FUNCTIONAL REQUIREMENTS FOR RENTHUB PROPERTY MANAGEMENT APPLICATION

3.1 FUNCTIONAL REQUIREMENTS (FEATURES, INPUTS, OUTPUTS & PROCESSING)

User Authentication & Authorization:

FEATURES

Users (property owners, property managers, tenants) should be able to create accounts and log in securely.

Different user roles (e.g., owner, manager, tenant) should be defined with appropriate permissions.

INPUTS:

User-provided login credentials, including username and password.

User roles (property owner, property manager, tenant).

Authorization rules and permissions associated with each role.

Security tokens (e.g., API tokens for third-party integrations).

PROCESSING:

User Registration:

Users can create an account by providing a valid email address and a strong password.

The system verifies the email address's validity and checks for uniqueness.

User profiles, including roles and permissions, are created upon successful registration.

User Login:

Users must enter their registered email and password to log in.

The system validates the credentials against stored user data.

Successful login results in the generation of a session or authentication token.

User Role and Permissions:

The system assigns roles to users (property owner, property manager, tenant) during registration.

Different roles have specific permissions and access rights within the app.

Permissions include the ability to create property listings, manage tenants, view financial reports, and perform other related actions.

Administrators have the highest level of access and can manage users' roles and permissions.

Authorization:

Authorization rules determine whether a user is allowed to perform specific actions or access particular app features.

The system checks a user's role and permissions before allowing actions such as creating, editing, or deleting property listings, processing lease agreements, or viewing financial reports.

Unauthorized users are denied access and receive appropriate error messages or notifications.

OUTPUTS:

Successful user registration results in a registered user account with the assigned role and permissions.

Successful user login generates a session or authentication token for secure access to the app.

User roles and associated permissions are stored and updated in the user database.

Authorized users can perform the expected actions or access app features based on their roles.

Unauthorized users are denied access and receive clear error messages or notifications explaining the lack of access.

Property Listing and Management:

FEATURES

Property owners and managers should be able to create and manage property listings, including property details, photos, pricing, and availability.

Properties should be categorized by type, location, and amenities.

INPUTS:

Property details, including title, description, address, property type (e.g., apartment, house, condo), and size.

Property photos or media files for visual representation.

Rental pricing information, including rent amount, security deposit, and payment schedule.

Availability and scheduling data, specifying available move-in and move-out dates.

Location information, including coordinates or address.

PROCESSING:

Property Listing Creation:

Users (property owners and managers) can initiate the creation of a new property listing.

The system validates and stores property details, photos, and pricing information in the database.

A unique identifier (e.g., listing ID) is generated for each property listing.

Availability, scheduling data, and location information are calculated and associated with the property.

Property Listing Modification:

Users can update or modify the details, photos, pricing, availability, or location of existing property listings.

Changes are validated and saved, ensuring the accuracy of property information.

Notifications can be sent to property managers or administrators for approval of changes.

Property Listing Approval:

Property managers or administrators review new property listings and modifications.

They have the authority to approve or reject these listings based on predefined criteria and standards.

Approved listings become visible to potential tenants.

Property Search and Viewing:

Tenants can search for available rental properties based on specific criteria (e.g., location, rent price, property type).

Property listings matching search criteria are displayed for tenants to view.

Detailed property information, photos, and pricing are accessible to tenants.

OUTPUTS:

Successful property listing creation results in a published property listing with a unique listing ID.

Property details, including title, description, address, and property type.

Property photos and media are made accessible to users.

Pricing, availability, and scheduling information is presented to tenants.

Property managers or administrators receive notifications for new listings or modifications.

Approved property listings become visible in search results for potential tenants.

Tenant Management:

FEATURES

Tenants should have the ability to search for available rental properties based on their preferences, such as location, rent price, and property type.

Tenants should be able to submit applications for rental properties, including relevant information, background checks, and references.

INPUTS:

Tenant application information, including personal details, rental history, employment information, and references.

Property search criteria, such as location, rent budget, and property type.

Lease agreement data, including terms, rent amount, and move-in date.

Maintenance requests submitted by tenants.

PROCESSING:

Tenant Application Submission:

Tenants can submit rental applications, providing personal and contact information, rental history, employment details, and references.

The system validates the completeness and accuracy of application data.

Property managers or administrators are notified of new tenant applications.

Tenant Screening:

Property managers or administrators can review tenant applications, checking references and conducting background checks.

They can assess the eligibility of tenants based on predefined criteria.

Approved tenants are notified, while rejected tenants receive notifications with explanations.

Property Search and Application:

Tenants can search for available rental properties using specific criteria, including location, rent price, and property type.

They can view property details, photos, and pricing before deciding to submit an application.

Application submission includes specifying the desired move-in date and any additional requirements or notes.

Lease Agreement Management:

Property managers or administrators can create lease agreements for approved tenants.

Lease details, including terms, rent amount, and move-in date, are documented.

Lease documents are electronically signed by both parties and securely stored in the system.

Maintenance Requests:

Tenants can submit maintenance requests through the app, describing the issue and its urgency.

Property managers or administrators can review and assign maintenance tasks.

Communication and updates related to maintenance are tracked within the system.

OUTPUTS:

Successful tenant applications result in tenant profiles with personal and rental history details.

Property managers or administrators receive notifications for new tenant applications.

Approved tenants are notified of their acceptance.

Rejected tenants receive notifications with explanations for the rejection.

Tenants can search for available properties and view details.

Lease agreements are created with details and electronically signed by both parties.

