basic AI recommendations will be used at a minimum.

### 3. System Requirements

Legend:

- F: Functional Requirement
- NF: Non-Functional Requirement
- UR: Usability Requirement

Subsystems abbreviations: UMS, PRM, BNS, ACM, PMS, CMS, ADS

#### 3. 1. User Management Subsystem (UM)

**Subsystem Overview:** The User Management Subsystem handles registration, authentication, profile management, role-based access, user verification, and trust scoring for three primary roles: Tenant, Owner (Landlord), and Admin. New accounts require admin approval for authenticity. It supports secure login and integrates trust indicators (e.g., trusted badge for owners active over one year).

**Legend**

- F: Functional Requirement
- NF: Non-Functional Requirement
- UR: Usability Requirement

<b>Req. Code</b>	<b>Requirement Description</b>	<b>Use Case / Notes</b>
UM-F-1.0	The system shall allow Tenants and Owners to self-register with details including name, email, phone, address, citizenship number/ID proof, and role selection (Tenant/Owner).	Register User
UM-F-1.1	The system shall support social registration (Google, Facebook) for Tenants and Owners.	Social Register
UM-F-2.0	The system shall allow users to login using email/password or social credentials.	Login User
UM-F-3.0	The system shall allow users to manage their profile (update details, upload ID proof).	Manage Profile
UM-F-4.0	The system shall assign roles (Tenant, Owner, Admin) and restrict access accordingly.	Role-Based Access
UM-F-5.0	The Admin shall approve/reject new user registrations and assign/verify roles.	Verify User
UM-F-6.0	The system shall calculate and display trust scores/badges (e.g., "Trusted Owner" after 1 year of activity, based on completed transactions/reviews).	Trust Scoring
UM-F-7.0	The Admin shall ban/unban users.	Ban/Unban User
UM-NF-1.1	Registration and login data shall be encrypted in transit (HTTPS/TLS) and at rest.	Security
UM-NF-1.2	Passwords shall require a combination of alphabets, numbers, and special characters (min. 8 characters).	Security
UM-NF-1.3	The system shall implement rate limiting and account lockout after failed login attempts.	Security

UM-NF-2.0	The system shall support at least 1,000 concurrent user sessions.	Performance
UM-UR-1.1	Password fields shall have a toggle to show/hide password, especially on mobile devices.	Usability on Mobile
UM-UR-1.2	Registration and login forms shall be responsive and support drag-and-drop for ID proof uploads.	Easy Upload
UM-UR-2.0	The system shall provide clear error messages and feedback during registration/login.	User Feedback

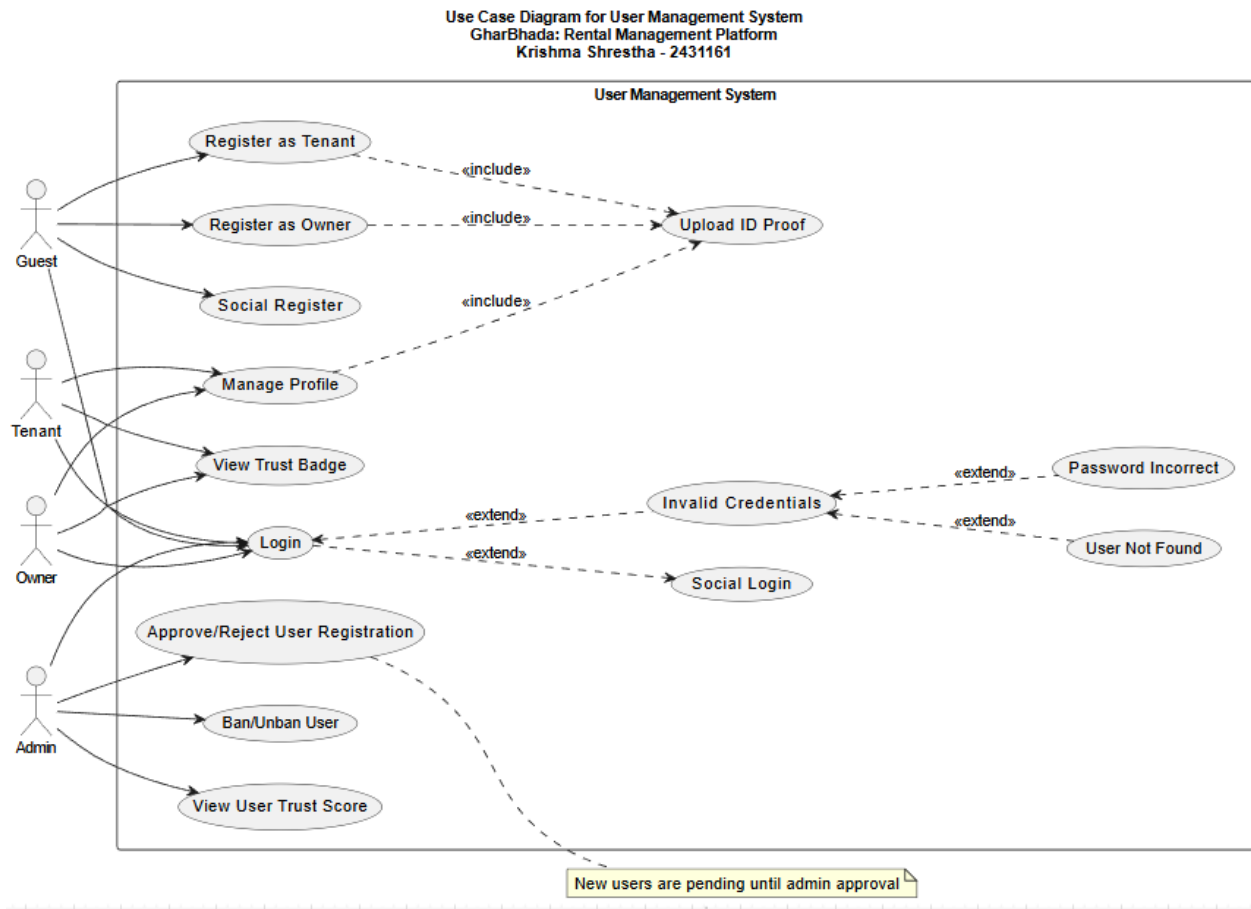


Figure 2: Use-Case Diagram for UMS

**Activity Diagram - User Management System**  
**GharBhada: Rental Management Platform**  
**Krishma Shrestha - 2431161**

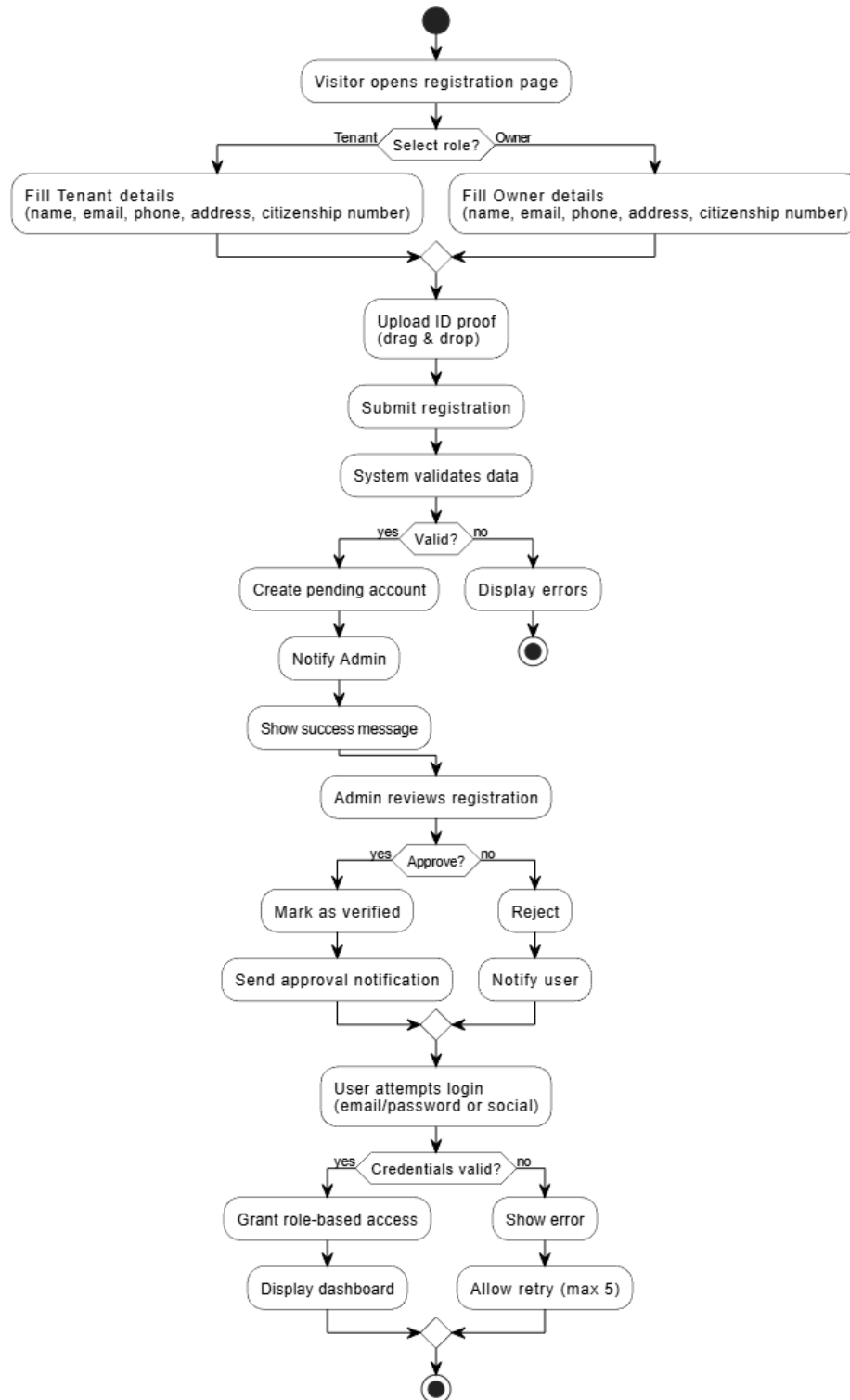


Figure 3: Activity Diagrams for UMS

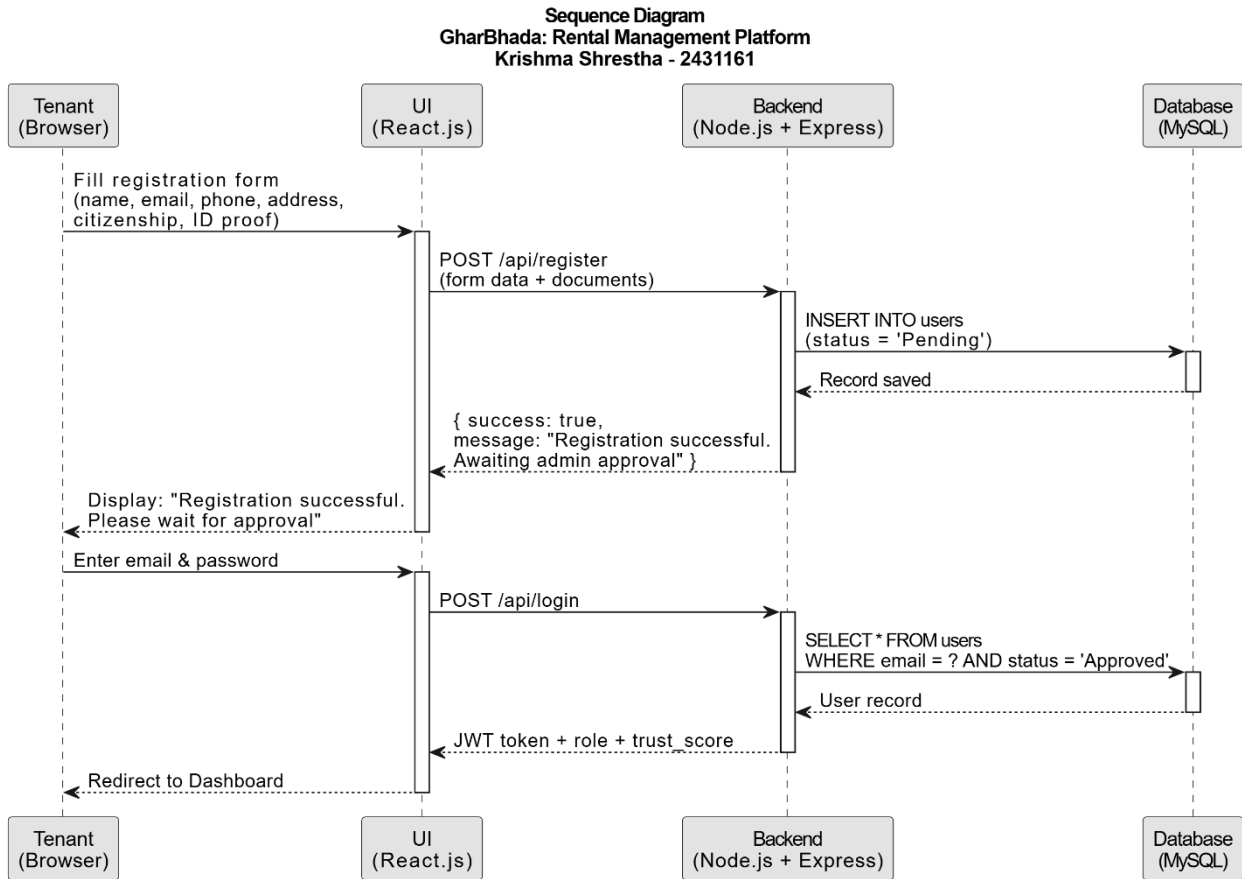


Figure 4: Sequence Diagrams for UMS

**Class Diagram - User Management System**  
**GharBhada: Rental Management Platform**  
**Krishma Shrestha - 2431161**

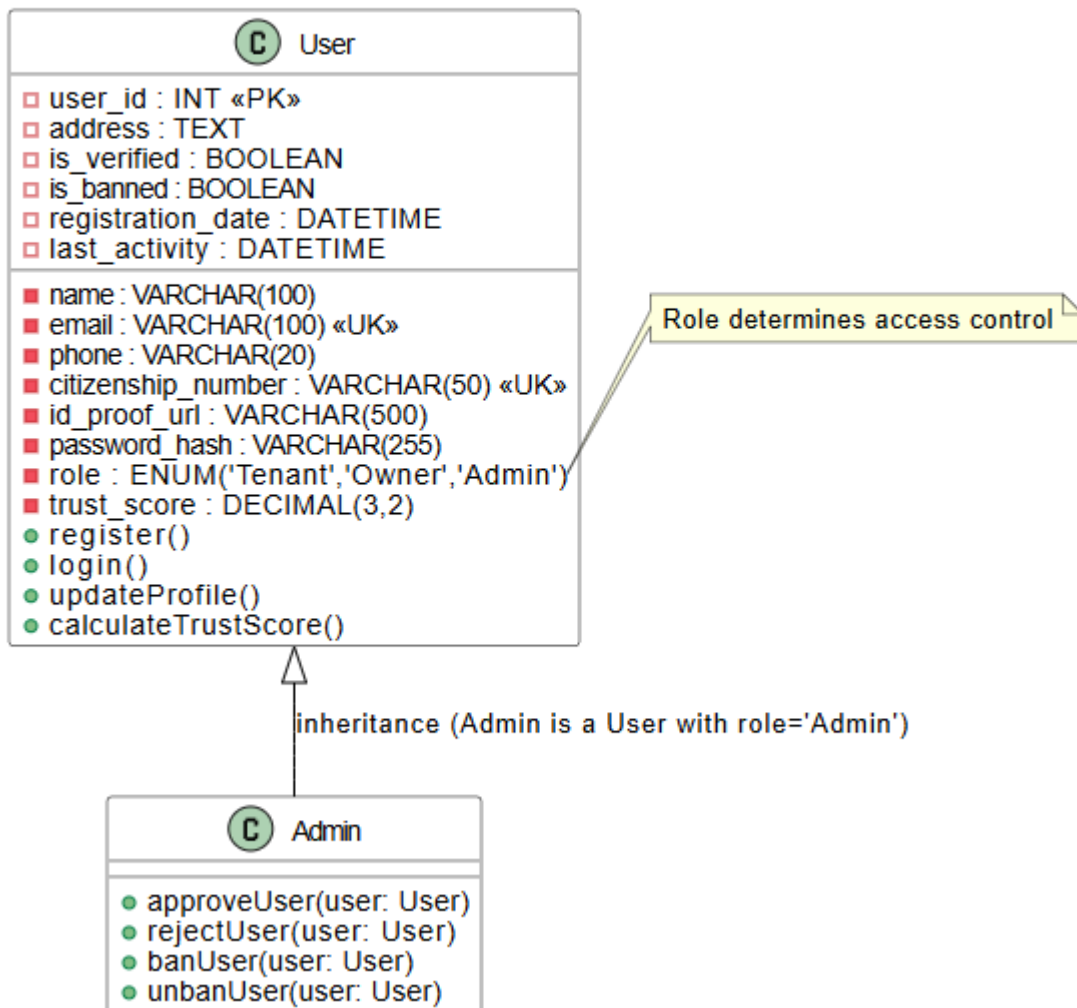


Figure 5: Class Diagram for UMS



### 3.2. Property Management Subsystem (PRM) - Property Listing and Search

**Subsystem Overview** This subsystem allows Owners to create and manage property listings. Tenants can search, filter, and view properties. Listings require admin approval. Includes detailed attributes, photos, amenities, and advanced search with map integration and AI recommendations.

Req. Code	Requirement Description	Use Case / Notes
PRM-F-1.0	Owners shall create property listings with details: address, rent price, deposit (min. NPR 5,000), type (e.g., 1BHK, 2BHK, room, house), photos (multiple), amenities, rules, and description.	Create Property
PRM-F-2.0	The system shall set new listings to "Pending Approval" status.	Pending Approval
PRM-F-3.0	Admin shall approve/reject property listings, making approved ones "Active".	Verify Property
PRM-F-4.0	Owners shall edit/delete their listings (subject to status).	Manage Property
PRM-F-5.0	Tenants shall search/filter properties by location, rent range, type, amenities, furnished/unfurnished, etc.	Search Properties
PRM-F-6.0	The system shall display property details including photos, owner info (with trust badge), and map view.	View Property Details
PRM-F-7.0	The system shall provide wishlist and AI-driven recommendations for Tenants.	Personalized Search
PRM-NF-1.1	Uploaded property images shall have minimum resolution 1024x760 and max file size 10MB.	Image Constraints
PRM-NF-1.2	Property IDs/SKUs shall be unique and auto-generated.	Uniqueness

PRM-NF-2.0	Search results shall load within 3 seconds for up to 10,000 properties.	Performance
PRM-UR-1.1	Image upload shall support drag-and-drop and multi-file selection.	Easy Upload
PRM-UR-1.2	Search interface shall be intuitive with auto-suggestions and responsive filters.	Usability

Use Case Diagram for Property Listing and Search System  
 GharBhada: Rental Management Platform  
 Krishma Shrestha - 2431161

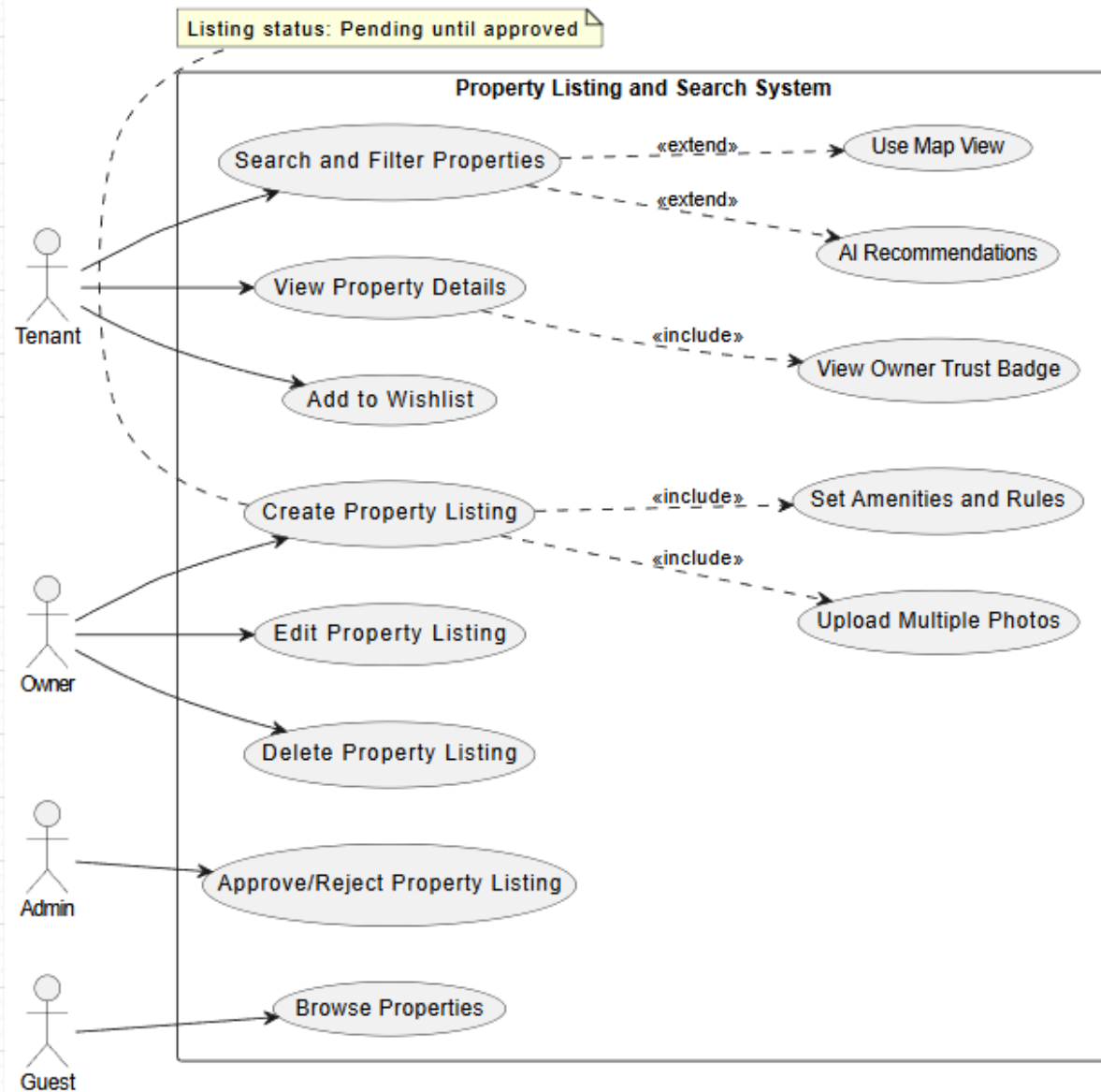


Figure 6: Use-Case Diagram for PRM

Activity Diagram - Property Listing and Search System  
GharBhada: Rental Management Platform  
Krishna Shrestha - 2431161

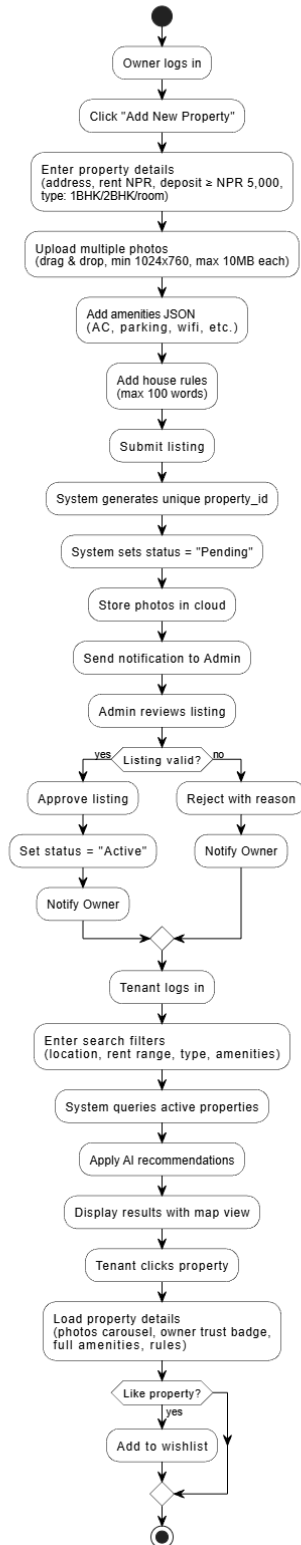


Figure 7: Activity Diagrams for PRM

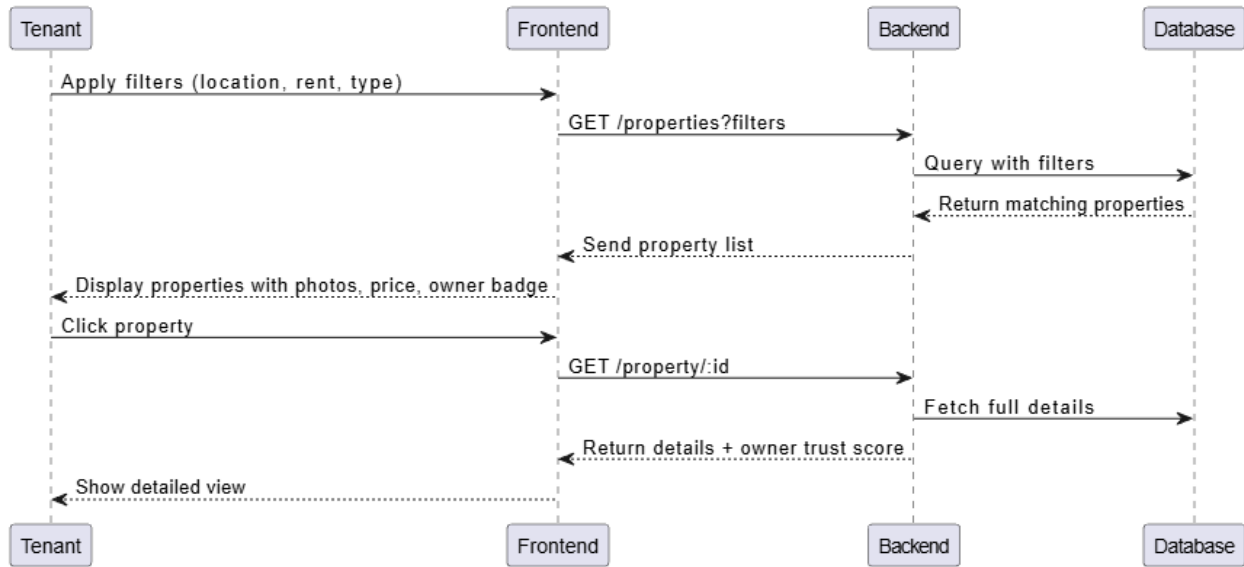


Figure 8: Sequence Diagrams for PRM

Class Diagram - Property Listing and Search System  
GharBhada: Rental Management Platform  
Krishma Shrestha - 2431161

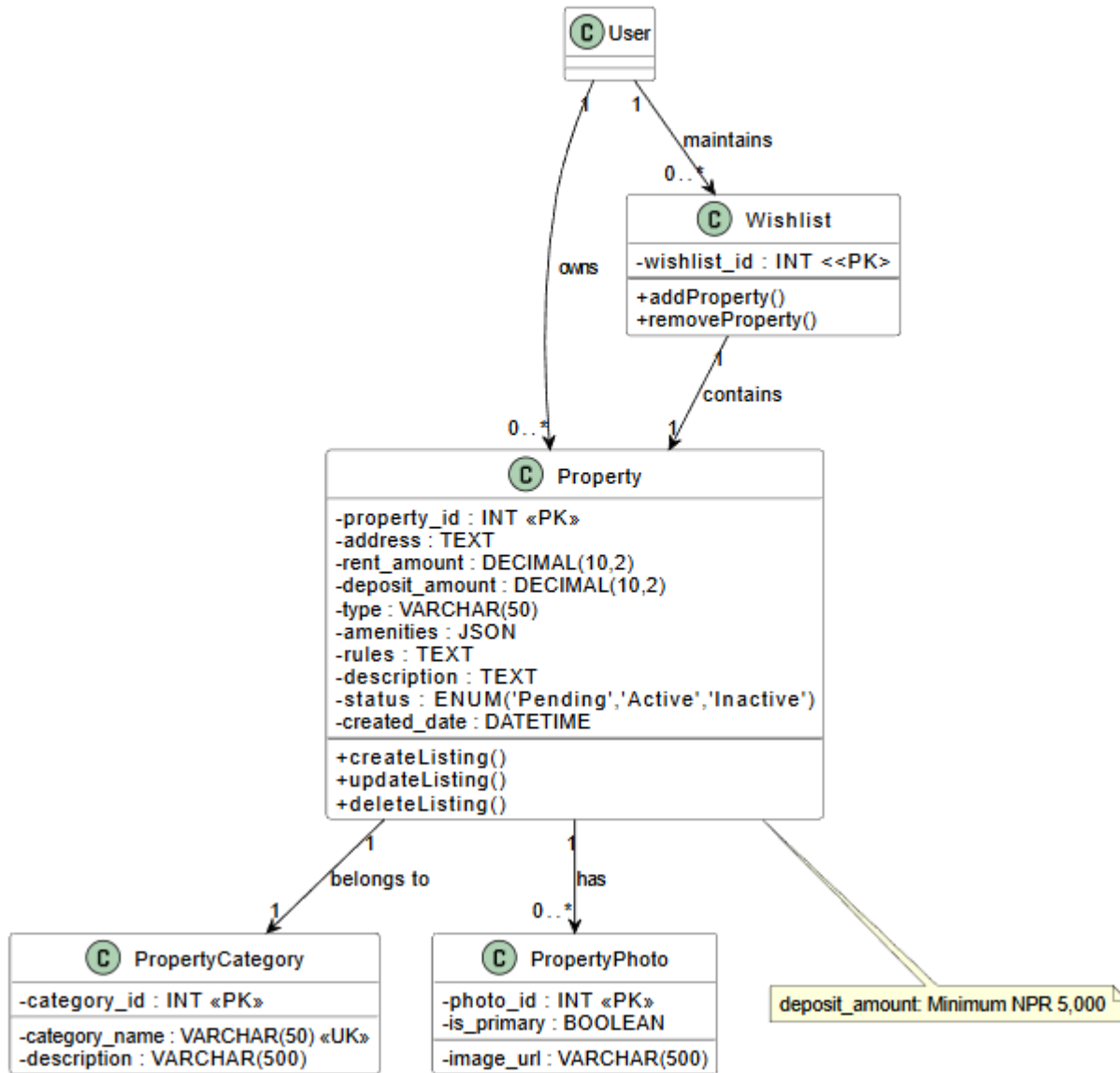


Figure 9: Class Diagram for PRM

### 3.3. Booking and Notification Subsystem (BN)

**Subsystem Overview** Manages booking requests from Tenants, approvals by Owners, and real-time notifications. Includes auto-cancellation logic for competing requests.

Req. Code	Requirement Description	Use Case / Notes
BN-F-1.0	Tenants shall send booking requests for active properties, providing personal details.	Send Booking Request
BN-F-2.0	Owners shall receive notifications and approve/reject requests.	Manage Booking Requests
BN-F-3.0	Upon approval, booking status changes to "Approved"; other pending requests auto-cancel.	Booking Approval
BN-F-4.0	The system shall send real-time notifications (email/SMS/in-app) for booking updates.	Notifications
BN-NF-1.0	Notifications shall be delivered within 10 seconds.	Performance
BN-UR-1.0	Notification center shall be easily accessible with unread indicators.	Usability

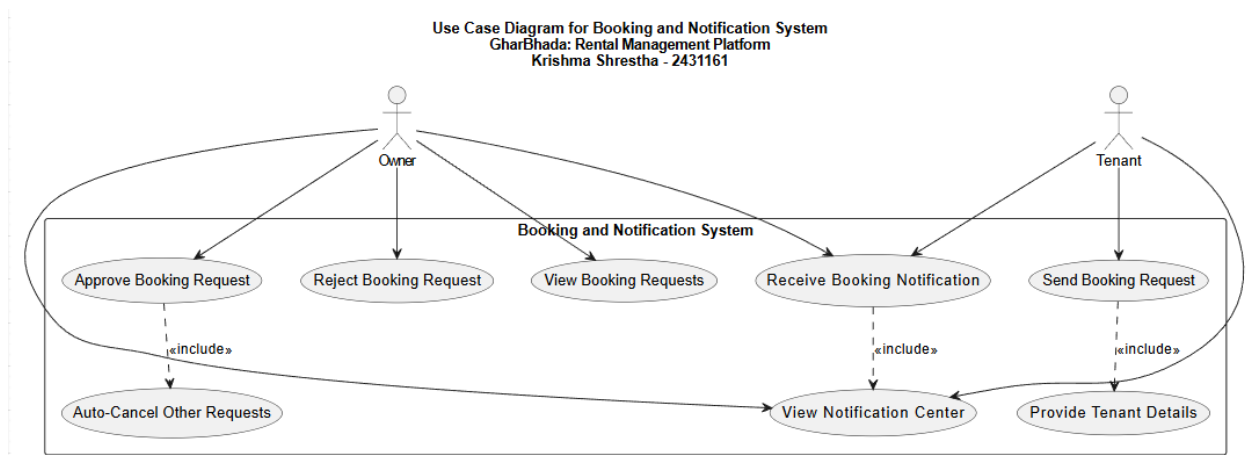


Figure 10: Use-Case Diagram for BN

**Activity Diagram - Booking and Notification System**  
**GharBhada: Rental Management Platform**  
**Krishma Shrestha - 2431161**