Maintenance requests are tracked, and tenants receive updates on task assignments and completion.

Lease Management:

FEATURES

Property owners and managers should be able to create and manage lease agreements, customize terms, and generate lease documents.

Electronic signatures and document storage should be supported.

INPUTS:

Tenant and property information, including tenant details, property details, and rental terms.

Lease customization options, including rent amount, lease duration, security deposit, and special clauses.

Digital signatures from both parties.

PROCESSING:

Lease Agreement Creation:

Property owners and managers can create new lease agreements by selecting a tenant and a property.

They input lease details, including rent amount, lease duration, security deposit, and any special clauses.

The system validates the accuracy and completeness of the lease data.

Customization and Editing:

Property owners and managers can customize lease agreements to meet specific requirements or add custom clauses.

They can edit lease terms, update rent amounts, and modify other details as needed.

Changes to the lease are tracked and recorded.

Electronic Signatures:

The lease agreement is electronically signed by both the tenant and the property owner or manager.

The system securely stores the signed lease agreement for reference.

Notifications are sent to both parties upon successful signing.

Lease Renewal:

Property owners and managers can initiate the renewal of existing lease agreements.

They can update terms, negotiate rent, and obtain electronic signatures for the renewed lease.

Renewed lease agreements are securely stored alongside the original agreements.

Lease Termination:

Property owners and managers can initiate the termination of lease agreements.

They can specify the termination date and reasons for termination.

Notifications are sent to both parties regarding lease termination.

OUTPUTS:

Successfully created lease agreements with details, rent amounts, and clauses.

Customized lease agreements tailored to specific requirements.

Electronically signed lease agreements with digital signatures from both parties.

Lease renewal agreements and notifications for renewed terms.

Lease termination notices with reasons for termination.

Rent Collection and Payment:

FEATURES

Property owners should be able to set up and manage rent collection processes, including payment methods and schedules.

Tenants should receive payment reminders and notifications.

INPUTS:

Tenant and property information, including tenant details, property details, and lease terms.

Rent payment amount, payment method (e.g., bank transfer, credit card, ACH), and payment schedule.

Payment reminders and notifications.

PROCESSING:

Rent Payment Setup:

Property owners and managers can set up rent collection for each property, specifying the rent amount, payment frequency, and due dates.

They can choose from various payment methods to accept rent, such as bank transfers, credit card payments, or ACH (Automated Clearing House) transfers.

Payment Reminders:

The system generates automatic rent payment reminders for tenants, notifying them of upcoming payments.

Reminders can be sent via email, in-app notifications, or SMS.

Rent Payment Submission:

Tenants can submit rent payments through the app, specifying the payment amount and method.

The system securely processes payments, verifying payment details and sending confirmations.

Payment Tracking and Record Keeping:

The app tracks rent payments for each tenant and property, maintaining a payment history.

Property owners and managers can access payment records and view payment statuses.

Late Payment Handling:

The system can automatically calculate and apply late fees for overdue rent payments.

Late payment notifications are sent to tenants, reminding them to settle outstanding amounts.

OUTPUTS:

Configured rent collection setups for each property, specifying rent amount, frequency, and payment methods.

Automated rent payment reminders and notifications to tenants.

Rent payments submitted by tenants, along with payment confirmations.

Payment tracking and records for property owners and managers.

Notifications to tenants regarding late payments and associated late fees.

Maintenance Requests:

FEATURES

Tenants should be able to submit maintenance requests through the app, describing issues and urgency.

Property managers should have tools to assign, track, and manage maintenance tasks, communicate with tenants, and provide updates.

INPUTS:

Maintenance request details, including a description of the issue, its urgency, and any supporting media (photos or videos).

Property and tenant information for context and assignment.

Communication between tenants and property managers or owners regarding maintenance requests.

PROCESSING:

Maintenance Request Submission:

Tenants can submit maintenance requests through the app, providing a detailed description of the issue, its urgency, and any relevant media (e.g., photos or videos).

The system validates the request and assigns a unique request ID.

Notifications are sent to property managers or owners regarding the new request.

Assignment and Tracking:

Property managers or owners can review maintenance requests, assign them to maintenance personnel, and specify due dates for resolution.

Assigned personnel are notified, and the request status is updated.

Property managers or owners can track the progress of assigned requests.

Communication and Updates:

Tenants can communicate with property managers or owners regarding the status of maintenance requests.

Property managers or owners can provide updates on the resolution process.

All communication related to the maintenance request is documented and available for reference.

Resolution and Verification:

Maintenance personnel complete the necessary repairs or maintenance tasks.

Tenants and property managers or owners can verify the completion of the request.

Maintenance records are updated with details of the resolution.

OUTPUTS:

Successfully submitted maintenance requests with unique request IDs.

Maintenance requests assigned to personnel with due dates.

Communication records between tenants, property managers, and maintenance personnel.

Updates on the status and resolution of maintenance requests.

Maintenance records containing details of completed tasks and verifications.

Reporting and Analytics:

FEATURES

Property owners and managers should have access to reporting and analytics tools to monitor financial performance, occupancy rates, and property management insights.

Reports should cover rental income, expenses, and other relevant metrics.

INPUTS:

Property and lease data, including rental income, expenses, occupancy rates, and lease agreements.

User-defined reporting criteria, including date ranges, property filters, and specific metrics.

Data visualization preferences and report formatting options.

PROCESSING:

Report Generation:

Property owners and managers can generate various reports to gain insights into their property portfolios.

Reports can include data on rental income, expenses, occupancy rates, lease agreements, and other relevant metrics.

Users specify reporting criteria, such as date ranges, property filters, and desired metrics.

Data Analysis and Visualization:

The system processes data to analyze and calculate key performance indicators (KPIs), such as return on investment (ROI), occupancy rates, and revenue forecasts.

Data is presented in visually appealing charts, graphs, and tables for easier comprehension.

Users can interact with data visualizations, such as drilling down for more details.

Scheduled Reports:

Property owners and managers can schedule regular automated reports, specifying the recipients and delivery frequency (e.g., daily, weekly, monthly).

Scheduled reports are generated and sent to designated recipients, ensuring they receive timely updates.

Custom Report Templates:

Users can create custom report templates, allowing them to save preferred report settings and formatting options for reuse.

Custom templates streamline the report generation process.

Data Export and Download:

Users can export and download reports in common formats, such as PDF, Excel, or CSV.

Exported data can be used for further analysis or external reporting.

OUTPUTS:

Generated reports with visualizations, KPIs, and key metrics.

Scheduled reports sent to designated recipients on a regular basis.

Custom report templates for users' convenience and time-saving.

Exported reports in user-selected formats for external use or sharing.

Communication and Messaging:

Users should be able to communicate and exchange messages within the platform.

Automated notifications should be used for lease renewals, rent payments, and maintenance updates.

Document Management:

The app should provide a secure repository for storing and managing property-related documents, such as lease contracts, invoices, and maintenance records.

Search and Filter Functionality:

Users should have search and filter options to find properties based on location, rent price, property type, and amenities.

User Support and Help Center:

The app should offer user support resources, including FAQs, chat support, and help materials.

Mobile Access:

Users should have access to the app through mobile-responsive design for convenient use on smartphones and tablets.

Security and Compliance:

The app should implement strong security measures, including data encryption and secure authentication.

It should comply with relevant data protection regulations and property management laws.

Backup and Recovery:

Regular data backups and a disaster recovery plan should be in place to ensure data integrity and availability.

3.2 USE CASES

User Registration and Authentication:

Primary Actor: New User

Scenario: A new user wants to create an account on the app. They provide their email and create a password. The system validates the registration, sends a verification link, and the user completes the registration process.

Property Listing Creation:

Primary Actor: Property Owner or Manager

Scenario: A property owner or manager wants to add a new property listing. They provide property details, upload photos, set rental pricing, and specify availability. The system validates the listing and makes it visible to potential tenants.

Tenant Property Search and Application:

Primary Actor: Tenant

Scenario: A tenant is looking for a rental property. They search based on location, rent price, and property type. The app displays matching listings. The tenant selects a property, reviews details, and submits an application, providing their information, references, and background checks.

Lease Agreement Creation and Signing:

Primary Actor: Property Owner or Manager

Scenario: A property owner or manager wants to create a lease agreement for an approved tenant. They customize the agreement, specifying terms, rent amount, and move-in date. The tenant receives the lease for electronic signing. Once both parties sign, the system stores the signed agreement.

Rent Collection and Payment:

Primary Actor: Property Owner or Manager

Scenario: It's time to collect rent from tenants. The property owner or manager sets up rent collection with the specified rent amount, due date, and payment method. Tenants receive payment reminders and submit payments through the app.

Maintenance Request Submission:

Primary Actor: Tenant

Scenario: A tenant encounters a maintenance issue in their rental property. They submit a maintenance request, describing the problem and its urgency. Property managers or owners receive the request, assign it to maintenance personnel, and track the resolution process.

Reporting and Analytics:

Primary Actor: Property Owner or Manager

Scenario: A property owner or manager wants to assess the performance of their rental properties. They generate reports, select metrics, and view charts, graphs, and tables showing rental income, expenses, occupancy rates, and other relevant data. Custom report templates and scheduled reports enhance their analysis.

User Communication and Messaging:

Primary Actor: All Users

Scenario: Users need to communicate with each other. Property owners, managers, and tenants can exchange messages within the app, discuss lease renewals, maintenance updates, or property inquiries. Automated notifications keep users informed about critical events.

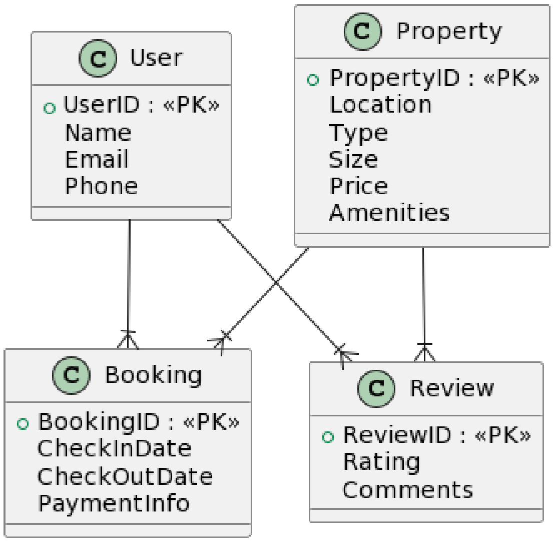
User Account Management:

Primary Actor: User

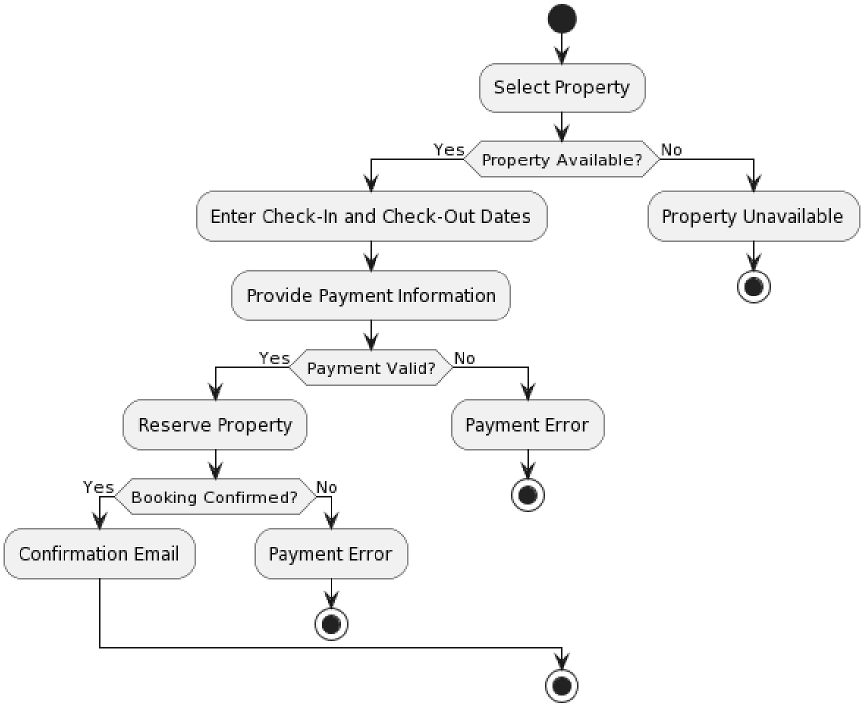
Scenario: Users need to manage their accounts. They can update personal information, change passwords, and review their activity and communication history. Administrators can manage user roles and permissions.

3.3 DATA MODELLING AND ANALYSIS

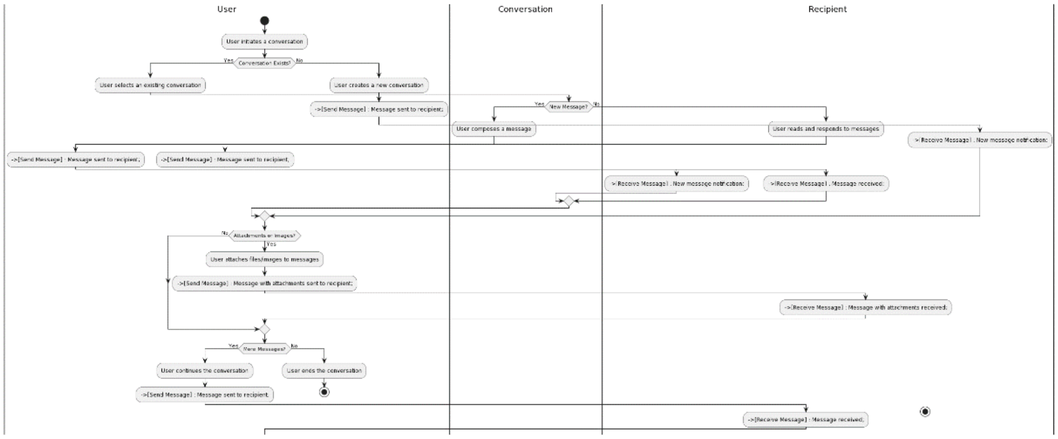
Normalized Data Model Diagram:



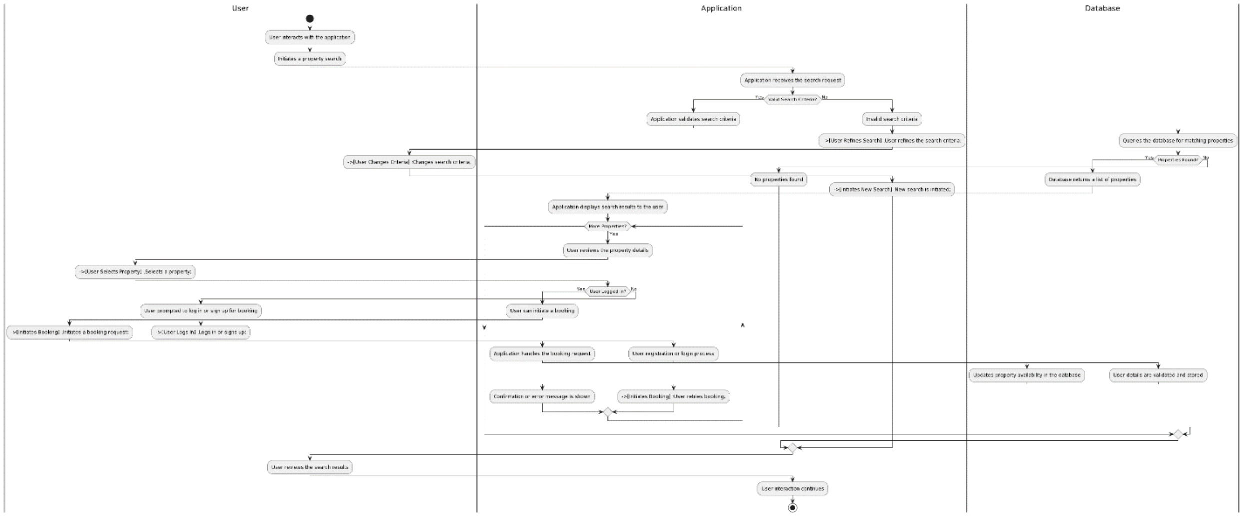
Activity Diagrams:



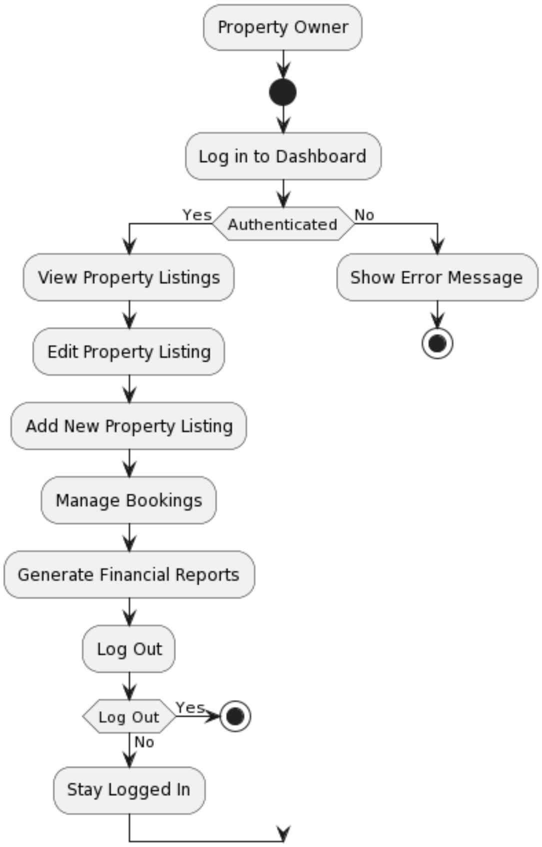
Activity Diagram For Booking & Reservation


User
Conversation
Recipient
User initiates a conversation
Yes
Conversation Exists?
No
User selects an existing conversation
User creates a new conversation
->[Send Message] : Message sent to recipient;
Yes
New Message?
No
User composes a message
User reads and responds to messages
->[Receive Message] : New message notification;
->[Send Message] : Message sent to recipient;
->[Send Message] : Message sent to recipient;
->[Receive Message] : New message notification;
->[Receive Message] : Message received;
No
Attachments or Images?
Yes
User attaches files/images to messages
->[Send Message] : Message with attachments sent to recipient;
->[Receive Message] : Message with attachments received;
Yes
More Messages?
No
User continues the conversation
User ends the conversation
->[Send Message] : Message sent to recipient;
->[Receive Message] : Message received;

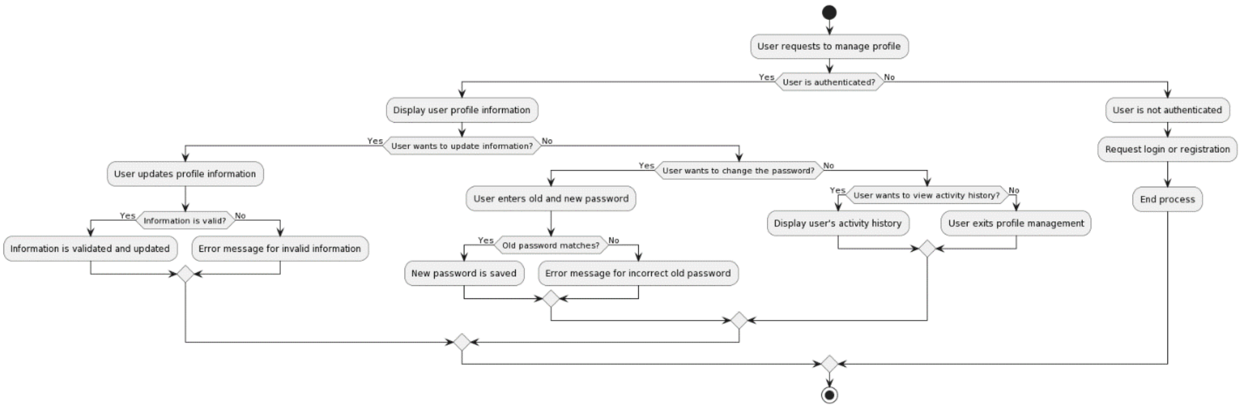
Activity Diagram For Messaging & Communication


User
Application
Database
User interacts with the application
Initiates a property search
Application receives the search request
Yes
Valid Search Criteria?
No
Application validates search criteria
Invalid search criteria
->[User Refines Search] ;User refines the search criteria;
Queries the database for matching properties
Yes
Properties Found?
No
->[User Changes Criteria] ;Changes search criteria;
No properties found
Database returns a list of properties
->[Initiates New Search] ;New search is initiated;
Application displays search results to the user
More Properties?
Yes
User reviews the property details
->[User Selects Property] ;Selects a property;
Yes
User Logged In?
No
User prompted to log in or sign up for booking
User can initiate a booking
->[Initiates Booking] ;Initiates a booking request;
->[User Logs In] ;Logs in or signs up;
Application handles the booking request
User registration or login process
Updates property availability in the database
User details are validated and stored
Confirmation or error message is shown
->[Initiates Booking] ;User retries booking;
User reviews the search results
User interaction continues