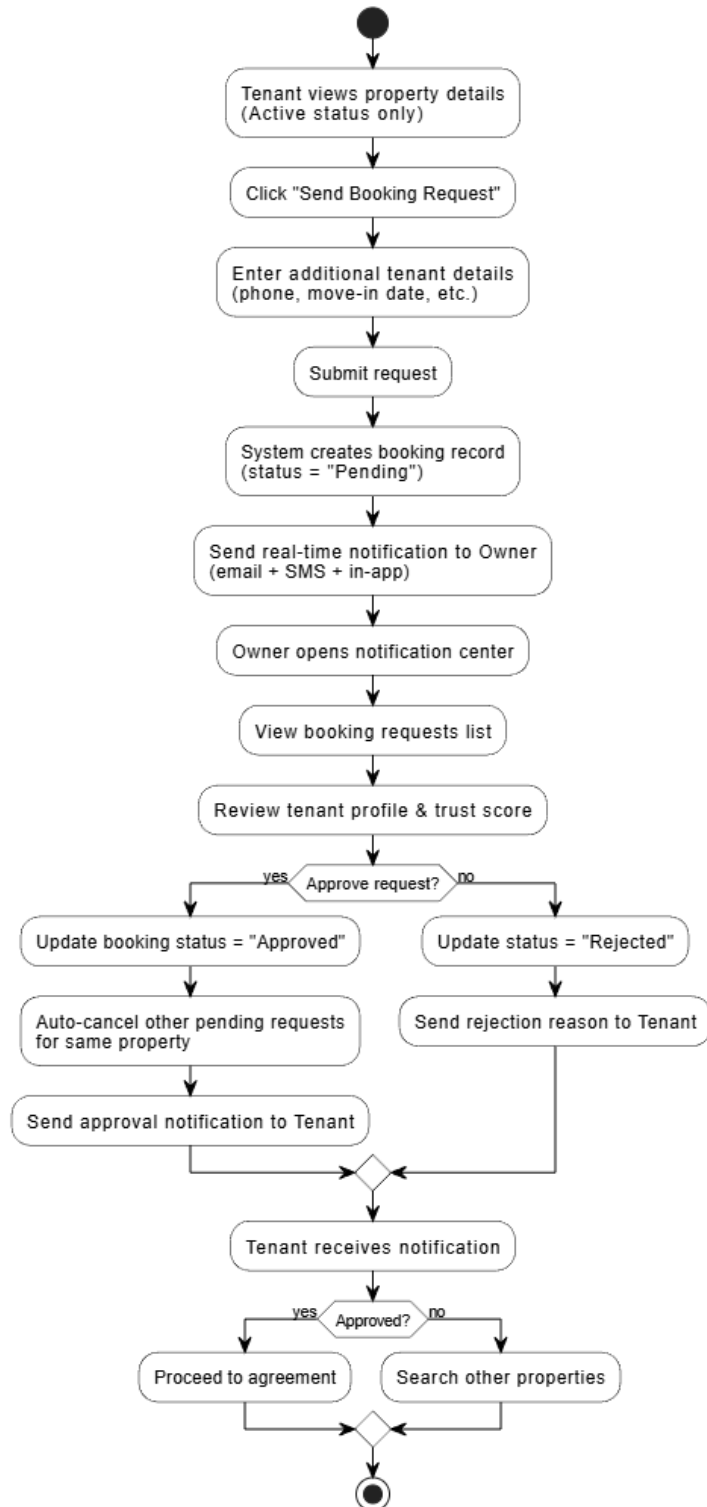


Figure 11: Activity Diagrams for BN



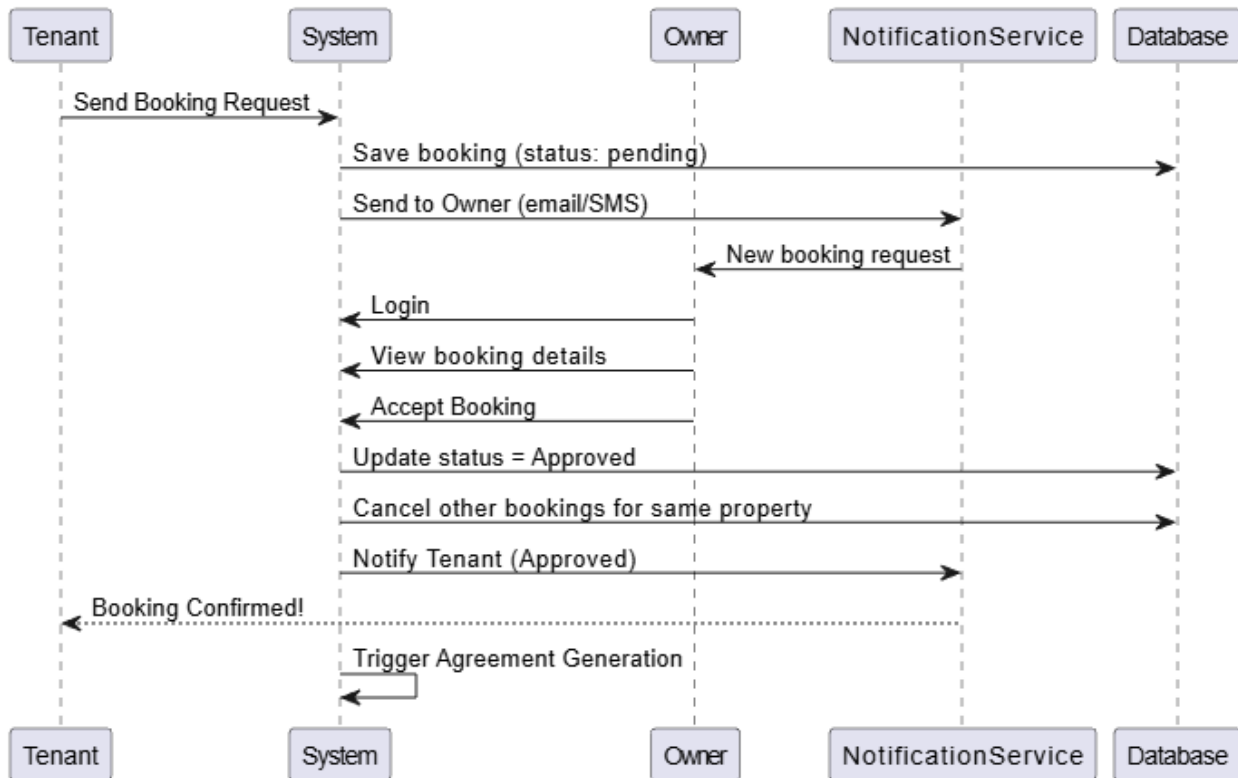


Figure 12: Sequence Diagrams for BN

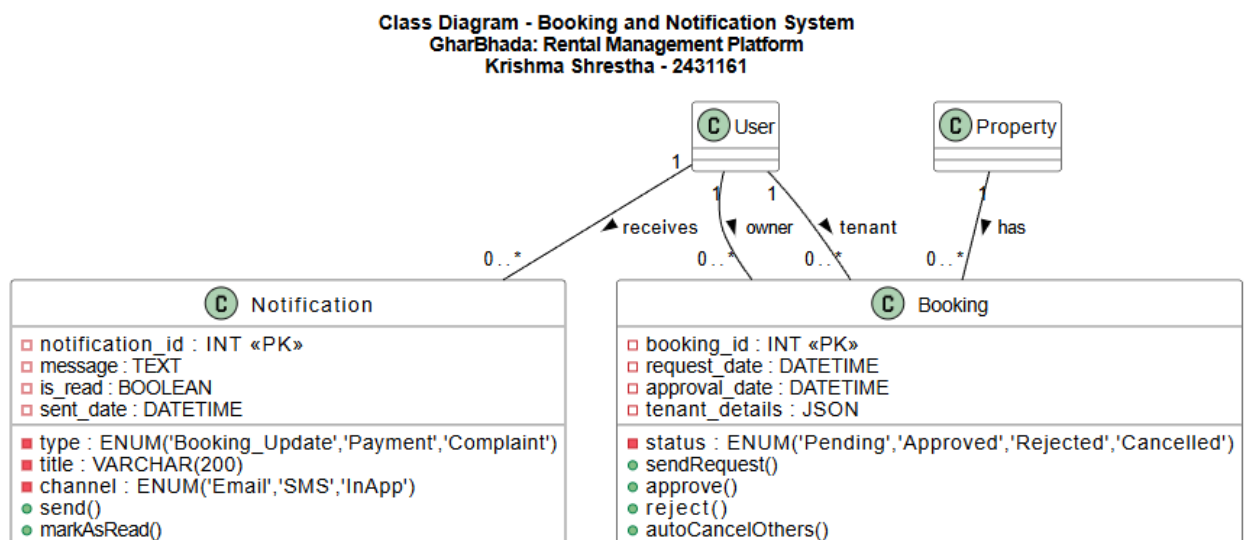


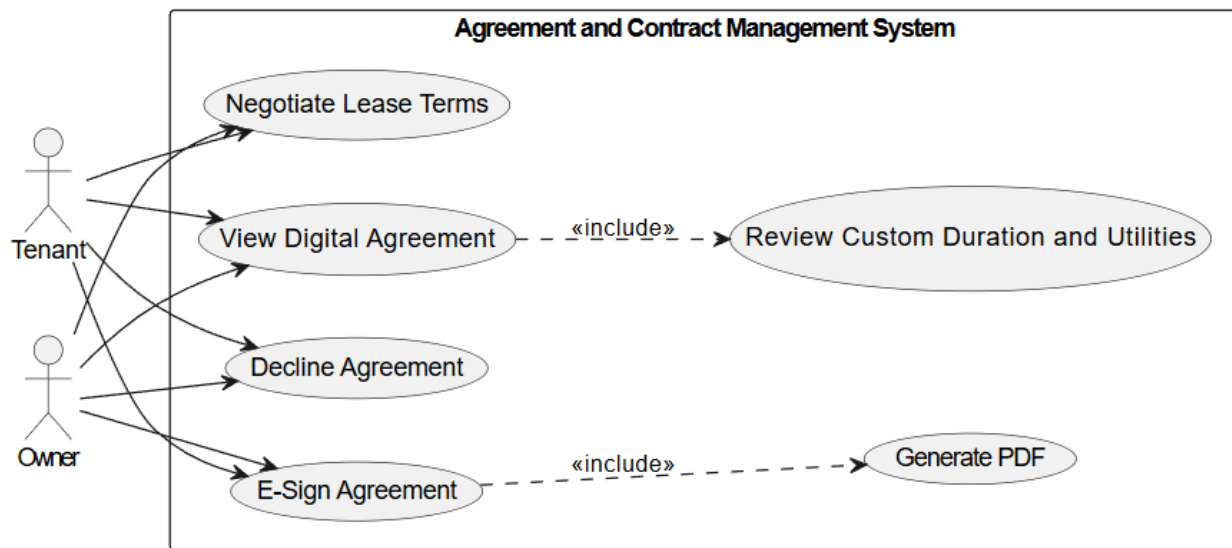
Figure 13: Class Diagram for BN

### 3.4. Agreement and Contract Management Subsystem (ACM)

**Subsystem Overview** Handles digital rental agreements with customizable terms and e-signatures after booking approval.

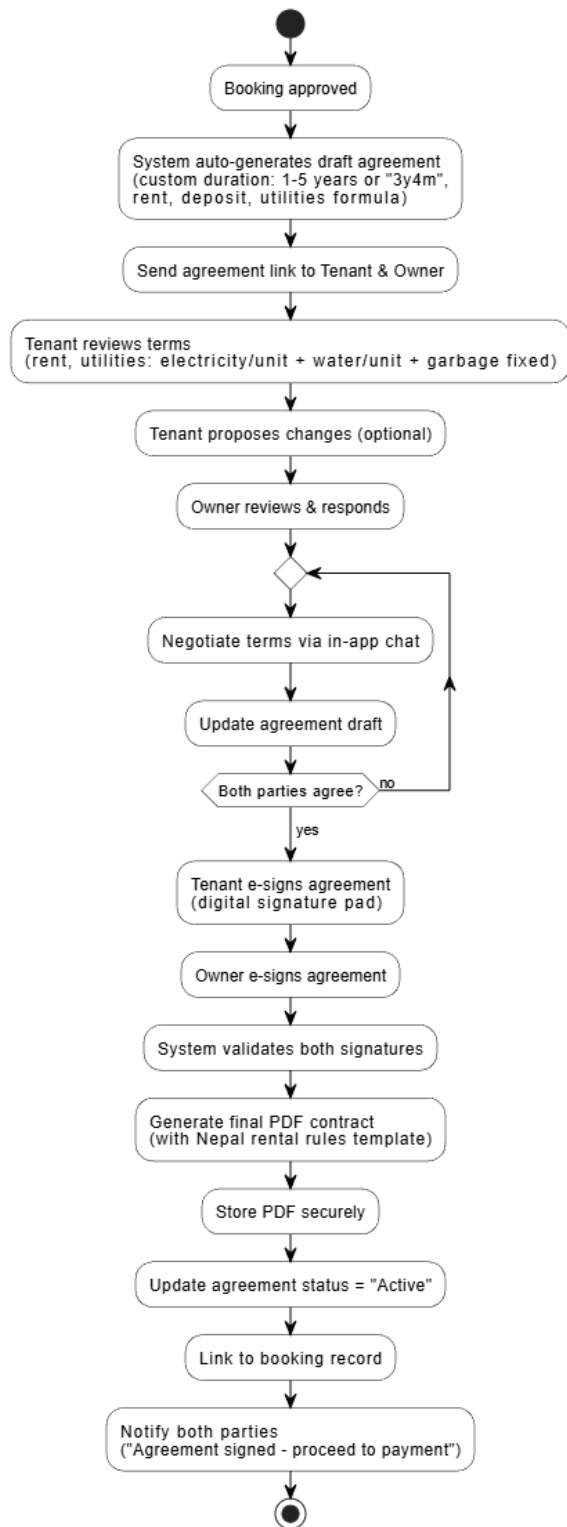
Req. Code	Requirement Description	Use Case / Notes
ACM-F-1.0	The system shall generate digital agreements with customizable terms (duration 1-5 years or custom, rent, utilities calculation: metered electricity/water per unit + fixed garbage).	Generate Agreement
ACM-F-2.0	Tenant and Owner shall review, negotiate terms, and e-sign the agreement.	Sign Agreement
ACM-F-3.0	Agreements shall be stored securely as PDF.	Store Agreement
ACM-NF-1.0	Agreements shall include standard rules/regulations template tailored to Nepal.	Compliance
ACM-UR-1.0	Agreement interface shall highlight changes and provide clear accept/decline options.	Usability

**Use Case Diagram for Agreement and Contract Management System**  
**GharBhada: Rental Management Platform**  
**Krishma Shrestha - 2431161**



*Figure 14: Use-Case Diagram for ACM*

**Activity Diagram - Agreement and Contract Management System**  
**GharBhada: Rental Management Platform**  
**Krishma Shrestha - 2431161**



*Figure 15:Activity Diagrams for ACM*

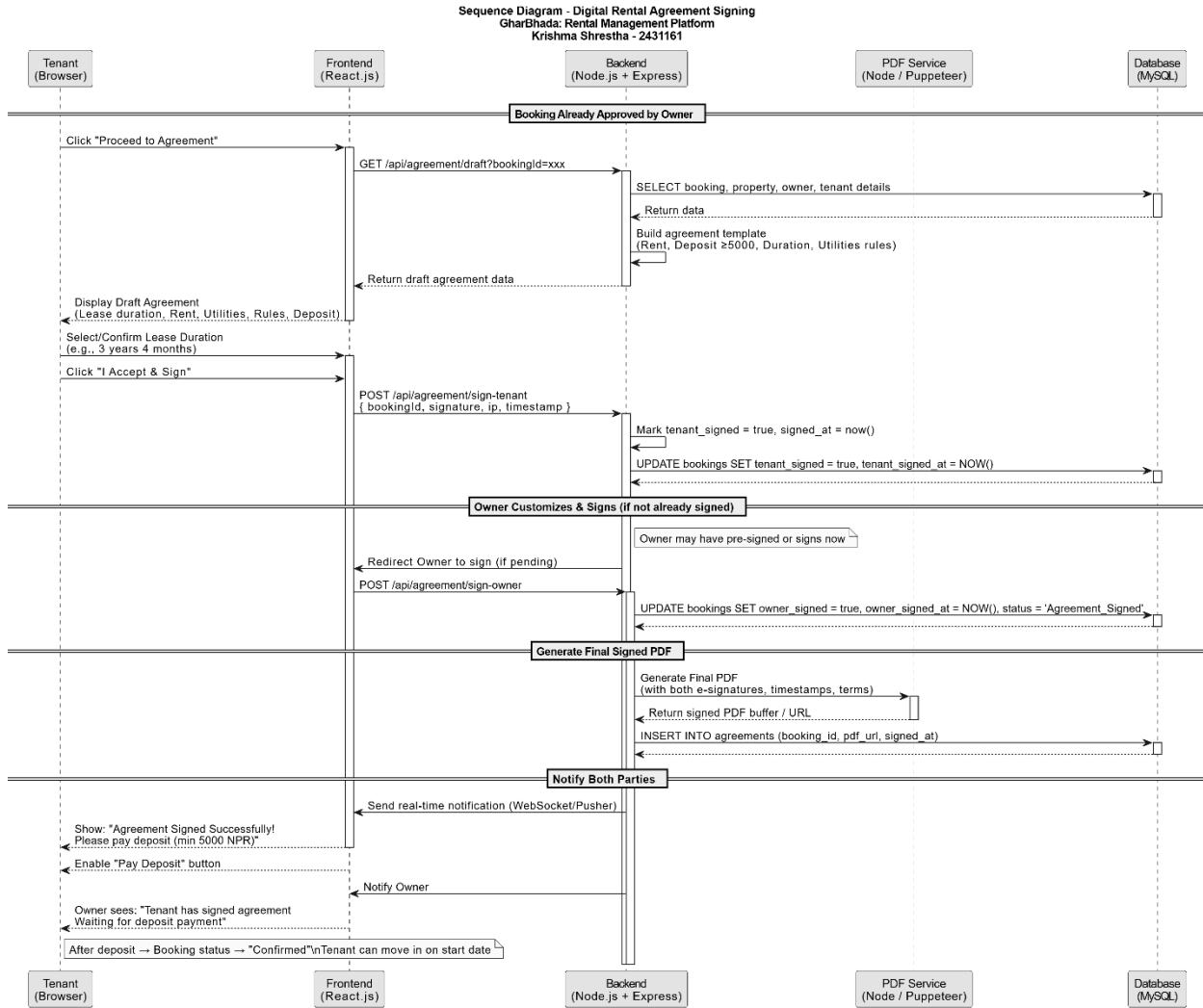


Figure 16: Sequence Diagrams for ACM

**Class Diagram - Agreement and Contract Management System**  
**GharBhada: Rental Management Platform**  
**Krishma Shrestha - 2431161**

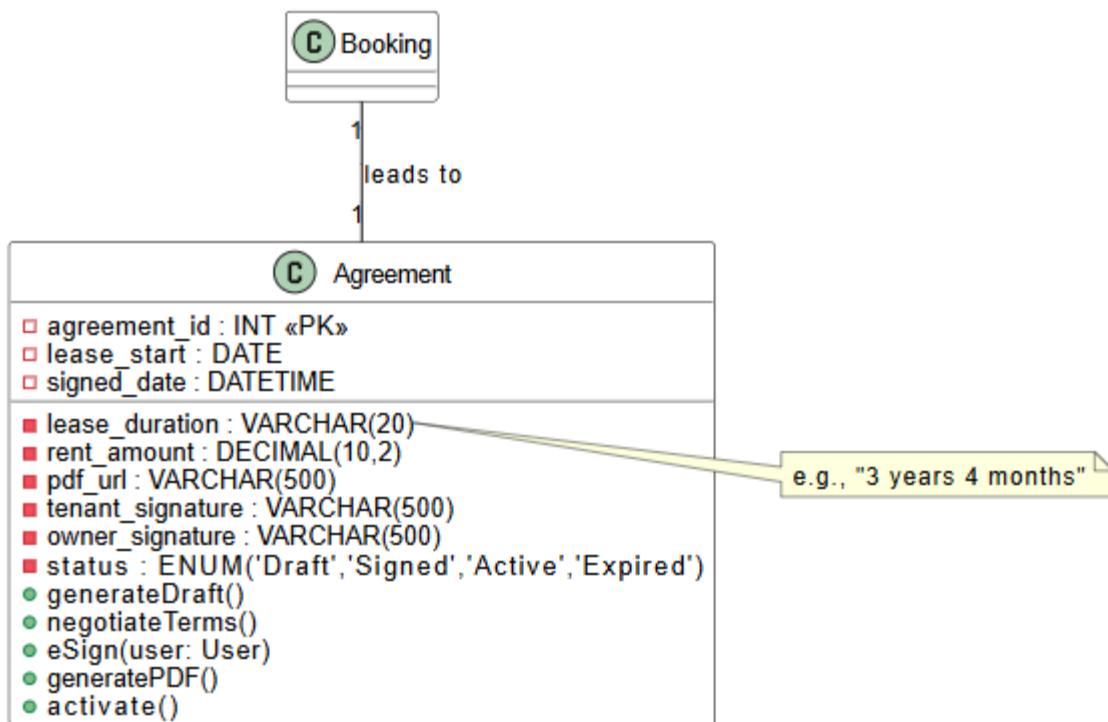


Figure 17: Class Diagram for ACM

### 3.5. Payment and Billing Subsystem (PMS)

**Subsystem Overview** Automates deposits, rent, and utility payments via local gateways (eSewa, Khalti). Includes reminders, receipts, and partial payments.