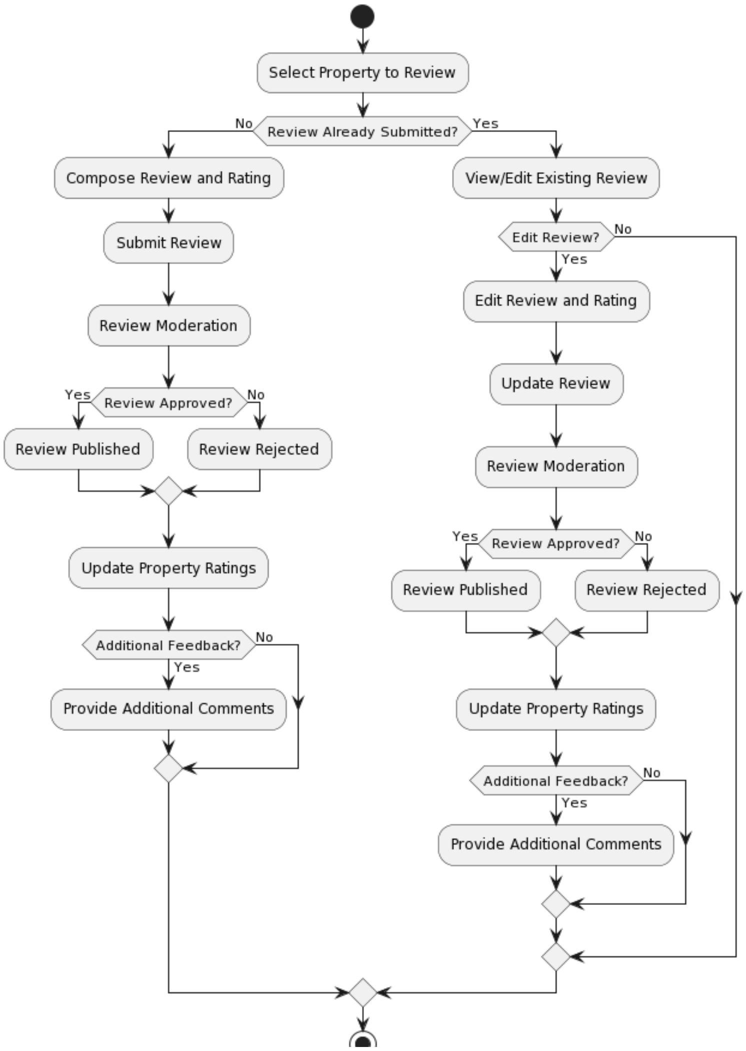
Activity Diagram For Property Listing & Search



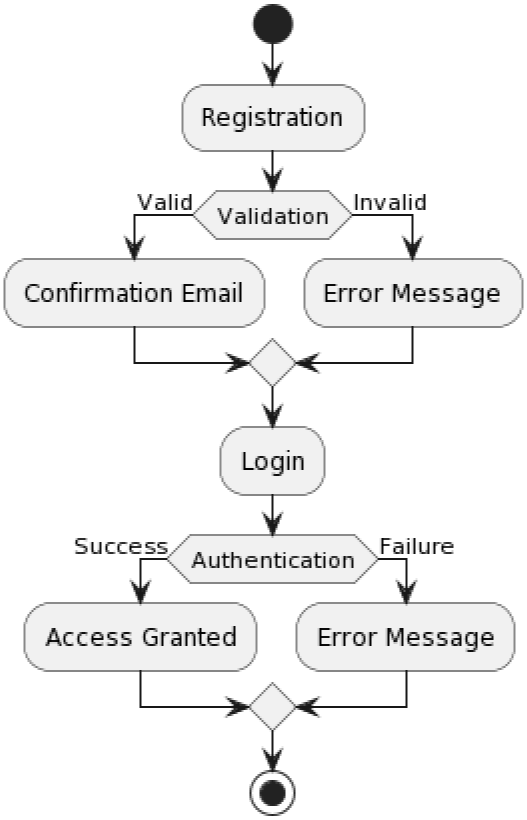
Activity Diagram For Property Owner Dashboard


User requests to manage profile
Yes
User is authenticated?
No
Display user profile information
User is not authenticated
Yes
User wants to update information?
No
Request login or registration
User updates profile information
Yes
User wants to change the password?
No
User enters old and new password
Yes
User wants to view activity history?
No
End process
Yes
Information is valid?
No
Information is validated and updated
Error message for invalid information
Display user's activity history
User exits profile management
Yes
Old password matches?
No
New password is saved
Error message for incorrect old password