Req. Code	Requirement Description	Use Case / Notes
PMS-F-1.0	Tenants shall pay deposit (min. NPR 5,000) + first month's rent post-agreement approval.	Initial Payment
PMS-F-2.0	The system shall generate monthly bills (rent + utilities) with reminders.	Automated Billing
PMS-F-3.0	Tenants shall pay via integrated gateways; support partial/advance payments.	Pay Rent/Utilities
PMS-F-4.0	The system shall generate receipts and notify Owner upon successful payment.	Payment Confirmation
PMS-NF-1.0	Payments shall comply with PCI DSS and local regulations.	Security
PMS-NF-2.0	Payment processing shall complete within 5 seconds.	Performance
PMS-UR-1.0	Payment interface shall show clear breakdown (rent, utilities, deposit deduction).	Transparency

**Use Case Diagram for Payment and Billing System**  
**GharBhada: Rental Management Platform**  
**Krishma Shrestha - 2431161**

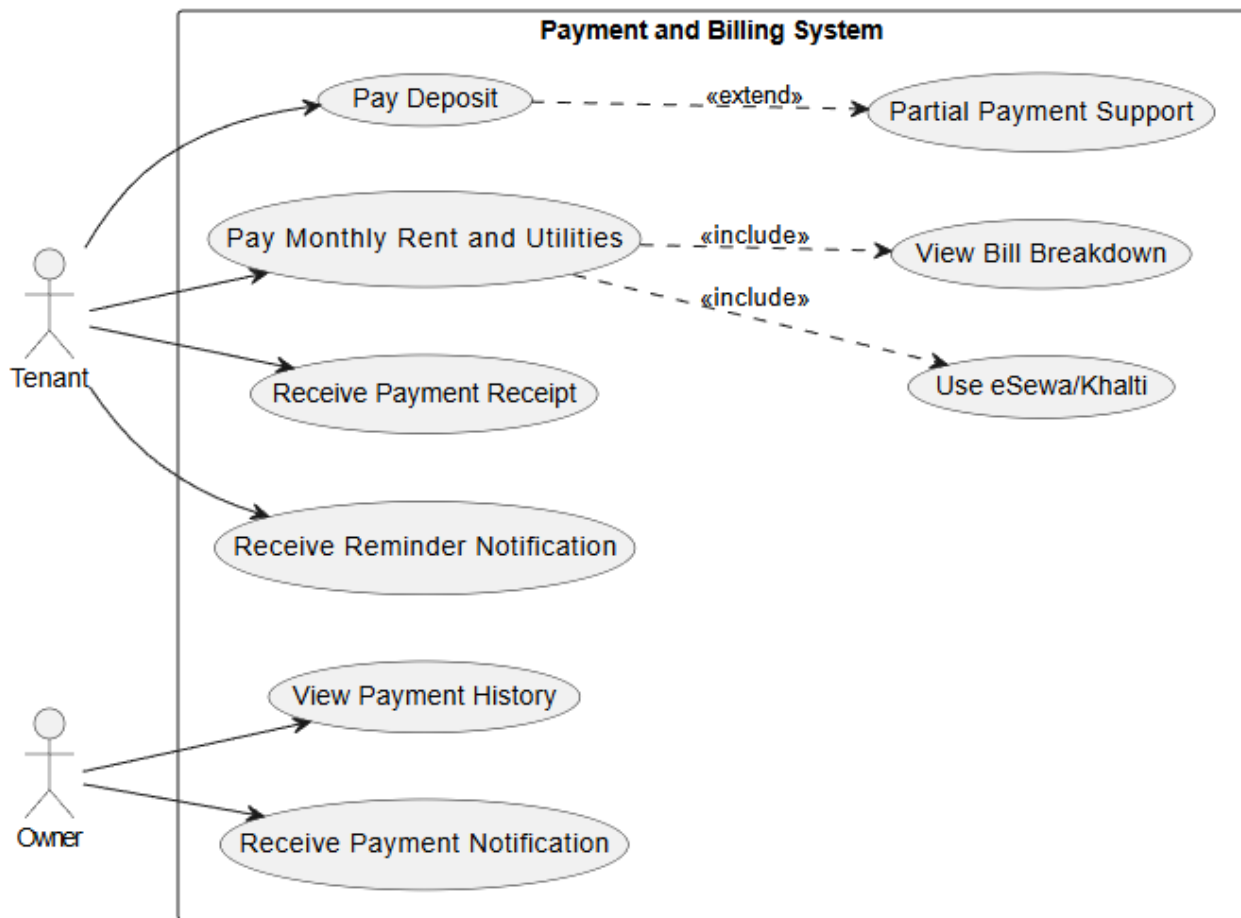


Figure 18: Use Case Diagram for PMS



**Activity Diagram - Payment and Billing System**  
**GharBhada: Rental Management Platform**  
**Krishma Shrestha - 2431161**

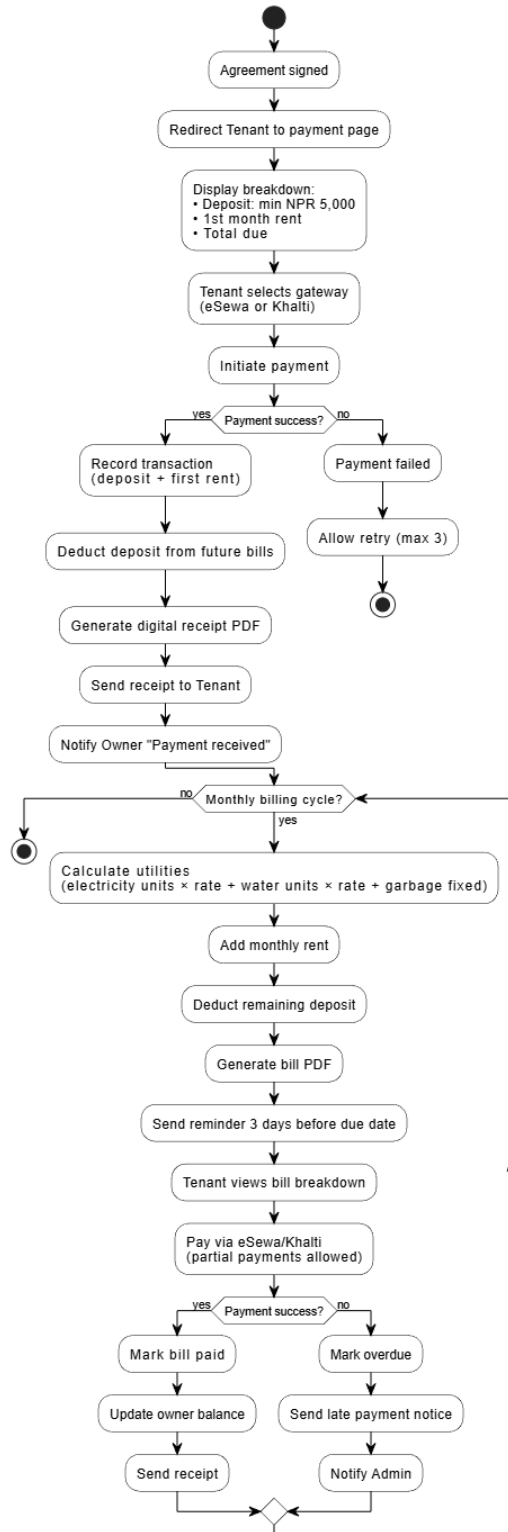


Figure 19: Activity Diagrams for PMS

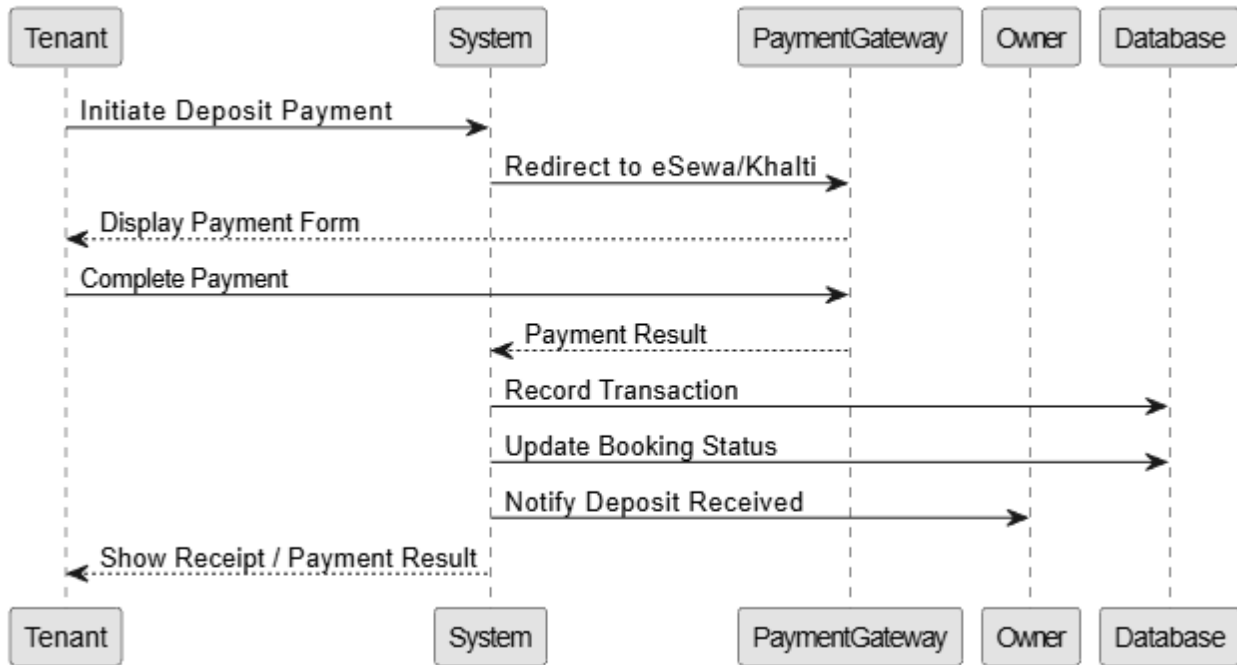


Figure 20: Sequence Diagrams for PMS

**Class Diagram - Payment and Billing System**  
**GharBhada: Rental Management Platform**  
**Krishma Shrestha - 2431161**

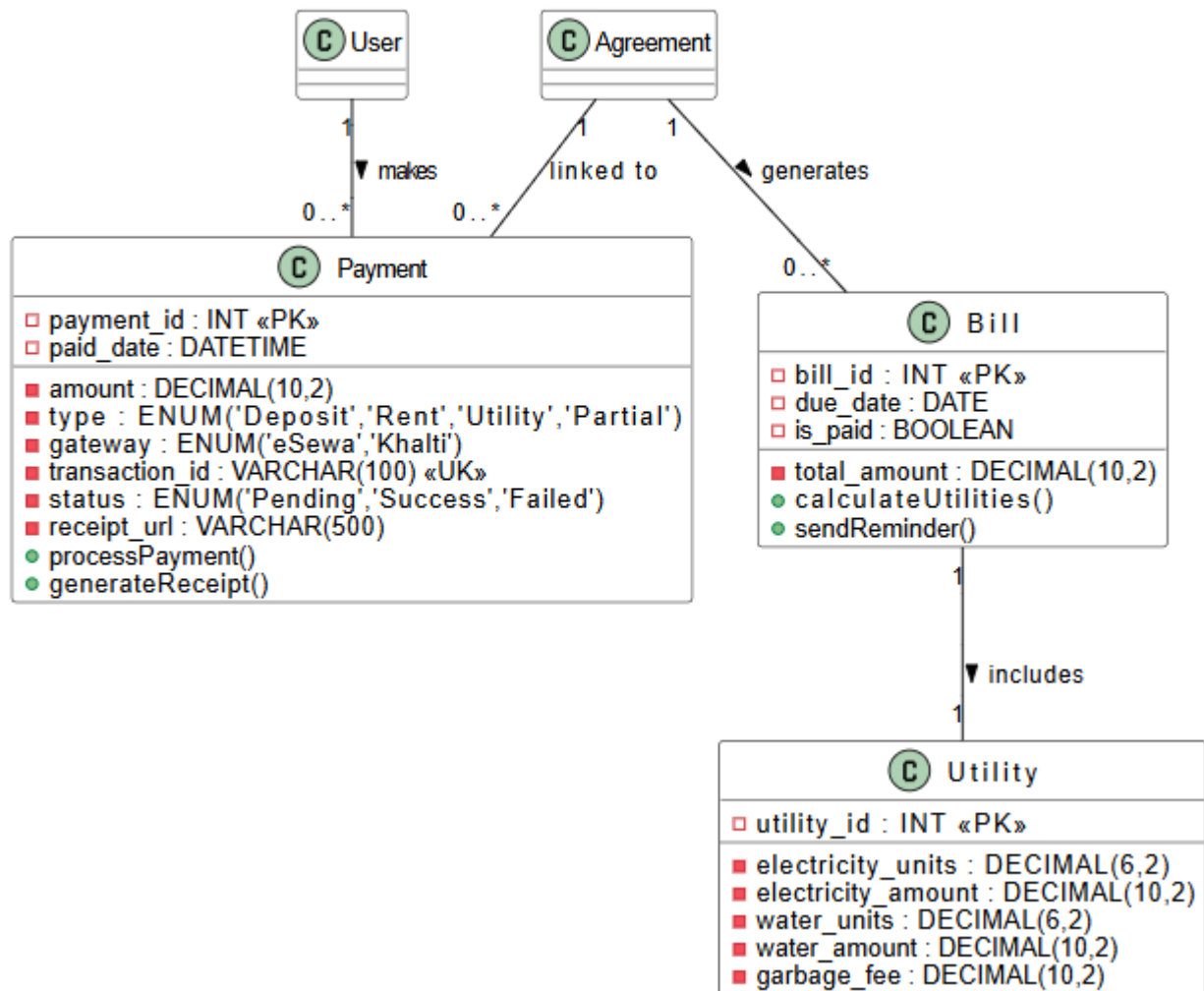


Figure 21: Class Diagram for PMS

### 3.6. Complaint and Maintenance Subsystem (CM) - Included for Completeness

**Subsystem Overview** Allows Tenants to raise complaints/maintenance requests, tracked with Owner/Admin intervention.

Req. Code	Requirement Description	Use Case / Notes
CM-F-1.0	Tenants shall submit complaints with description/photos.	Raise Complaint
CM-F-2.0	Owners/Admins shall view, update status, and resolve complaints.	Manage Complaints
CM-F-3.0	The system shall notify parties on updates and track resolution timeline.	Notification & Tracking

Use Case Diagram for Complaint and Maintenance System  
GharBhada: Rental Management Platform  
Krishma Shrestha - 2431161