Activity Diagram For User Profile Management

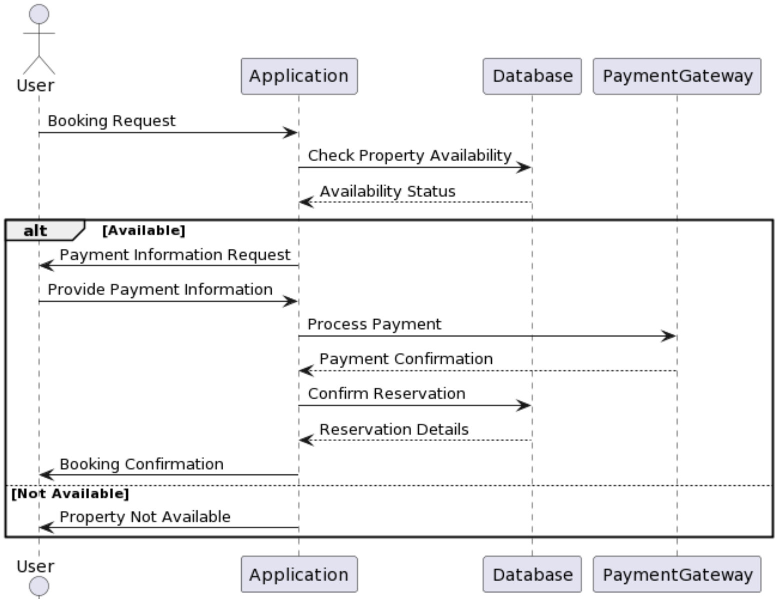


Activity Diagram For User Review & Rating

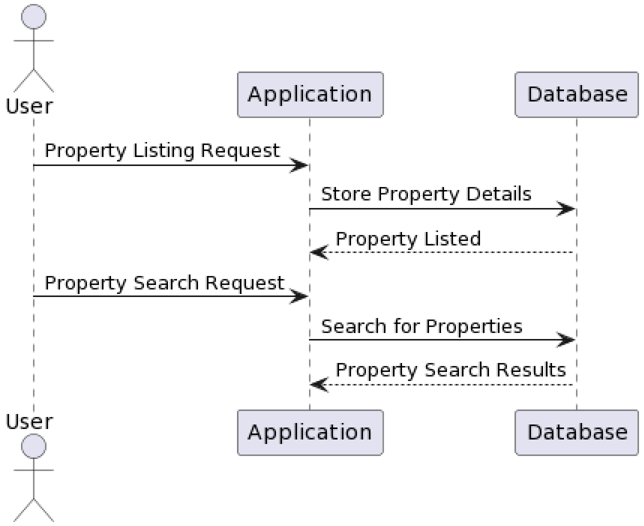


Activity Diagram For Authentication & Authorization

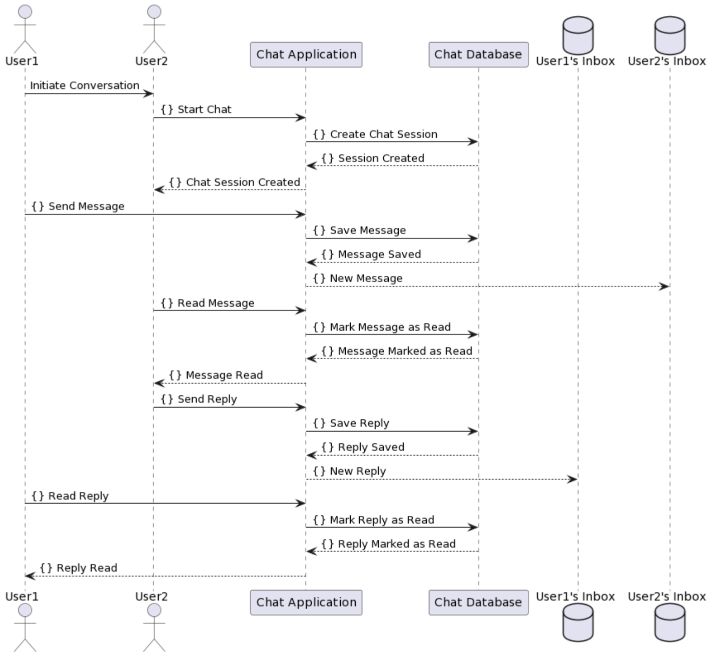
Sequence Diagrams:



Sequence Diagram For Booking & Reservation

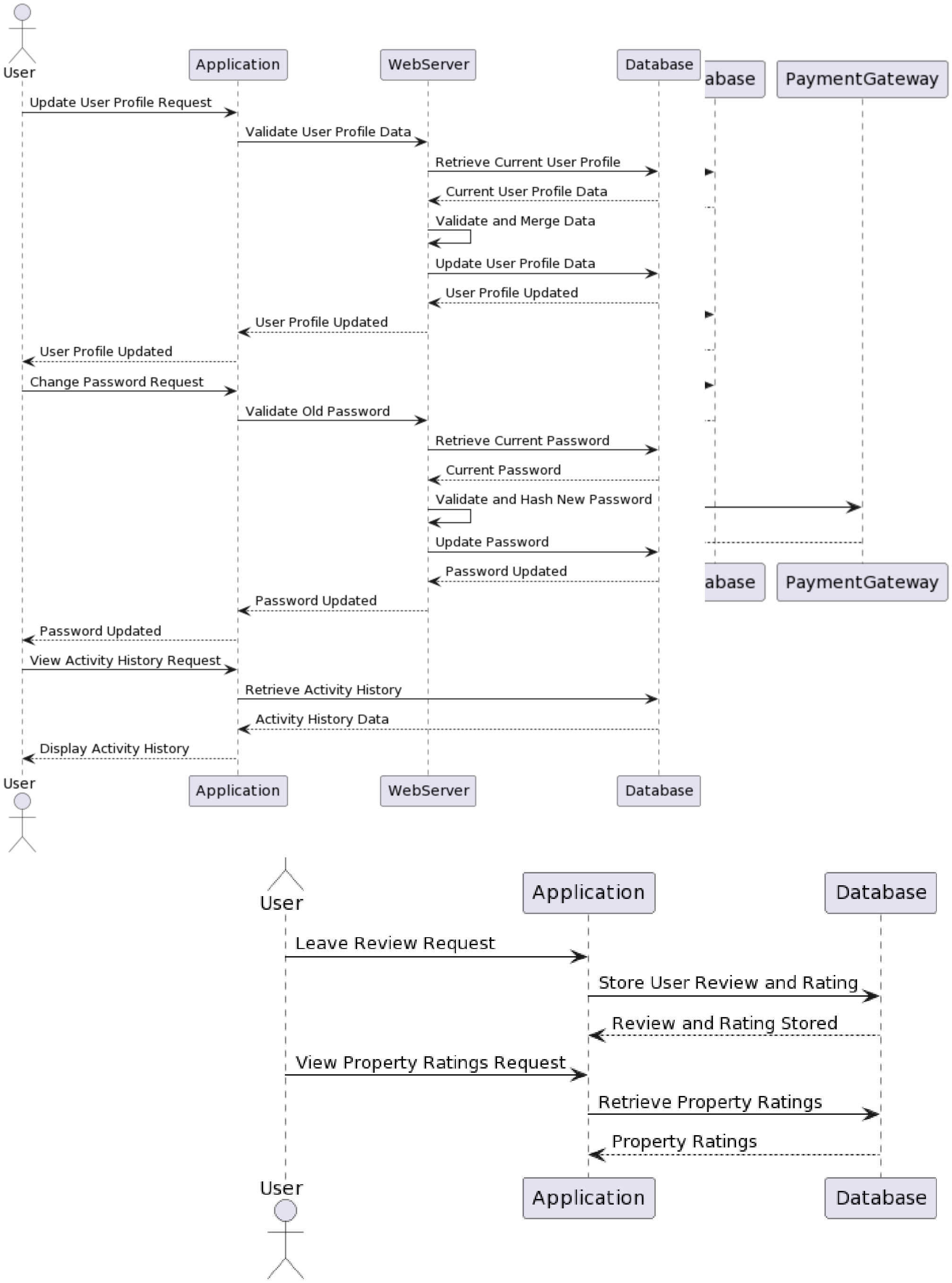


Sequence Diagram For Property Listing & Search



Sequence Diagram For Message & Communication

Sequence Diagram For Owner Dashboard



Sequence Diagram For Login & Registration

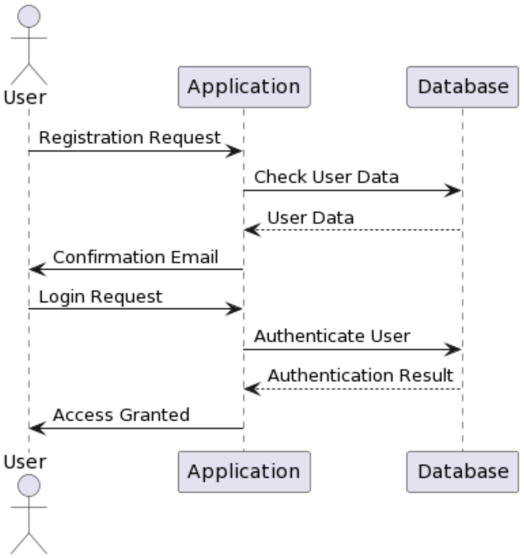
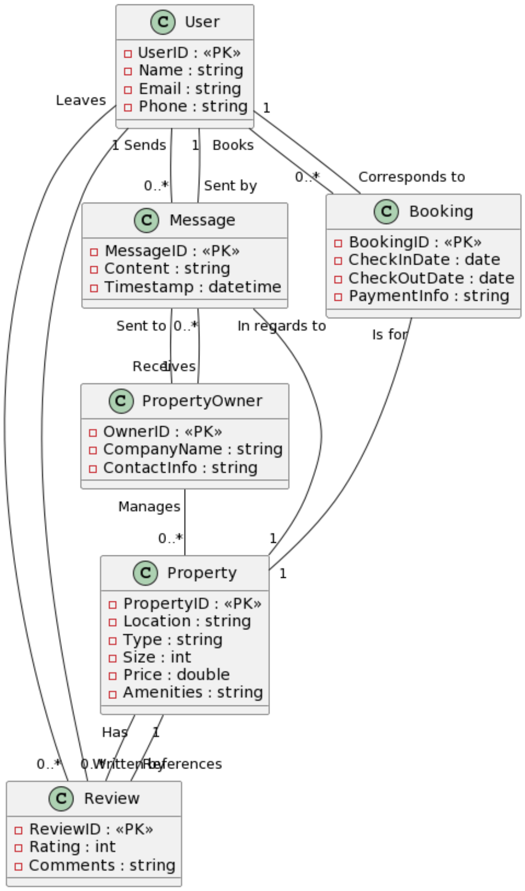
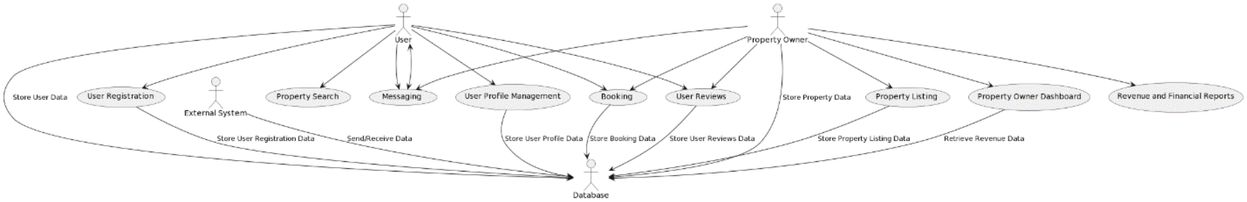


Figure 1Sequence Diagram For User Profile Management

UML Class Diagram:



Data Flow Diagram:


User
Property Owner
Store User Data
User Registration
External System
Property Search
Messaging
User Profile Management
Booking
User Reviews
Store Property Data
Property Listing
Property Owner Dashboard
Revenue and Financial Reports
Store User Registration Data
Send/Receive Data
Store User Profile Data
Store Booking Data
Store User Reviews Data
Store Property Listing Data
Retrieve Revenue Data
Database