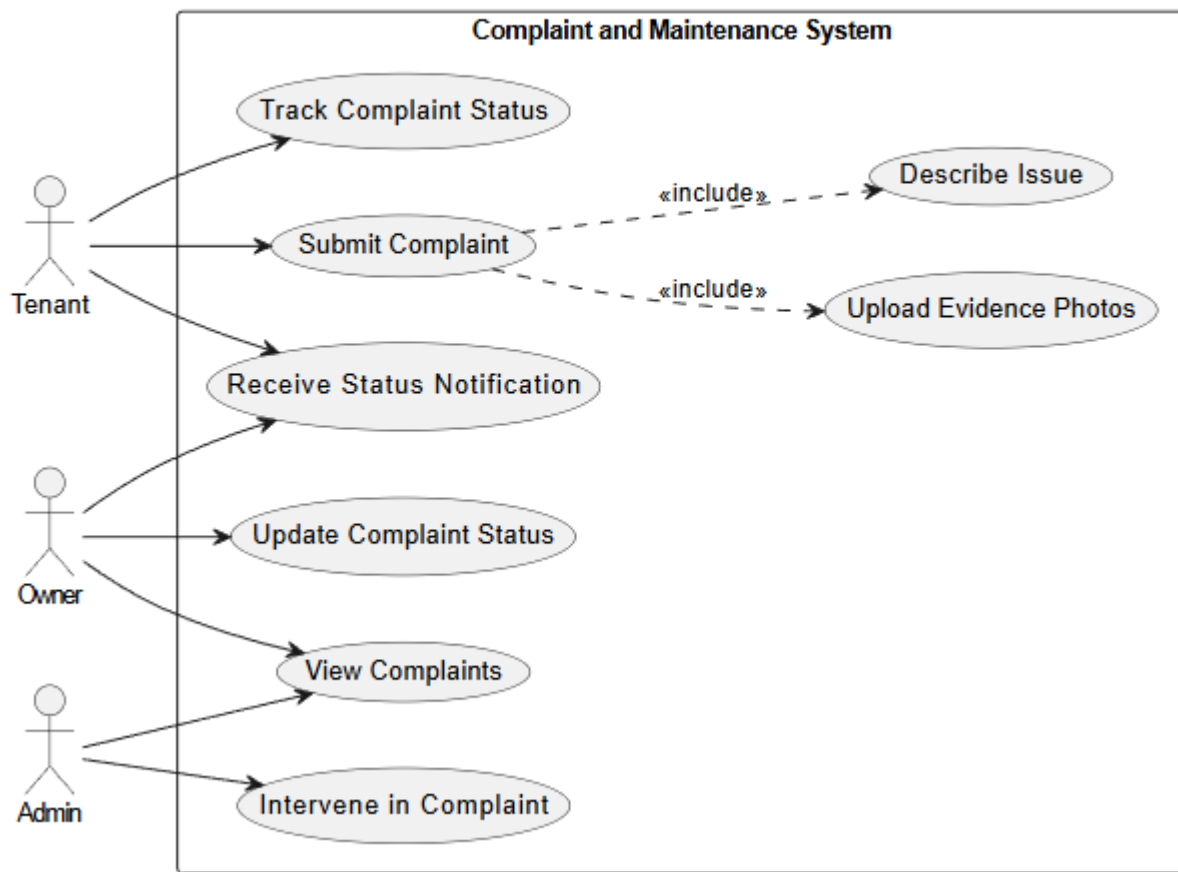


Figure 22: Use Case Diagram for CM

**Activity Diagram - Complaint and Maintenance System**  
**GharBhada: Rental Management Platform**  
**Krishma Shrestha - 2431161**

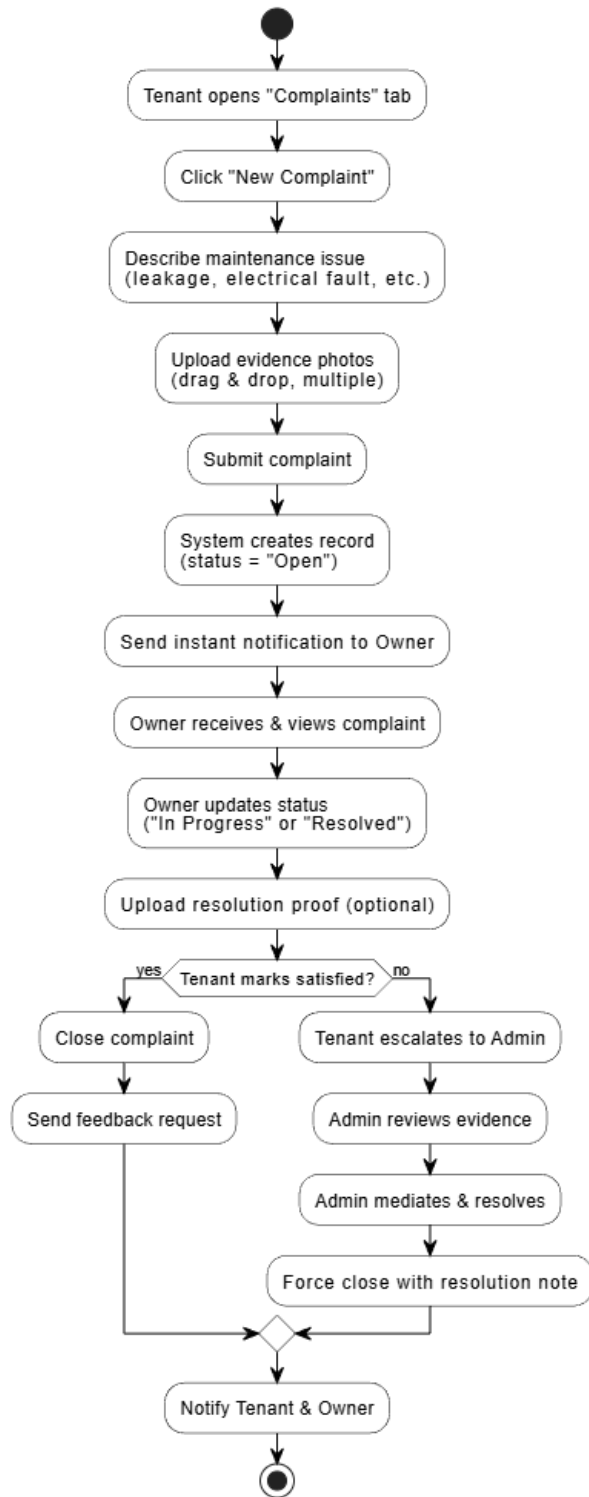


Figure 23: Activity Diagram for CM

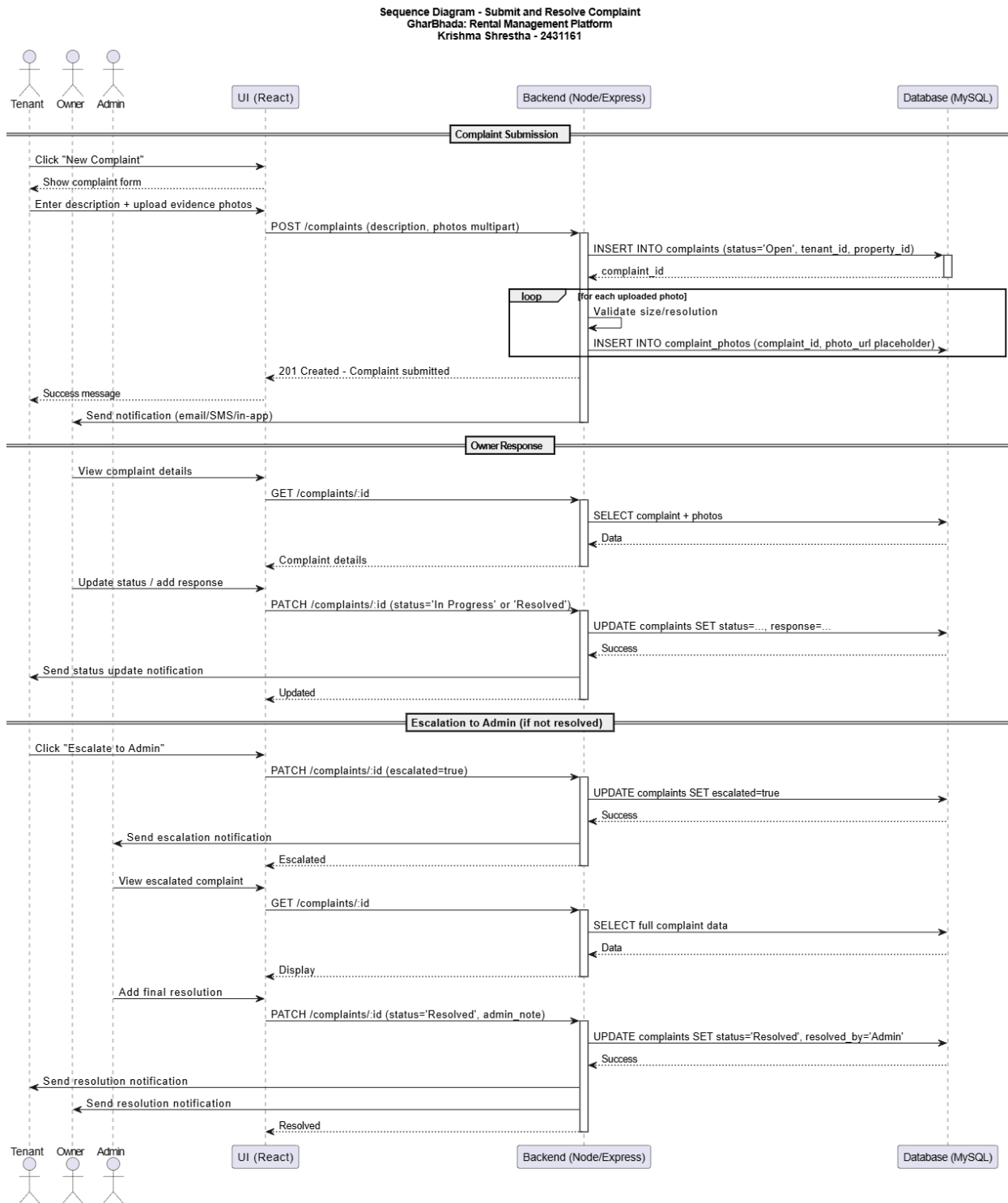


Figure 24: Sequence Diagrams for CM

**Class Diagram - Complaint and Maintenance System**  
**GharBhada: Rental Management Platform**  
**Krishma Shrestha - 2431161**

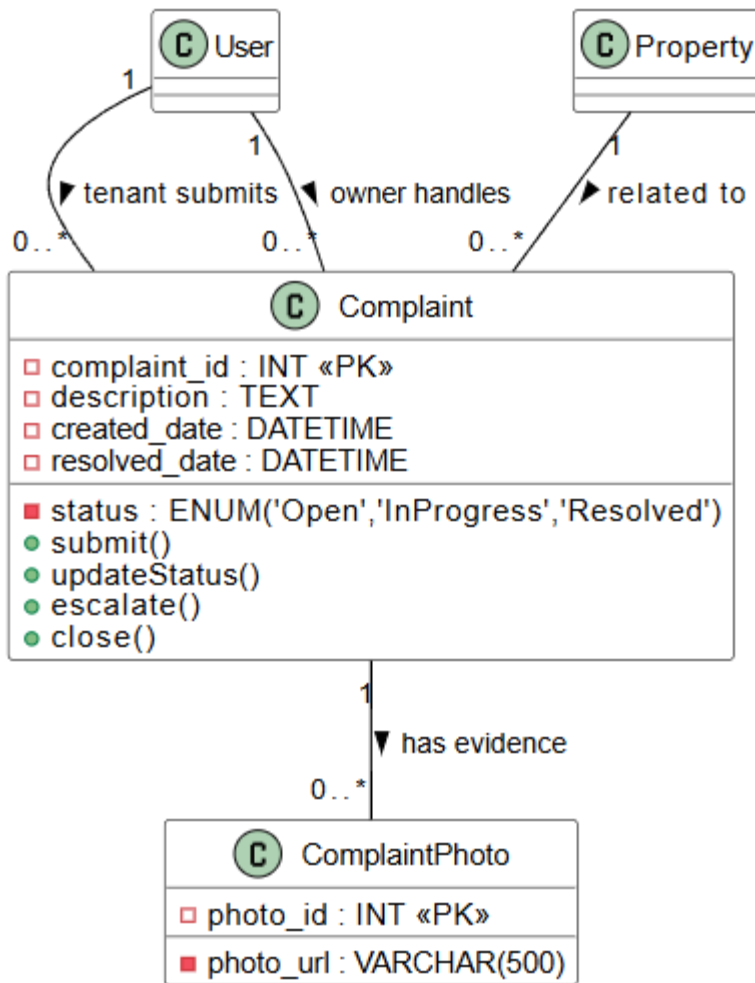


Figure 25: Class Diagrams for CM

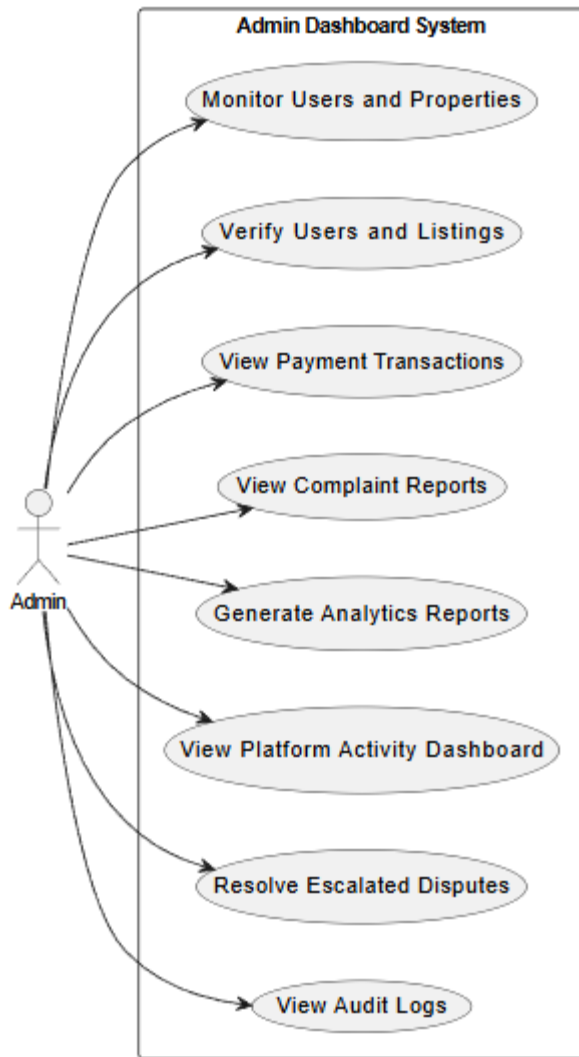


### 3.7. Admin Dashboard Subsystem

**Subsystem Overview** Provides oversight: monitor users/properties/payments/complaints, generate reports/analytics.

Req. Code	Requirement Description	Use Case / Notes
AD-F-1.0	Admin shall view dashboards with analytics (transactions, disputes, platform activity).	Monitor System
AD-F-2.0	Admin shall generate reports on payments, users, properties.	Reporting
AD-F-3.0	Admin shall intervene in escalated disputes/complaints.	Dispute Resolution

**Use Case Diagram for Admin Dashboard System**  
**GharBhada: Rental Management Platform**  
**Krishma Shrestha - 2431161**



*Figure 26: Use Case Diagram for AD*

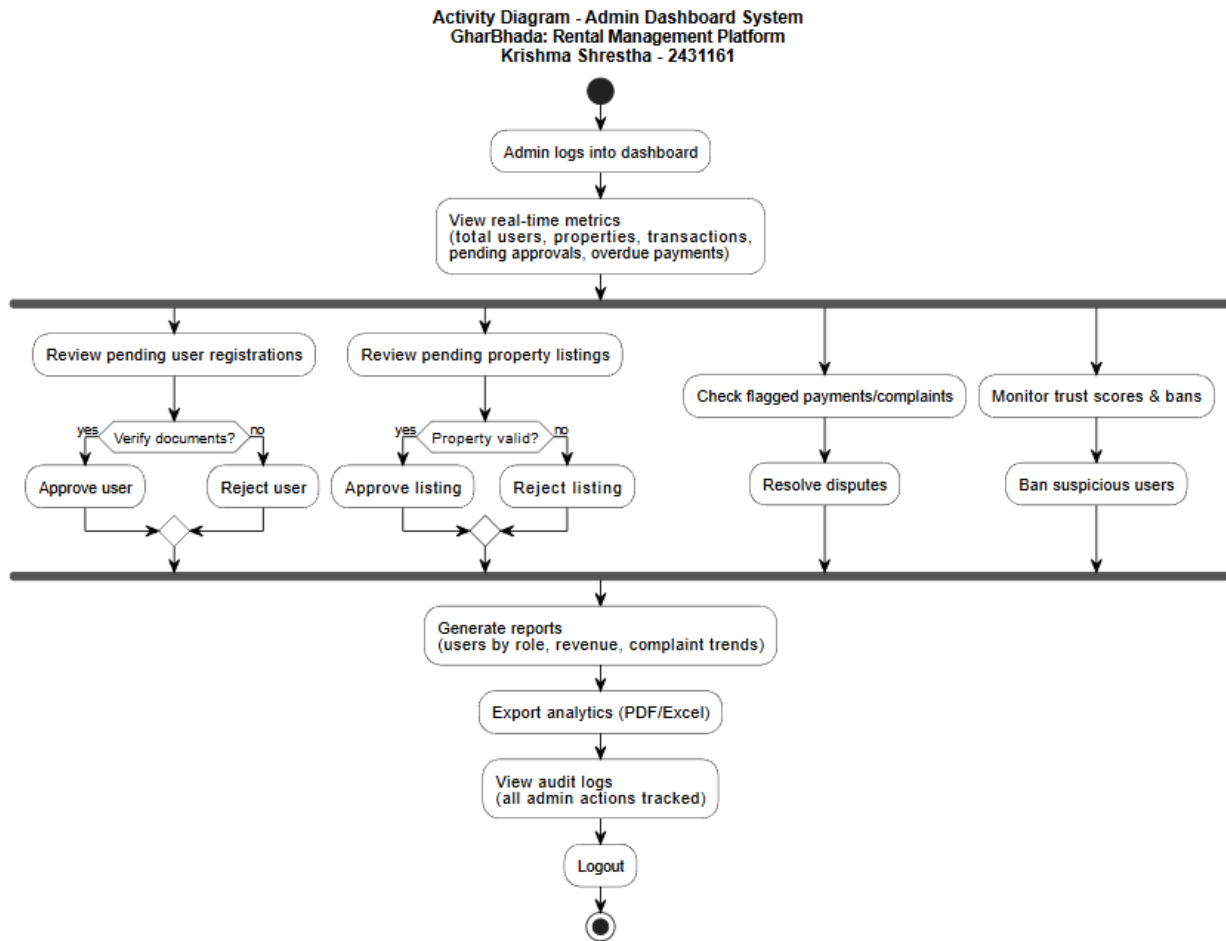


Figure 27: Activity Diagrams for AD

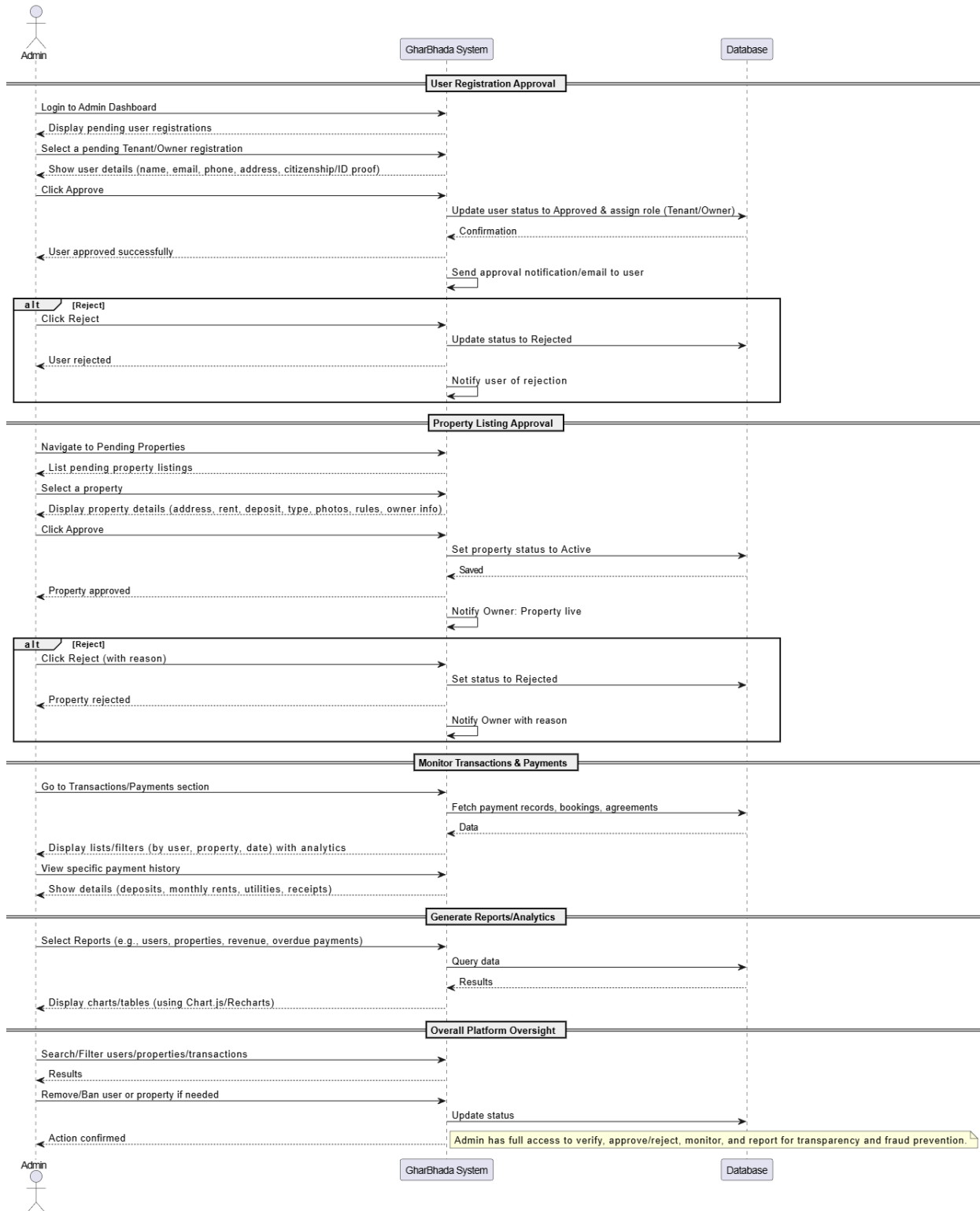


Figure 28: Sequence Diagrams for AD

**Class Diagram - Admin Dashboard System**  
**GharBhada: Rental Management Platform**  
**Krishma Shrestha - 2431161**

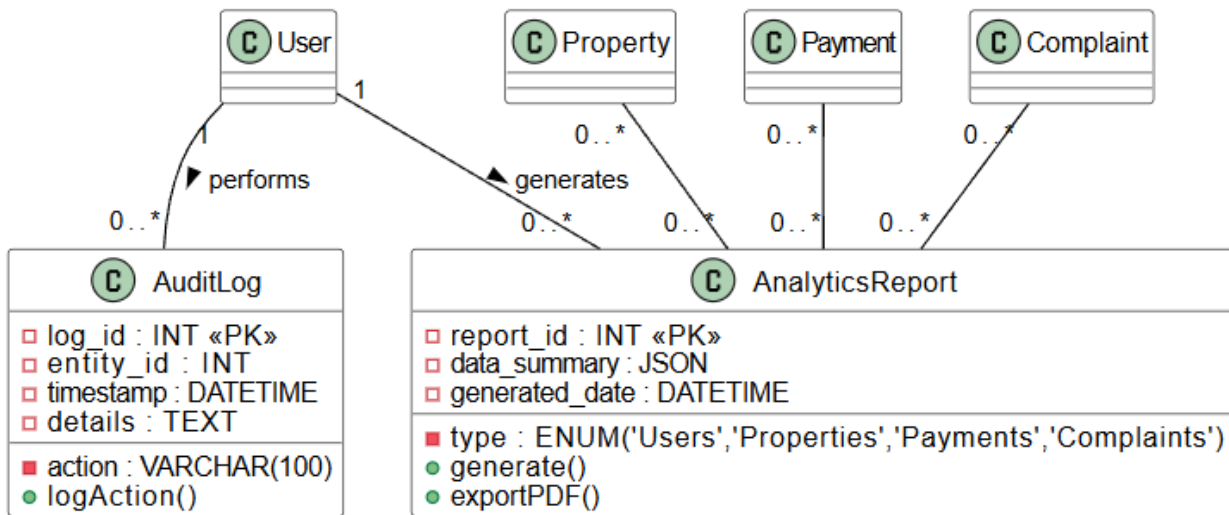


Figure 29: Class Diagram for AD

## 4. System Design (Whole System Level)

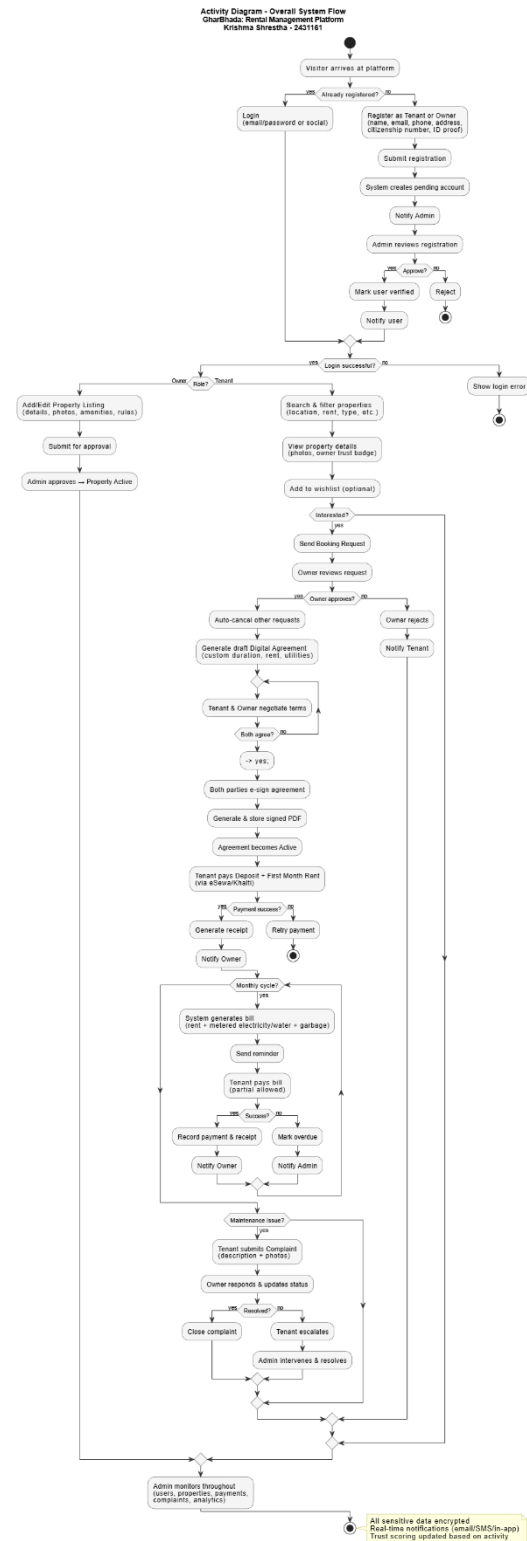


Figure 30: Activity Diagram for the Whole System

Use Case Diagram - Overall GharBhada System  
GharBhada Rental Management Platform  
Krishna Shrestha - 2431161

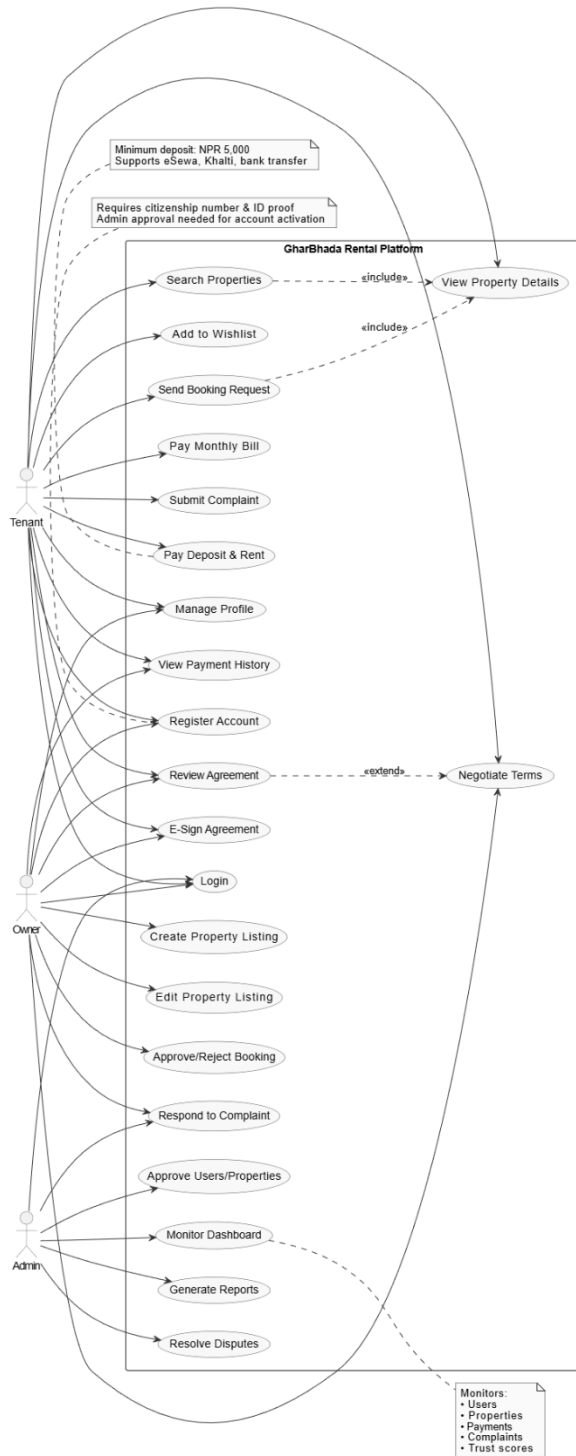


Figure 31: Use-Case Diagram for the Whole System

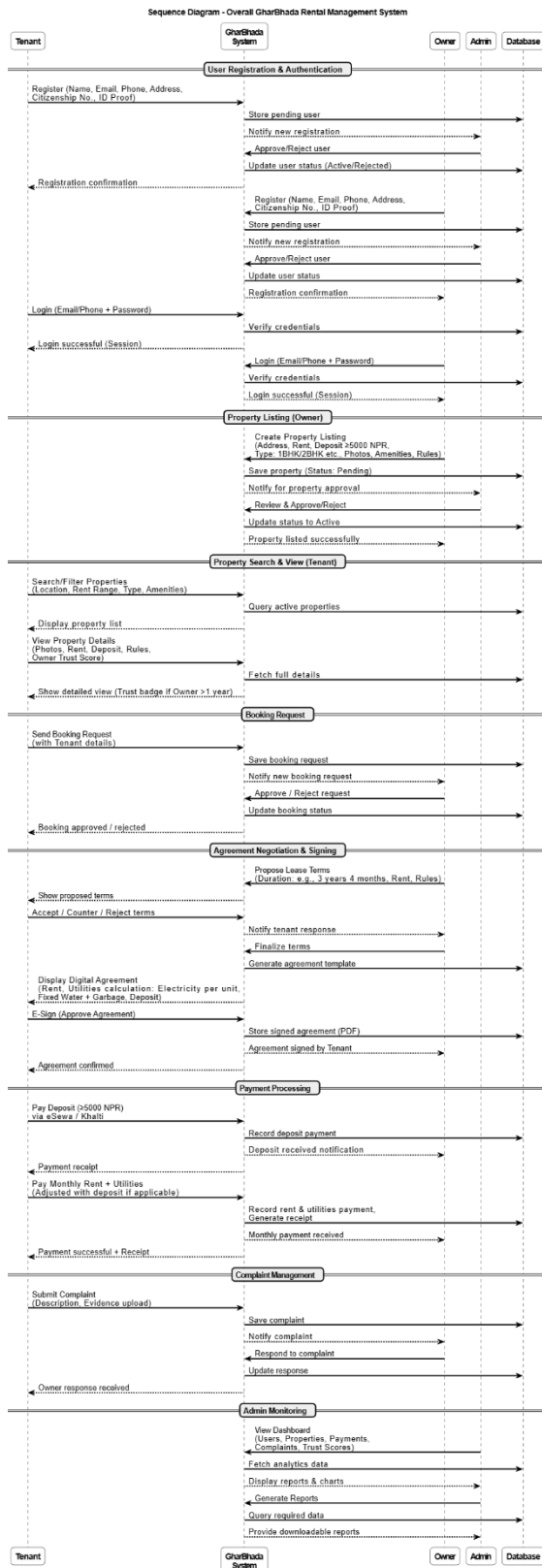


Figure 32: Sequence Diagram for the Whole System



Class Diagram - GharBhada Rental Management Platform  
Krishna Shrestha - 2431161

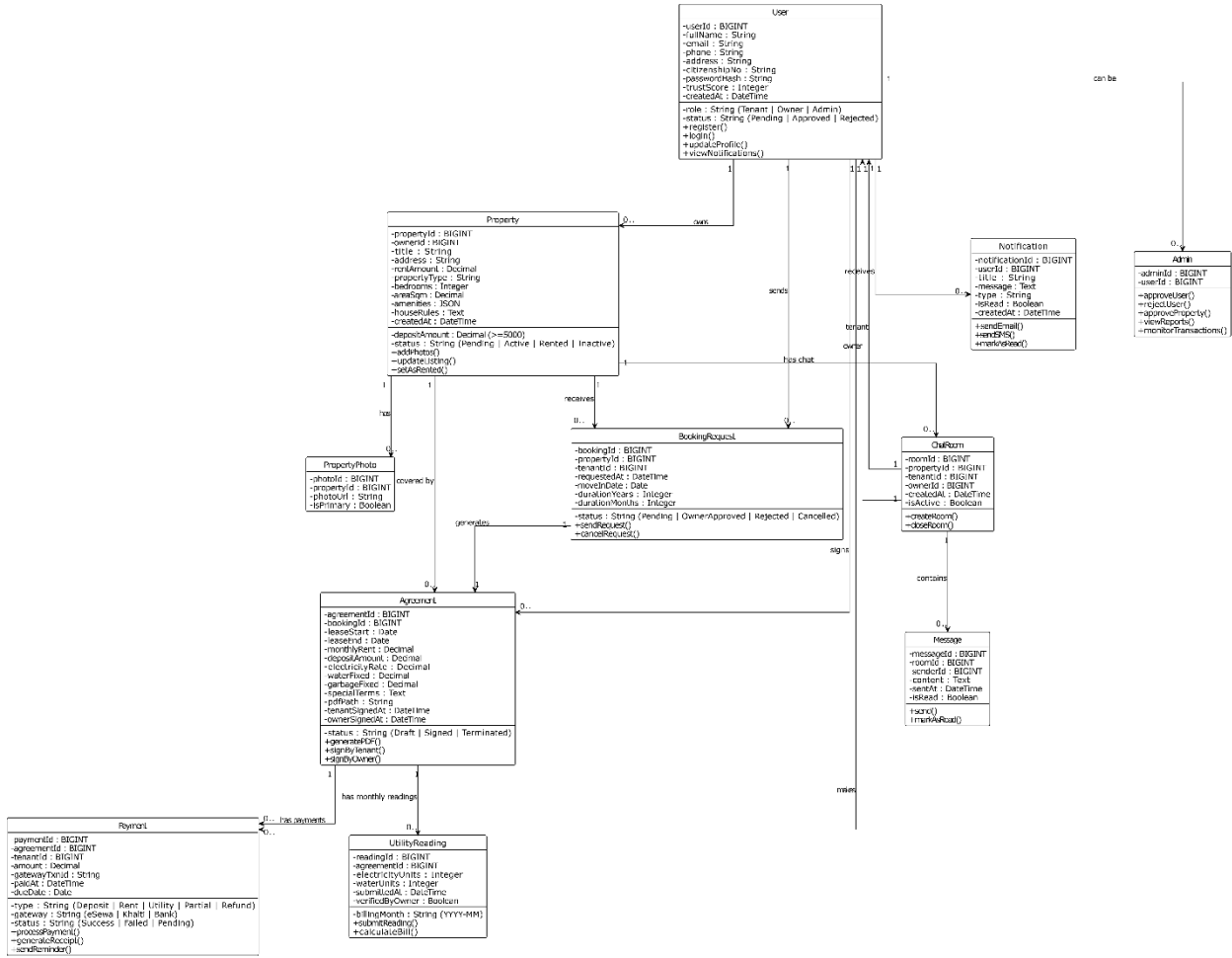


Figure 33: Class Diagram for the Whole System

## 5. Entity Relationship Diagram (ERD)

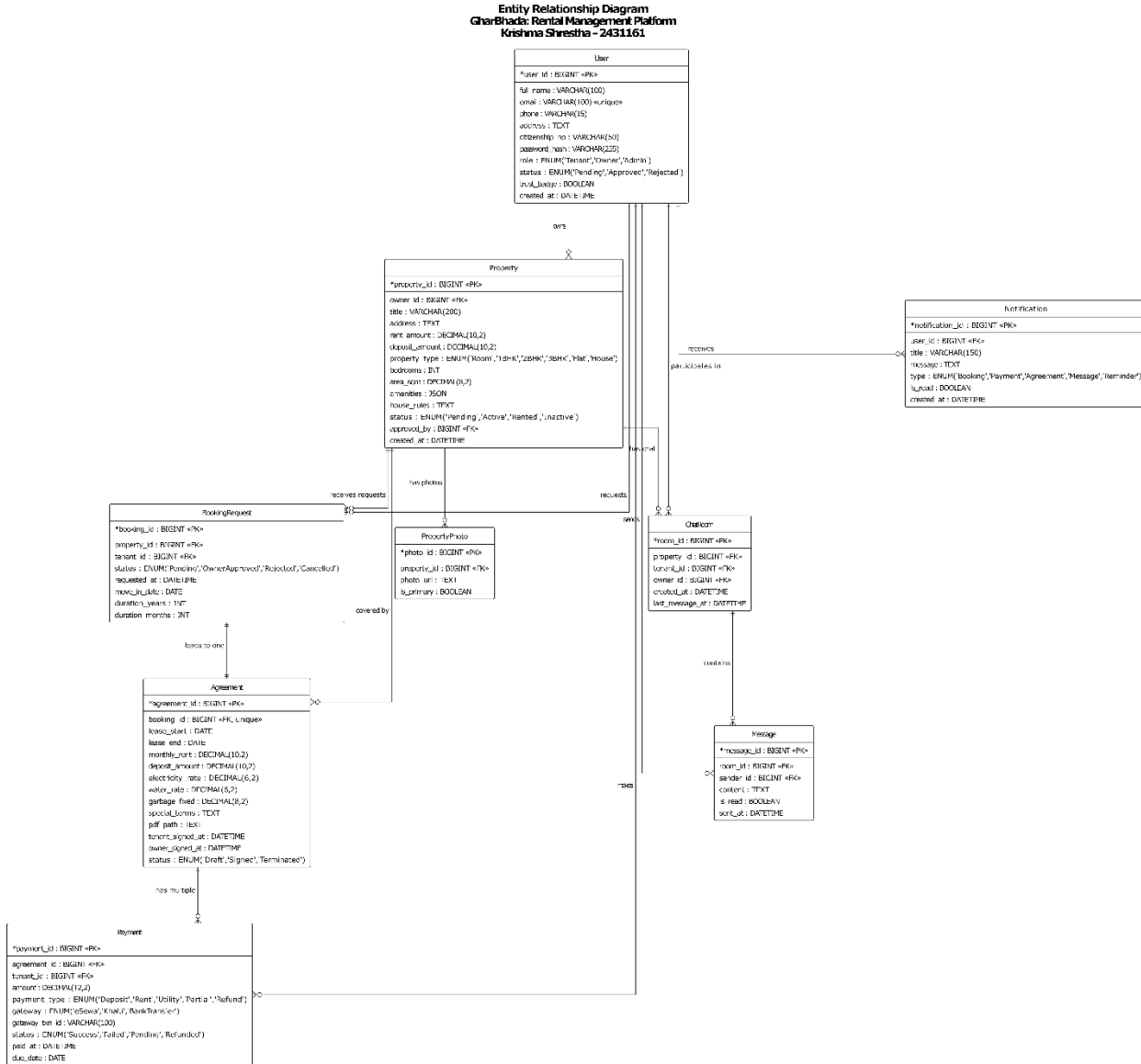


Figure 34: Entity Relationship Diagram (ERD)

## 6. Data Dictionary for GharBhada Rental Management Platform

The Data Dictionary provides a comprehensive definition of all key data elements (entities and attributes) across each subsystem. It is structured per subsystem, with tables listing **Field** (attribute name), **Type** (data type), **Null** (YES/NO), **Key** (PK=Primary Key, FK=Foreign Key, UK=Unique Key), and **Description**. Entities are derived from the SRS functional requirements, proposal objectives, subsystems descriptions, and full system flow (e.g., users, properties, bookings, payments, utilities, trust scoring, admin approvals). All entities support Nepal-specific features like NPR currency, citizenship numbers, eSewa/Khalti payments, and utility billing (metered electricity/water + fixed garbage).

### 1. User Management System (UM)

Entities: users (core for registration, login, profiles, roles, trust scoring, admin approvals/bans).

Field	Type	Null	Key	Description
user_id	INT	NO	PK	Unique identifier for each user (auto-increment).
name	VARCHAR(100)	NO		Full name of the user (Tenant/Owner/Admin).
email	VARCHAR(100)	NO	UK	Unique email for login and notifications.
phone	VARCHAR(20)	NO		Phone number for SMS notifications and verification.
address	TEXT	YES		Full residential address.
citizenship_number	VARCHAR(50)	NO	UK	Nepal citizenship/ID proof number for verification.
id_proof_url	VARCHAR(500)	YES		URL/path to uploaded ID proof image (min 1024x760, max 10MB).

password_hash	VARCHAR(255)	NO		Encrypted password (bcrypt, min 8 chars: alphabets, numbers, special chars).
role	ENUM('Tenant','Owner','Admin')	NO		Role-based access (Tenant, Owner, Admin).
trust_score	DECIMAL(3,2)	YES		Trust score (0-5.00); "Trusted" badge if >4.0 or active >1 year.
is_verified	BOOLEAN	NO		TRUE if admin-approved; FALSE for pending/new accounts.
is_banned	BOOLEAN	NO		TRUE if admin-banned; FALSE otherwise.
registration_date	DATETIME	NO		Timestamp of user registration.
last_activity	DATETIME	YES		Timestamp of last login/activity for trust calculation.
social_provider	VARCHAR(50)	YES		Google/Facebook/Twitter if social registration used.

## 2. Property Listing and Search (PRM)

Entities: properties (listings with details, photos, status), property\_categories (types like 1BHK), property\_photos (multiple images), wishlists (tenant favorites).

Field	Type	Null	Key	Description
<b>properties</b>				
property_id	INT	NO	PK	Unique property identifier.
owner_id	INT	NO	FK (users.user_id)	References property owner.
category_id	INT	NO	FK (property_categories.category_id)	Property type (1BHK, etc.).
address	TEXT	NO		Full property address (with map integration).
rent_amount	DECIMAL(10,2)	NO		Monthly rent in NPR.
deposit_amount	DECIMAL(10,2)	NO		Minimum NPR 5,000 deposit.
type	VARCHAR(50)	NO		Flat/Apartment/House/Room.
amenities	JSON	YES		Array of amenities (e.g., ["AC", "Parking"]).
rules	TEXT	YES		House rules/description (max 100 words for categories).
description	TEXT	YES		Detailed property info.

status	ENUM('Pending','Active','Inactive')	NO		Listing status post-admin approval.
created_date	DATETIME	NO		Listing creation timestamp.
<b>property_categories</b>				
category_id	INT	NO	PK	Unique category ID (e.g., 1BHK).
category_name	VARCHAR(50)	NO	UK	Name like "1BHK", "2BHK".
description	VARCHAR(500)	YES		Short description (max 100 words).
<b>property_photos</b>				
photo_id	INT	NO	PK	Unique photo ID.
property_id	INT	NO	FK (properties.property_id)	Linked property.
image_url	VARCHAR(500)	NO		Photo path (min 1024x760px, max 10MB).
is_primary	BOOLEAN	NO		TRUE for main photo.
<b>wishlists</b>				
wishlist_id	INT	NO	PK	Unique wishlist entry.
tenant_id	INT	NO	FK (users.user_id)	Tenant who added.

property_id	INT	NO	FK (properties.property_id)	Wishlisted property.
-------------	-----	----	--------------------------------	----------------------

### 3. Booking and Notification System (BN)

Entities: bookings (requests, approvals), notifications (email/SMS/in-app).

Field	Type	Null	Key	Description
<b>bookings</b>				
booking_id	INT	NO	PK	Unique booking identifier.
tenant_id	INT	NO	FK (users.user_id)	Requesting tenant.
property_id	INT	NO	FK (properties.property_id)	Booked property.
owner_id	INT	NO	FK (users.user_id)	Property owner.
status	ENUM('Pending','Approved','Rejected','Cancelled')	NO		Booking workflow status.
request_date	DATETIME	NO		Booking request timestamp.
approval_date	DATETIME	YES		Owner approval timestamp (auto-cancel others).
tenant_details	JSON	YES		Additional tenant info submitted with request.

<b>notifications</b>				
notification_id	INT	NO	PK	Unique notification ID.
user_id	INT	NO	FK (users.user_id)	Recipient user.
type	ENUM('Booking_Update', 'Payment', 'Complaint')	NO		Notification category.
title	VARCHAR(200)	NO		Notification title.
message	TEXT	NO		Full message body.
is_read	BOOLEAN	NO		TRUE if read.
sent_date	DATETIME	NO		Delivery timestamp (<10s SLA).
channel	ENUM('Email', 'SMS', 'InApp')	NO		Delivery method.



#### 4. Agreement and Contract Management (ACM)

Entities: agreements (digital contracts, e-signatures).

Field	Type	Null	Key	Description
agreement_id	INT	NO	PK	Unique agreement ID.
booking_id	INT	NO	FK (bookings.booking_id)	Linked booking.
tenant_id	INT	NO	FK (users.user_id)	Tenant signer.
owner_id	INT	NO	FK (users.user_id)	Owner signer.
lease_start	DATE	NO		Agreement start date.
lease_duration	VARCHAR(20)	NO		Custom duration (e.g., "3 years 4 months", 1-5 years).
rent_amount	DECIMAL(10,2)	NO		Agreed monthly rent (NPR).
pdf_url	VARCHAR(500)	NO		Secure PDF storage path.

tenant_signature	VARCHAR(500)	YES		E-signature hash/image.
owner_signature	VARCHAR(500)	YES		E-signature hash/image.
status	ENUM('Draft','Signed','Active','Expired')	NO		Agreement lifecycle.
signed_date	DATETIME	YES		Final signature timestamp.

## 5. Payment and Billing System (PMS)

Entities: payments (deposits, rent, utilities), bills (monthly invoices), utilities (metered + fixed).

Field	Type	Null	Key	Description
<b>payments</b>				
payment_id	INT	NO	PK	Unique payment ID.
user_id	INT	NO	FK (users.user_id)	Payer (tenant).
agreement_id	INT	YES	FK (agreements.agreement_id)	Linked agreement.
property_id	INT	YES	FK (properties.property_id)	Property reference.
amount	DECIMAL(10,2)	NO		Payment amount (NPR).
type	ENUM('Deposit','Rent','Utility','Partial')	NO		Payment category.
gateway	ENUM('eSewa','Khalti')	NO		Payment gateway used.
transaction_id	VARCHAR(100)	NO	UK	Gateway transaction

				n reference.
status	ENUM('Pending','Success','Failed')	NO		Payment status.
receipt_url	VARCHAR(500)	YES		Generated receipt PDF.
paid_date	DATETIME	YES		Payment timestamp.
<b>bills</b>				
bill_id	INT	NO	PK	Unique bill ID.
agreement_id	INT	NO	FK (agreements.agreement_id)	Linked agreement.
total_amount	DECIMAL(10,2)	NO		Rent + utilities (after deposit deduction ).
due_date	DATE	NO		Monthly due date.
is_paid	BOOLEAN	NO		Payment status.

<b>utilities</b>				
utility_id	INT	NO	PK	Unique utility record.
bill_id	INT	NO	FK (bills.bill_id)	Linked bill.
electricity_units	DECIMAL(6,2)	YES		Metered electricity units.
electricity_amount	DECIMAL(10,2)	YES		Calculated cost per unit.
water_units	DECIMAL(6,2)	YES		Metered water units.
water_amount	DECIMAL(10,2)	YES		Calculated cost.
garbage_fee	DECIMAL(10,2)	NO		Fixed garbage fee.

## 6. Complaint and Maintenance Subsystem (CM) - Added for Completeness

Entities: complaints (tenant reports, tracking).

Field	Type	Null	Key	Description
complaint_id	INT	NO	PK	Unique complaint ID.
tenant_id	INT	NO	FK (users.user_id)	Reporting tenant.
property_id	INT	NO	FK (properties.property_id)	Affected property.
owner_id	INT	NO	FK (users.user_id)	Notified owner.
description	TEXT	NO		Issue description.
photo_urls	JSON	YES		Array of evidence photos.
status	ENUM('Open','InProgress','Resolved')	NO		Resolution status.
created_date	DATETIME	NO		Report timestamp.
resolved_date	DATETIME	YES		Closure timestamp.

## 7. Admin Dashboard Subsystem (AD)

Entities: analytics\_reports (summaries), audit\_logs (tracking). References all prior entities for monitoring.

Field	Type	Null	Key	Description
<b>analytics_reports</b>				
report_id	INT	NO	PK	Unique report ID.
admin_id	INT	NO	FK (users.user_id)	Generating admin.
type	ENUM('Users','Properties','Payments','Complaints')	NO		Report category.
data_summary	JSON	NO		Aggregated metrics (e.g., total transactions).
generated_date	DATETIME	NO		Report creation timestamp.
<b>audit_logs</b>				
log_id	INT	NO	PK	Unique log entry.

user_id	INT	YES	FK (users.user_id)	Acting user (or NULL for system).
action	VARCHAR(100)	NO		Action like "User Approved", "Property Verified".
entity_id	INT	YES		Affected entity ID (e.g., user_id).
timestamp	DATETIME	NO		Log timestamp.
deails	TEXT	YES		Additional context.



## 7. Test Cases

I created these test cases to validate the features and functionality of the **GharBhada** application.

The detailed test cases are provided in the Excel file: [Test Case of Gharbahada](#)

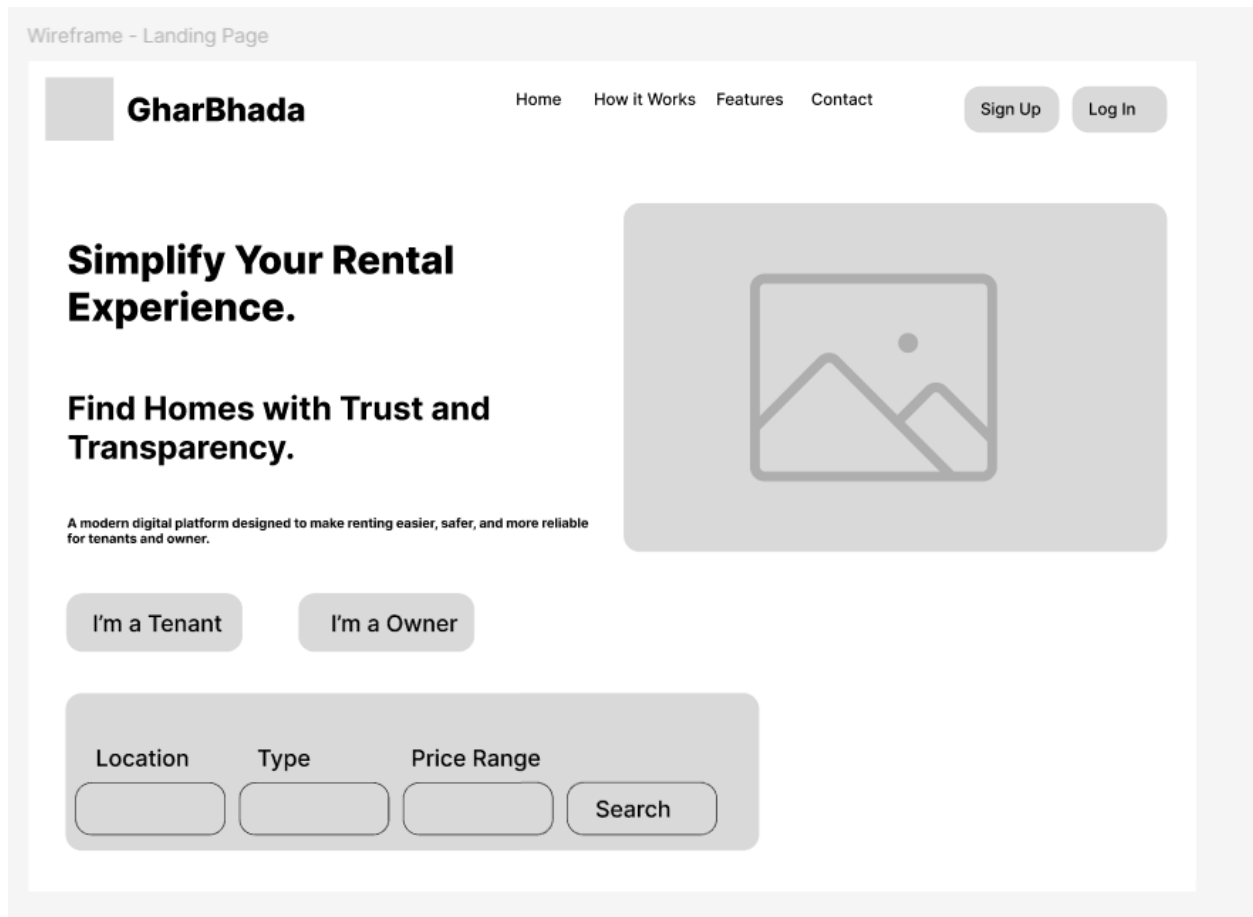
	A	B	C	D	E	F	G
1	TC ID	Subsystem	Description	Steps	Expected Result	Actual Results	Status
2	UMS-TC-1	User Management	Tenant Registration with Citizenship ID & Image Upload	Navigate to homepage, click "Sign Up as Tenant", fill Name, Email, Phone, Address, Password, Confirm Password, enter Citizenship ID number, upload valid citizenship image (JPG/PDF), click Submit	The user account is created successfully with status "Pending Approval". Success message shown: "Registration successful, awaiting admin approval". User appears in admin pending list.	Success message displayed correctly. New tenant user appears in admin pending approvals with all details including uploaded citizenship image.	Pass
3	UMS-TC-2	User Management	Owner Registration with Citizenship ID & Image Upload	Navigate to homepage, click "Sign Up as Owner", fill all required fields including Citizenship ID, upload valid citizenship image, click Submit	The user account is created successfully with status "Pending Approval". Success message shown and user appears in admin pending list.	Success message displayed correctly. New owner user appears in admin pending approvals.	Pass
4	UMS-TC-3	User Management	Registration – Invalid Citizenship Image Upload	Go to registration page (Tenant or Owner), fill all fields correctly, upload invalid file (e.g., .exe, >5MB, or non-image/PDF), click Submit	Error message displayed: "Invalid file type or size. Only JPG/PDF allowed, max 5MB". Form not submitted.	Error message shown correctly for invalid type/size. Form submission blocked.	Pass
5	UMS-TC-4	User Management	Registration – Missing Citizenship ID or Image	Go to registration page (Tenant or Owner), fill all fields correctly, upload invalid file (e.g., .exe, >5MB, or non-image/PDF), click Submit	Error message for missing fields: "Citizenship ID and proof image are required". Form not submitted.	Required field errors displayed clearly. Form not submitted.	Pass
6	UMS-TC-5	User Management	Successful Login After Admin Approval	Register as Tenant/Owner (complete UMS-TC-1 or UMS-TC-2), login as Admin and approve the user, go to login page, select	User is logged in successfully and redirected to respective dashboard (Tenant or Owner).	Redirected correctly to Tenant/Owner dashboard after approval.	Pass
7	UMS-TC-6	User Management	Login Attempt Before Admin Approval	Complete registration (UMS-TC-1), immediately try to login with the new credentials	Error message: "Your account is pending admin approval". Login denied.	Exact pending approval message shown. Login blocked.	Pass
8	UMS-TC-7	User Management	Password Mismatch During Registration	Go to registration page, fill all fields correctly, enter different password in Confirm Password field, click Submit	Error message: "Passwords do not match". Form not submitted.	Error message displayed instantly. Submission prevented.	Pass
9	ADM-TC-1	Admin Panel	Admin Views Pending Registrations with Citizenship Proof	Login as Admin, navigate to Pending Approvals section	List of pending users displayed with Name, Email, Phone, Citizenship ID, and clickable/viewable citizenship image proof.	All pending users listed with full details. Citizenship image preview/lightbox works perfectly.	Pass
10	ADM-TC-2	Admin Panel	Admin Approves User Registration	Login as Admin, go to Pending Approvals, select a pending user, click Approve	User status updated to "Approved". User can now login successfully.	Status changed to Approved instantly. Approved user logs in without issues.	Pass
11	ADM-TC-3	Admin Panel	Admin Rejects User Registration	Login as Admin, go to Pending Approvals, select a pending user, click Reject (optional reason)	User status updated to "Rejected". User removed from pending list.	Rejection successful. User removed from pending list.	Pass
12	ADM-TC-4	Admin Panel	Admin Suspends an Approved User	Login as Admin, go to User Management, search approved user, click Suspend	User status changed to "Suspended". Suspended user cannot login anymore.	Suspend applied correctly. Suspended user receives access denied on login attempt.	Pass
13	DASH-TC-1	Dashboard	Tenant Dashboard Access After Approval	Complete successful registration and admin approval (UMS-TC-1 + ADM-TC-2), login as Tenant	Tenant dashboard loads successfully with teal theme, sidebar menu, and welcome section.	Dashboard loads quickly with full responsive layout and teal theme.	Pass
14	DASH-TC-2	Dashboard	Owner Dashboard Access After Approval	Complete successful registration and admin approval (UMS-TC-2 + ADM-TC-2), login as Owner	Owner dashboard loads successfully with relevant sections visible.	Dashboard loads successfully with owner menu options visible.	Pass
15	SEC-TC-1	Security	Forgot Password Functionality	Go to login page, click "Forgot Password?", enter registered email, click Submit	Success message displayed or reset email sent (currently expected to show message).	Nothing happens on submit. Page refreshes with no message or email sent.	Fail

Figure 35: Test Cases of GharBhada

## 8. Wireframes


The wireframes for my system is given below:

### Landing Page:



## Login and Signup Page:

Wireframe - Sign Up



Log InSign Up

Create Your Account

Full Name

Email Address

Phone Number

Permanent Address

Date of Birth

Citizenship Number

Email Address

--Select Role--

v

Password


Confirm Password

☐ I agree to the [Terms and Privacy Policy](#)

Create Account

Already have an account? [Log in here](#)

Wireframe - Log In



Log InSign Up

Welcome Back

Log in to manage your rentals

Email or Phone Number

Password

Show

☐ Remember me

Forget password?

Log In

or

Don't have an account? [Sign up free](#)





## Tenant Dashboard:

Wireframe - Tenant Dashboard

GharBhada

Tenant Portal


Search properties, tenants, bookings...



K

Krishma Shrestha

Tenant Account



Overview

Browse Properties

Whishlist

My Applications

Payments

Agreements

Documents

Complaints

Profile

Dashboard Overview

Welcome back! Find your perfect home with ease.

Browse Properties

Available Properties

0

Whishlist Items

Rs.0

Applications

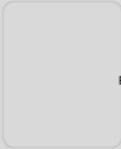
0

Total Payments

Rs.0

Featured Properties

View All




Baneshwor, Kathmandu, Nepal

Beautiful 2BHK Apartment in Baneshwor

Rs.25,000.00/mo

Recent Applications

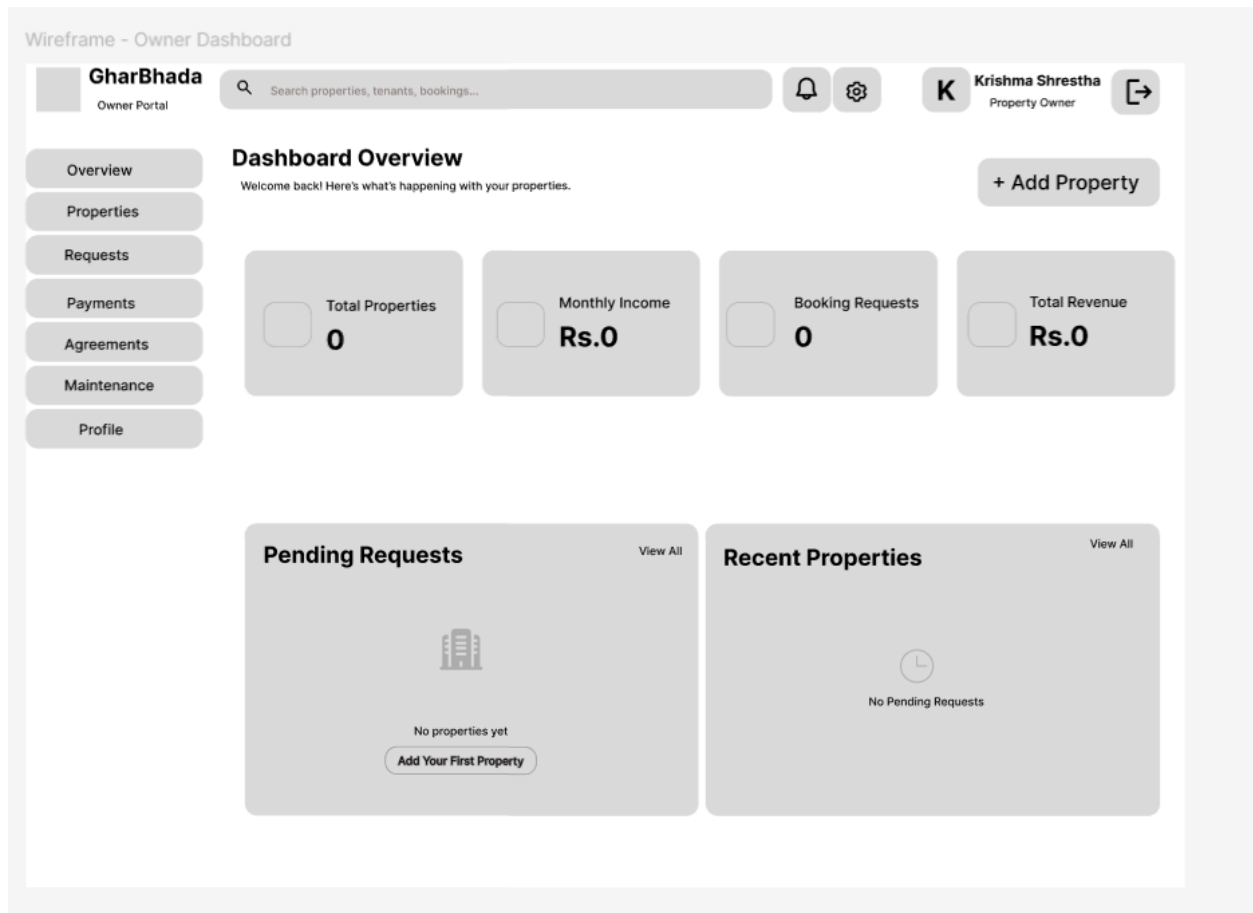
View All



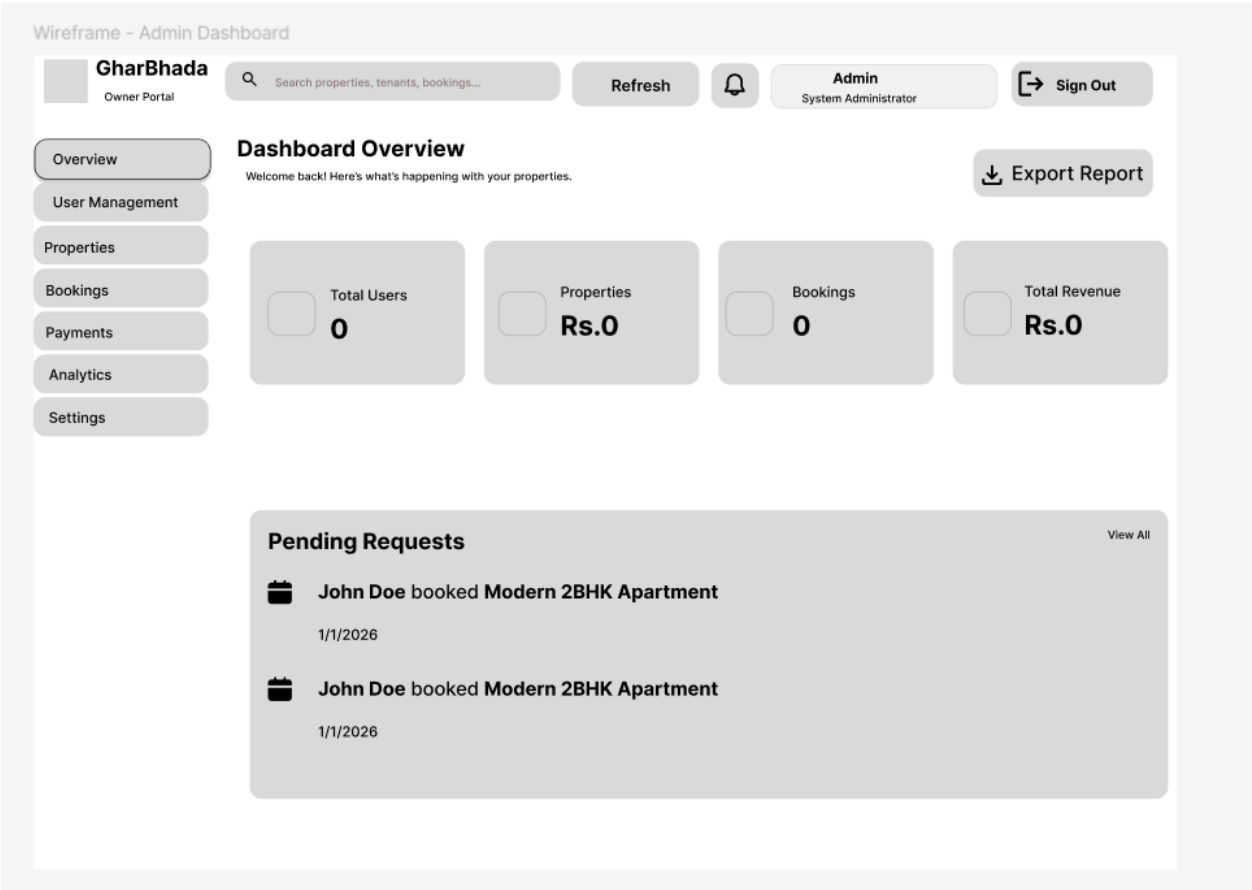
No Applications yet

62

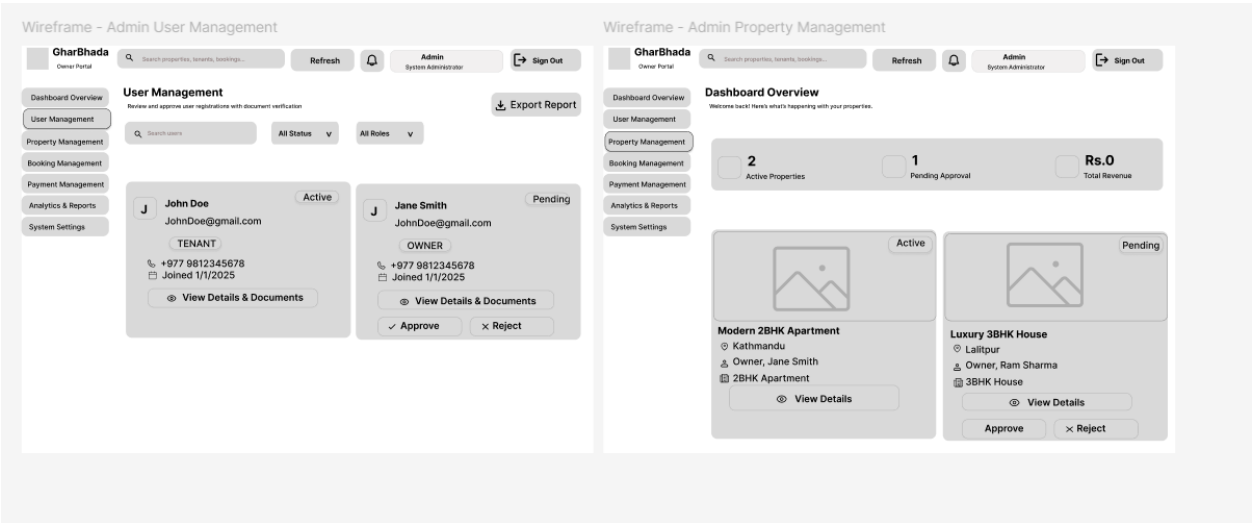
## Owner Dashboard:



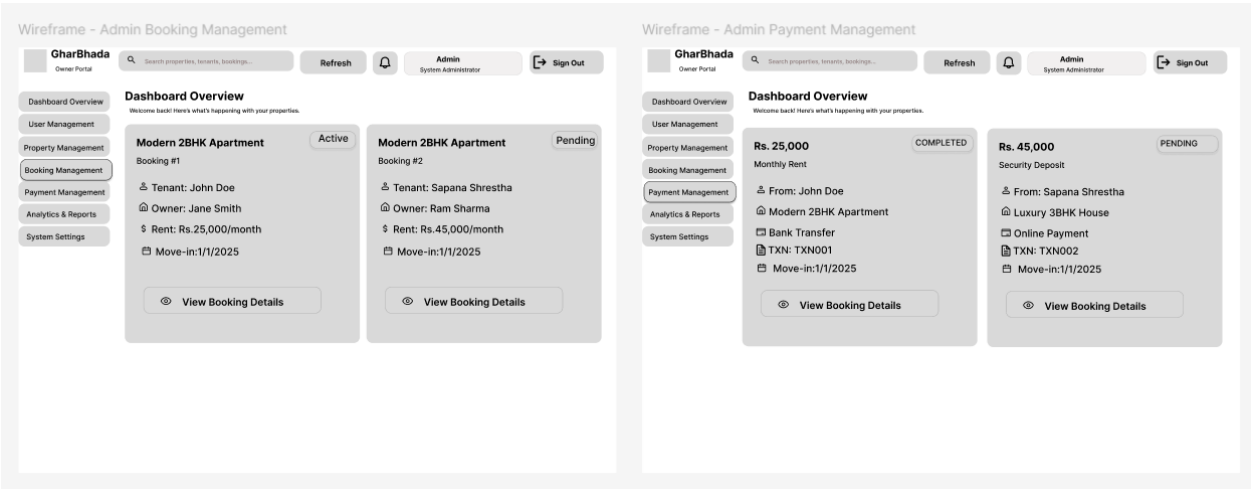
Admin Dashboard:



Admin User Management and Property Management:



# Admin Booking and Payment Management:



# Admin Analytics & Reports and System Settings:

