



Software Design Specification

for

Housepooling System

Version <2.0>

Group No. : 6

Poh Ern Qi	1211101398
Tan Jia Jin	1211101800
Wan Izzul Wafiq bin Wan Noor Asmawi	1201100927
Sheikh Adam Zachary bin Sheikh Naziruddin	1201100927

Date: 14/1/2024

Contents

Contents.....	2
Revisions.....	3
1 System Overview.....	4
1.1 Description.....	4
1.2 Actors.....	4
1.3 Assumptions and Dependencies.....	5
2 Use Cases.....	7
2.1 Use Case Diagram.....	7
2.2 Owner.....	8
2.3 Tenant.....	25
2.4 Admin.....	38
2.5 Searcher.....	58
3 Data Design.....	67
3.1 Design Class Diagram.....	67
3.2 Data Dictionary.....	70
3.4 Data Structures.....	78
4 Architecture Design.....	83
4.1 Software Architecture.....	83
4.2 State Transition Diagram.....	87
5 Interface Design.....	88
5.1 Index Screen / Menu Page (Main Screen).....	88
5.2 Account Management Subsystem.....	89
5.3 Manage Property Listing Subsystem.....	93
5.4 Booking Subsystem.....	99
5.5 Manage Agreement Subsystem.....	101
5.6 Review and Rating Subsystem.....	105
6 Component Design.....	107
6.1 Main Components.....	107
7 Deployment Design.....	144
7.1 Deployment Diagram.....	144
8 Updated Requirements.....	145

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Version 1	Poh Ern Qi Sheikh Adam Zachary bin Sheikh Naziruddin Tan Jia Jin Wan Izzul Wafiq bin Wan Noor Asmawi	Integrated use cases done in Project 1	30/12/2023
Version 2	Poh Ern Qi Sheikh Adam Zachary bin Sheikh Naziruddin Tan Jia Jin Wan Izzul Wafiq bin Wan Noor Asmawi	draw diagrams	10/1/2024
Version 3	Poh Ern Qi Sheikh Adam Zachary bin Sheikh Naziruddin Tan Jia Jin Wan Izzul Wafiq bin Wan Noor Asmawi	finalize project	14/1/2024

1 System Overview

1.1 Description

The roles of housepooling system include owner, tenant, admin and searcher. All actors have functionalities to register for an account, login to the system, manage their profile, and view property details. The owners are able to view the property details in the manage property section, which also provides features for owner to add property, update property and delete property. The newly added and updated property will then be sent to admin for approval. Owner can also manage lease. For existing lease, owners can choose either to renew or terminate the lease. Owners can also create new lease for new tenants.

As for the searcher, they are able to scroll and view the property list that is offered by visiting the website. They may view the details of the property that they are interested in. There are also options for them to book timing to view the property by selecting a slot that is available for the viewing. If they decide to officially become a tenant, they need to register for an account and can manage their profile there.

The Admin oversees and manages various crucial processes within the system. They serve as a key moderator, responsible for reviewing and approving property listings uploaded by the Owner. Additionally, the Admin manages reviews, ensuring compliance by reviewing and deleting Tenant reviews when necessary. They possess the capability to view booking schedules between Owners and Searchers. In cases where conflicts arise in booking requests or discrepancies occur, the Admin acts as a communicator, engaging with involved parties to find a resolution. Furthermore, the Admin plays a pivotal role in managing leases, being able to view lease details between Owners and Tenants. They also act as user support, having the authority to manage role change for tenants wanting to become an owner. As part of their responsibilities, the Admin handles financial oversight, ensuring accurate and complete management of financial transactions.

The tenants are basically the customers who rent the property. After the searcher finds a suitable property, they will then become a tenant that will rent the property. The tenant can renew or terminate their lease by communicating with the owner. Deposit payments will be made by the tenant when they have a new agreement . They can also make reviews of the property for other tenants or searchers to see whether the property is suitable or not. If the tenant decides to become an owner, they can submit their application to the admin, with the condition that they can only be accepted for role change if they do not have a late payment history or an active lease.

1.2 Actors

Actor	Use Cases
-------	-----------

Owner	-Register -Login -Manage property (remove, update, add, view property) -Manage profile -Add/ renew/ terminate lease
Tenant	-Register -Login -View property details -Payment -renew/ terminate lease -Review -Create/edit profile
Admin	-Register -Login -Manage profile -Manage property listing (approve, reject property uploaded by owner) -Manage reviews -Manage booking -Manage lease(view) -Manage user -Manage transaction (view)
Searcher	- Register - Login - Manage profile - View property details - Book timing to view property

1.3 Assumptions and Dependencies

Assumptions

- The lack of API for sending emails

Running and testing our system on localhost typically require an API or mail server to handle the email transmission, which might be complex to implement.

- Time constraint

Tight deadlines may affect the development process, leading to the need for prioritization of features and potentially impacting the quality of the final product.

Dependencies

- Integration with External Systems

The lack of payment gateway provider for its API and other materials to complete the payment process might affect its functionalities, as the system likely relies on external payment services for processing deposit payments.

2 Use Cases

2.1 Use Case Diagram

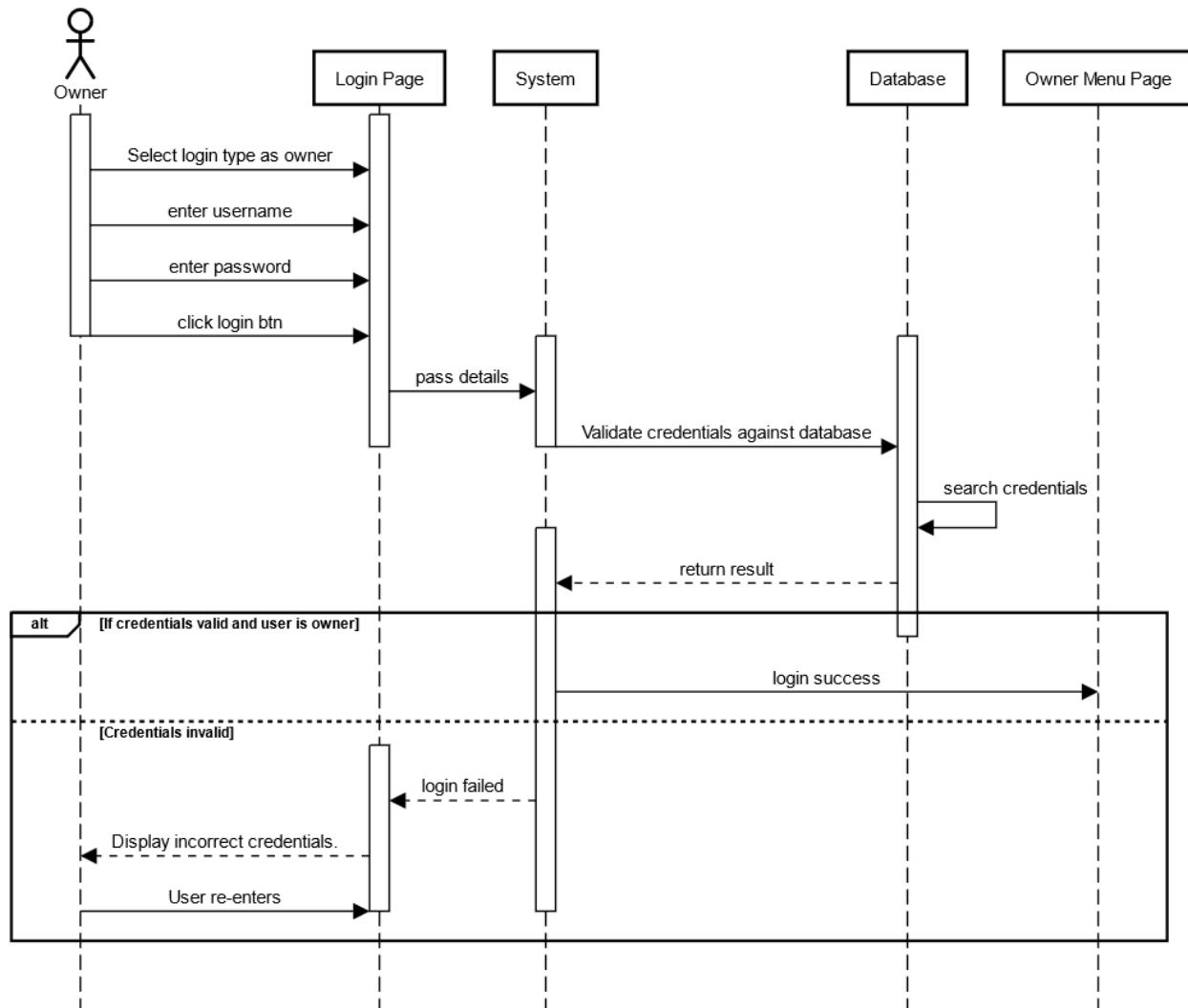


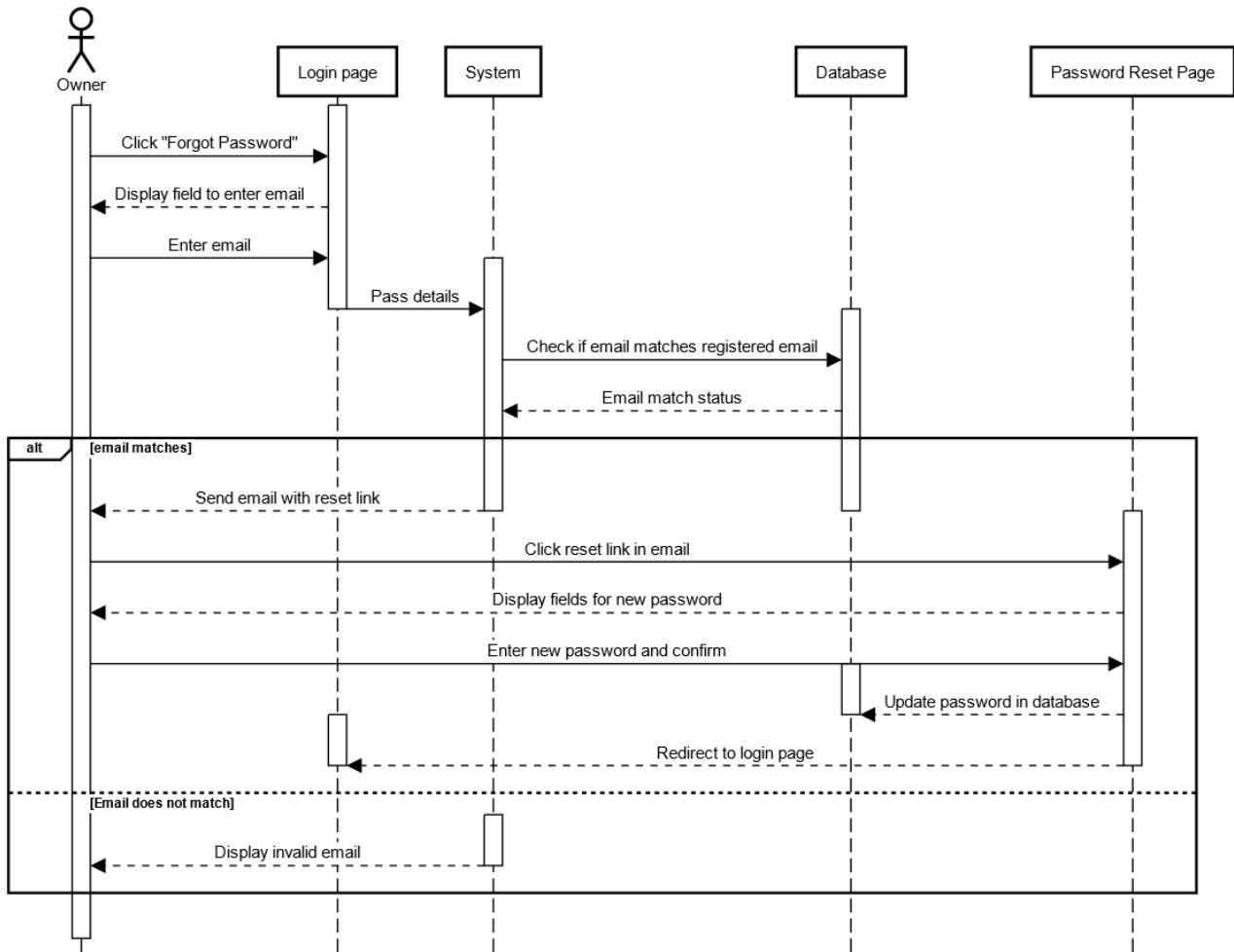
2.2 Owner

2.2.1 Login

Use Case Name:	Login
Description:	This use case outlines the process by which an owner logs into the application, gains access to their account, and navigates to personalized features. It also includes option for password resets.
Primary Actor:	Owner
Precondition	Owner has a registered account.
Postcondition	Owner is logged into the system and can access personalized features.
Main Success Scenario:	<ol style="list-style-type: none">1. The owner selects the login type as 'owner' on the Login Page.2. The owner enters their username into the appropriate field.3. The owner enters their password into the appropriate field.4. The owner clicks the login button to submit their credentials.5. The system passes the details to a backend service that validates credentials against a database.6. The database searches for the entered credentials.7. If the credentials are found and valid, and the user is confirmed as an 'owner,' the database returns a successful result to the system.8. The system then directs the owner to the Owner Menu Page, completing the login process successfully.
Alternative Scenario:	

	<ol style="list-style-type: none">1. If the credentials are invalid or that the user is not an 'owner', the login page displays a message indicating the credentials are incorrect.2. The owner is prompted to re-enter their credentials, returning to step 2 of the main scenario.3. If the owner selects the "forgot password" option, the system presents a field to enter their email.4. The owner enters their email and clicks the reset password button.5. The system checks if the email matches the registered email.6. If it matches, the system sends a password reset link to the user's email.7. The owner clicks on the link received in their email, which leads to a page to enter and confirm a new password.8. The owner enters the new password details and clicks submit.9. The owner is redirected to the login page to log in with the new password.
--	---



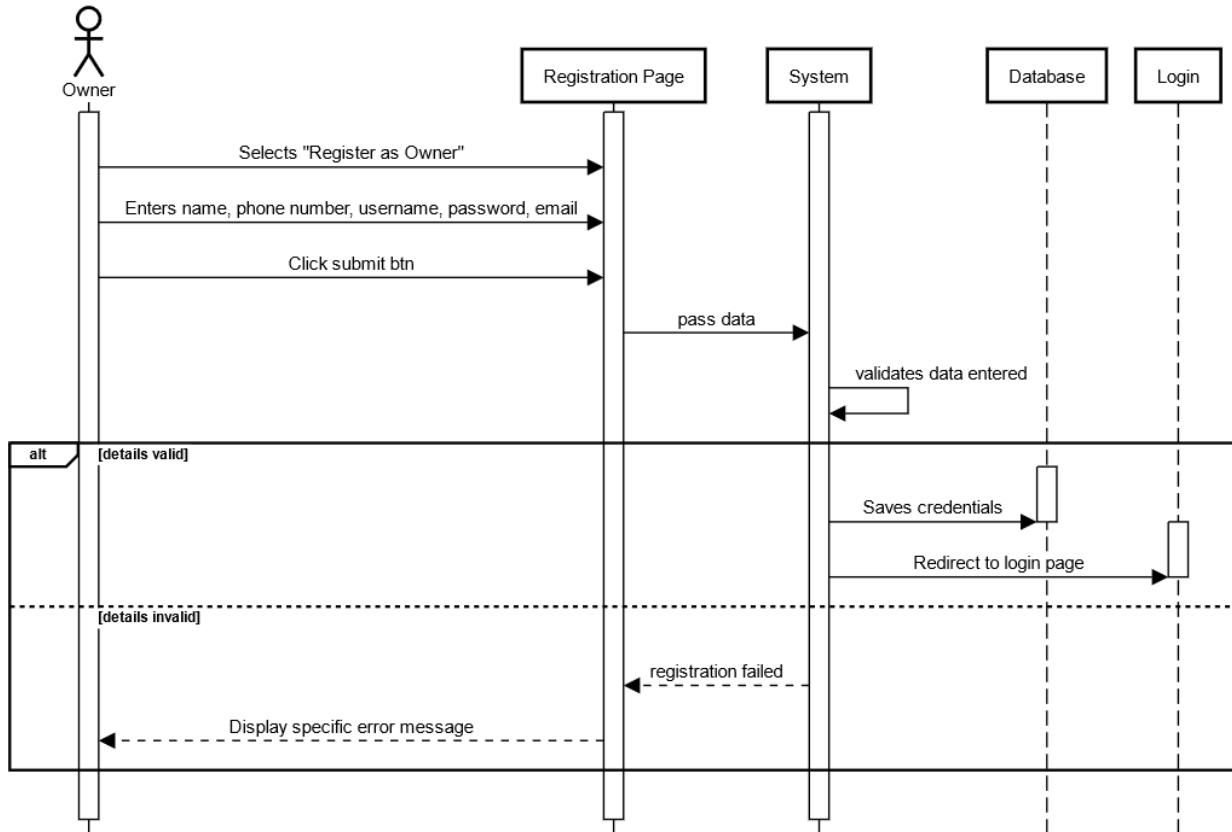


2.2.2 Register

<TO DO: Write the use case description and place the sequence diagram.>

Use Case Name:	Register
Description:	This use case outlines the process for registering a new owner account within the system.
Primary Actor:	Owner

Precondition	No existing account should exist for the prospective owner credentials.
Postcondition	The property owner has a registered account. The owner can log in with their registered username and password.
Main Success Scenario:	<ol style="list-style-type: none"> 1. User selects register as owner. 2. Owner fills in details including name, phone number, username, password, email, then clicks create account to submit the registration details. 3. System validates the credentials. If the information is complete with no errors, it proceeds to save the data to database. User is then redirect to login page. Else, if there are validation errors, the app displays the error messages respectively. 4. Once the information is validated, the app creates a new owner account with the provided details.
Alternative Scenario:	<ol style="list-style-type: none"> 1. If the owner tries to register with an email address that is already associated with an existing account, it will not proceed with registration and will display user already exists message and prompt the owner to either log in with the existing account or use a different email address. 2. If name or phone number contains special characters, it will display “special characters not allowed”. 3. If phone number contains alphabets, it will display “phone number cannot contain alphabets”. 4. If email address entered is invalid format, it will display “please enter a valid email”.

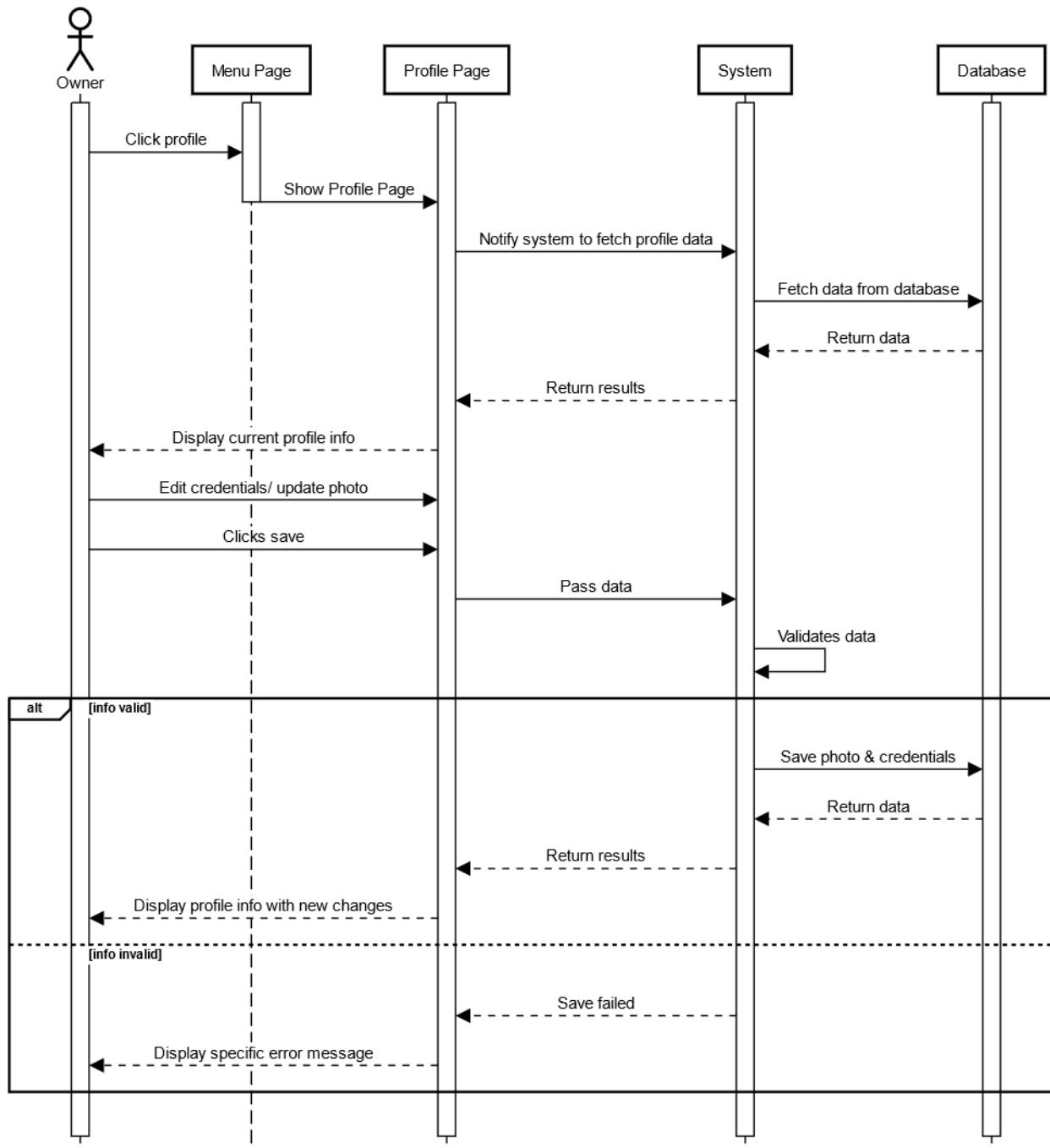


2.2.3 Manage Profile

<TO DO: Write the use case description and place the sequence diagram.>

Use Case Name:	Manage Profile
Description:	This use case describes the process by which an owner can update their profile information within the system. The owner has the option to update their profile photo, credentials, and other personal details. The system ensures the integrity of the updates through validation before saving any changes to the database.
Primary Actor:	Owner
Precondition	Owner is logged into the system.

Postcondition	<ul style="list-style-type: none">● The user's profile is updated in the system with new information, if any changes are made.● The system's database reflects the latest state of the user's profile data after successful updates.
Main Success Scenario:	<ol style="list-style-type: none">1. The owner clicks on the profile option in the Menu Page.2. The Profile Page is displayed to the owner.3. The system is notified to fetch the owner's profile data from the database.4. The system retrieves the data from the database.5. The Profile Page displays the current profile information to the owner.6. The owner edits their credentials and/or updates their photo on the Profile Page.7. The owner clicks the save button to submit the changes.8. The system passes the updated data to the database.9. The database validates the updated data.10. The database saves the updated photo and credentials.11. The database returns a confirmation that the data has been updated.12. The system displays the profile information with the new changes to the owner on the Profile Page.
Alternative Scenario:	If the information entered is invalid, the page displays a message indicating which credential is invalid and its specific errors.



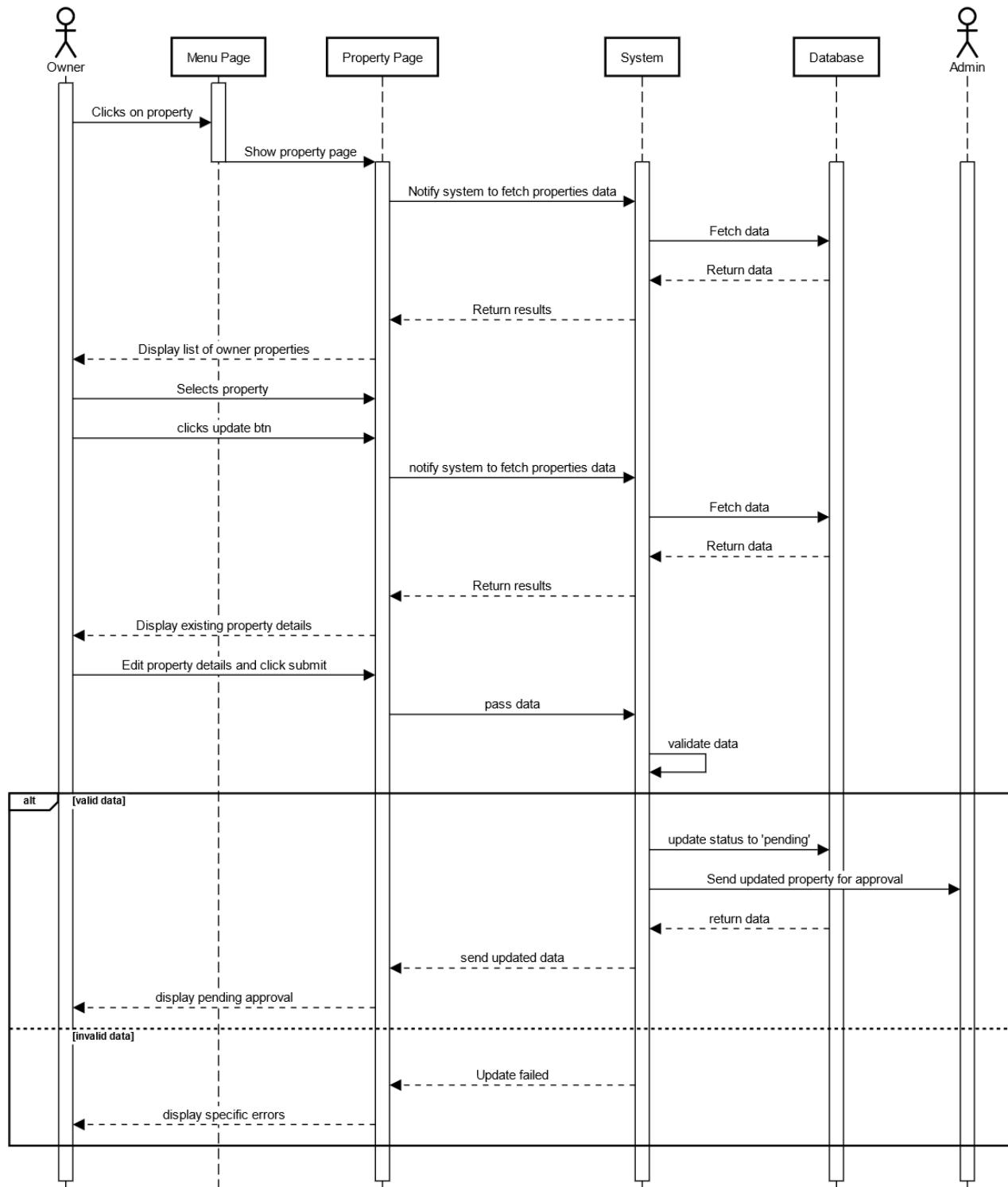
2.2.4 Manage Property

<TO DO: Write the use case description and place the sequence diagram.>

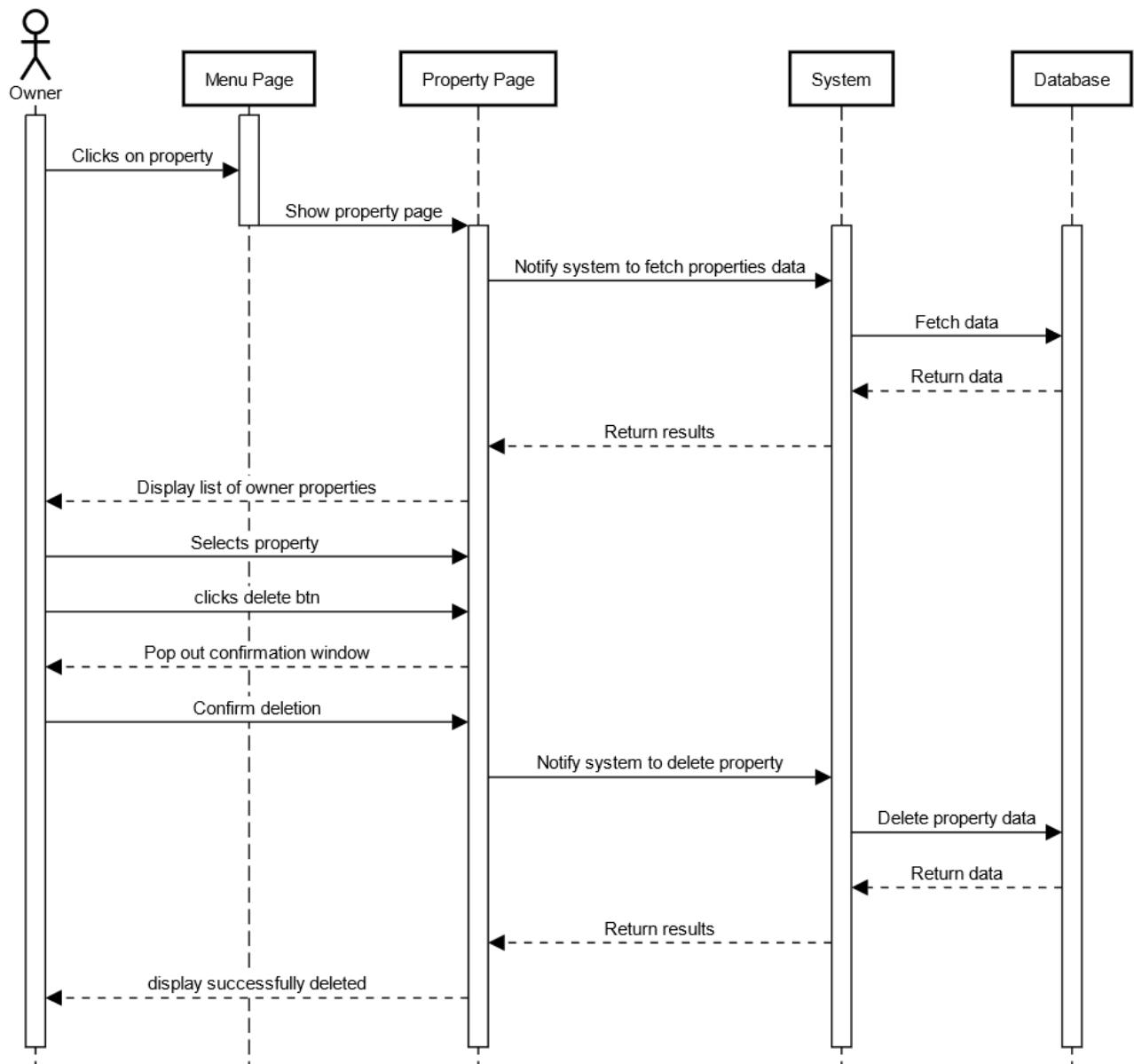
Use Case Name:	Manage Property
Description:	Owner can manage the property such as adding property, delete property, update existing property.
Primary Actor:	Owner
Precondition	Owner is logged into the system.
Postcondition	<ul style="list-style-type: none"> ● For add or update property, once the owner has submitted the property details and the system has validated the data, the property information is sent to the administrator for approval. ● For remove property, after the owner confirms the deletion of a property and the system successfully removes the data from the database, the property is no longer listed or accessible in the system. ● For view property details, system will display the property details fro owner to view once the owner navigates into the property page.
Main Success Scenario:	<ol style="list-style-type: none"> 1. The owner clicks on the property option on the Menu Page. 2. The Property Page is displayed to the owner. 3. The system is notified to fetch the owner's property data from the database. 4. The system retrieves the list of properties from the database. 5. The Property Page displays a list of the owner's properties. 6. For updating property, the owner selects a property from the list, then clicks the update button for the selected property. 7. The system fetches the detailed data for the property from the database. 8. The Property Page displays the existing details of the property. 9. The owner edits the property details and clicks submit. 10. The system validates the data and is sent to the database where the status is updated to 'pending'. 11. The updated property data is sent to admin for approval.

	<p>12. The system displays a message indicating that the update is pending approval.</p> <p>13. For deleting property, the owner also selects a property from the list, then clicks delete button.</p> <p>14. A confirmation window pops up asking the owner to confirm the deletion.</p> <p>15. The owner confirms the deletion of the property and the database deletes the property data and returns a success result to the system.</p> <p>16. The system displays a message to the owner that the property was successfully deleted.</p> <p>17. For adding property, the owner clicks add button and enter the property details, then clicks submit.</p> <p>18. The system validates the new data and upon successful validation, the database updates the status of the property to 'pending'.</p> <p>19. The new property is sent for admin approval.</p> <p>20. The system displays a message indicating that the new property is pending approval.</p>
Alternative Scenario:	If the information entered is invalid, the page displays a message indicating which credential is invalid and its specific errors.

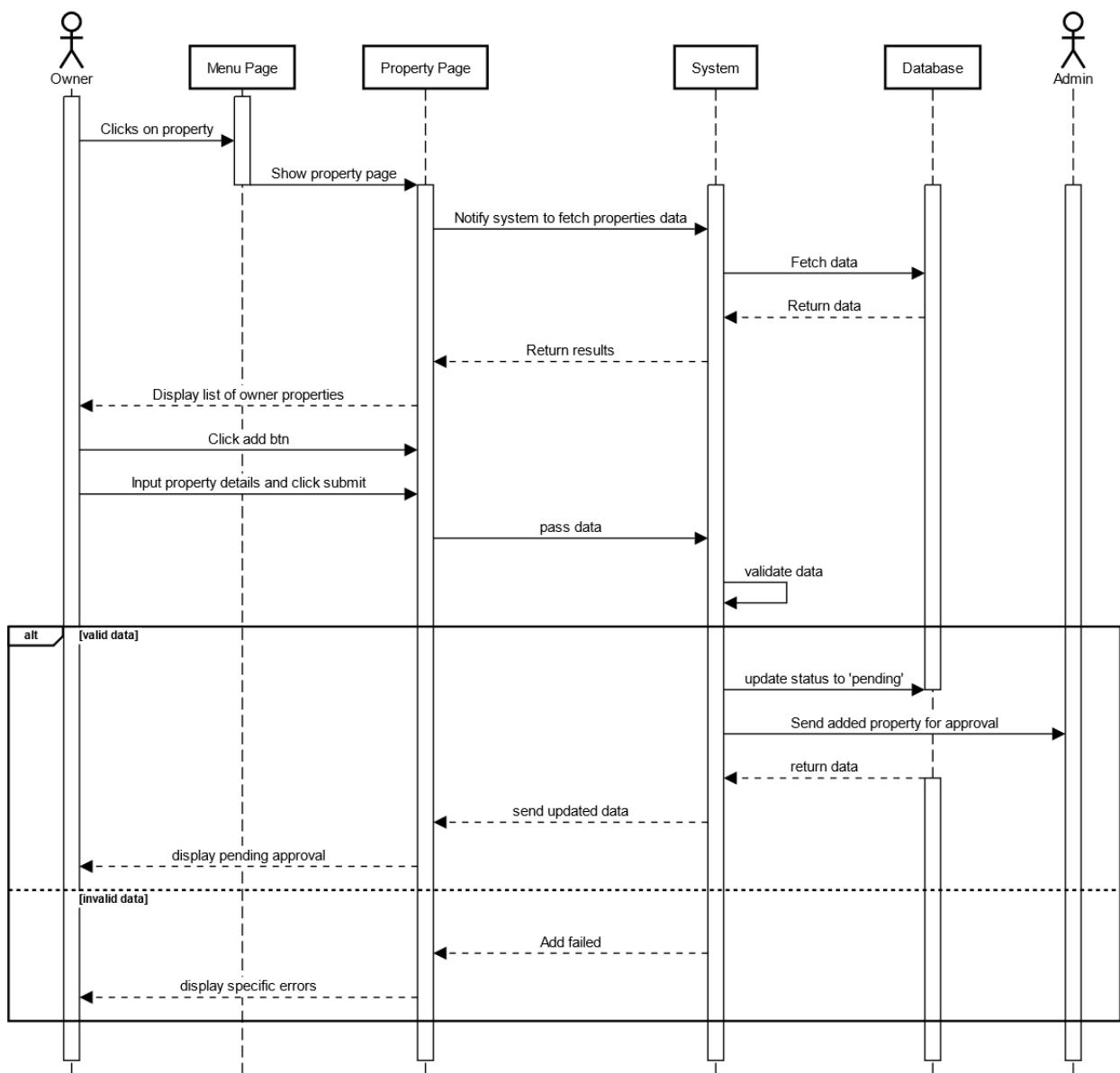
(update property)



(delete property)



(add property)

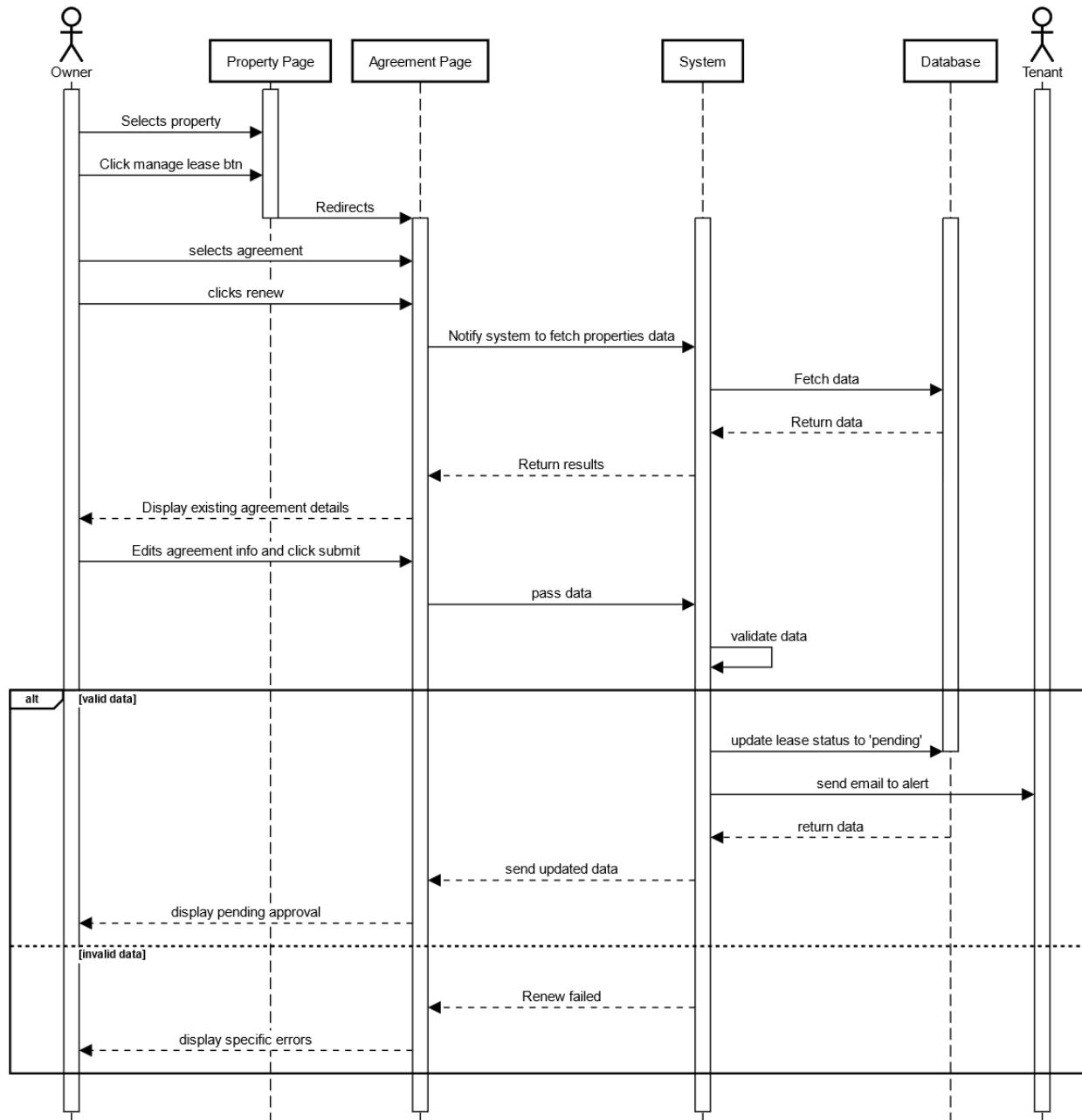


2.2.5 Manage Lease

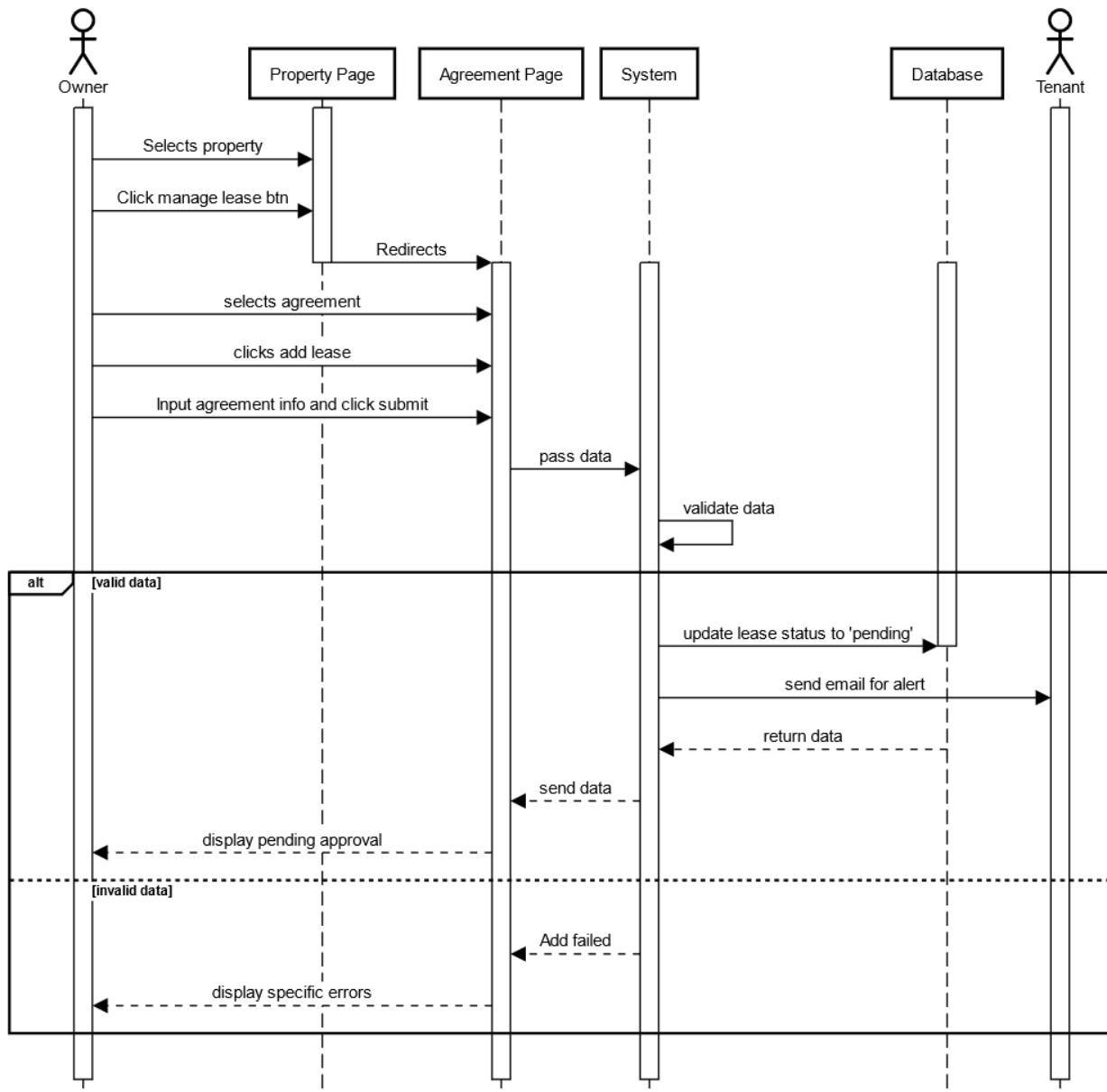
Use Case Name:	Manage Lease
Description:	Owner can add lease, terminate lease, or renew lease for tenant. Tenant then decides whether to accept or reject the lease.
Primary Actor:	Owner
Precondition	Owner is logged into the system.
Postcondition	<ul style="list-style-type: none"> ● The updated lease status to 'pending' is reflected in the tenant's account. ● The tenant can view the updated agreement and take action whether to accept or reject.
Main Success Scenario:	<ol style="list-style-type: none"> 1. The owner selects a property on the Property Page. 2. The owner clicks the manage lease button, which redirects them to the Agreement Page. 3. On the Agreement Page for renewing lease, the owner selects an agreement and clicks renew. 4. The system is notified to fetch the property's data. 5. The system retrieves the property data from the database. 6. The Agreement Page displays the existing agreement details to the owner. 7. The owner edits the agreement information and clicks submit. 8. The system validates the data and upon successful validation, the database updates the lease status to 'pending'. 9. An email is sent to alert the tenant about the pending lease agreement. 10. The system displays a message indicating that the lease agreement renewal is pending approval. 11. For adding a new lease, the owner clicks add button and input the necessary details and clicks submit. 12. The system validates the data and upon successful validation, the database updates the lease status to 'pending'. 13. An email is sent to alert the tenant about the pending lease agreement.

	<ol style="list-style-type: none">14. The system displays a message indicating that the lease agreement renewal is pending approval.15. For terminating lease, the owner selects an agreement and clicks the terminate lease button for the selected agreement.16. A confirmation window pops up asking the owner to confirm the termination of the lease.17. The owner confirms the termination.18. The system updates the lease status to 'terminated' in the database.19. The system sends an email to the tenant to alert them of the lease termination.20. The system returns the results indicating the lease has been successfully terminated.
Alternative Scenario:	If the information entered is invalid, the page displays a message indicating which credential is invalid and its specific errors.

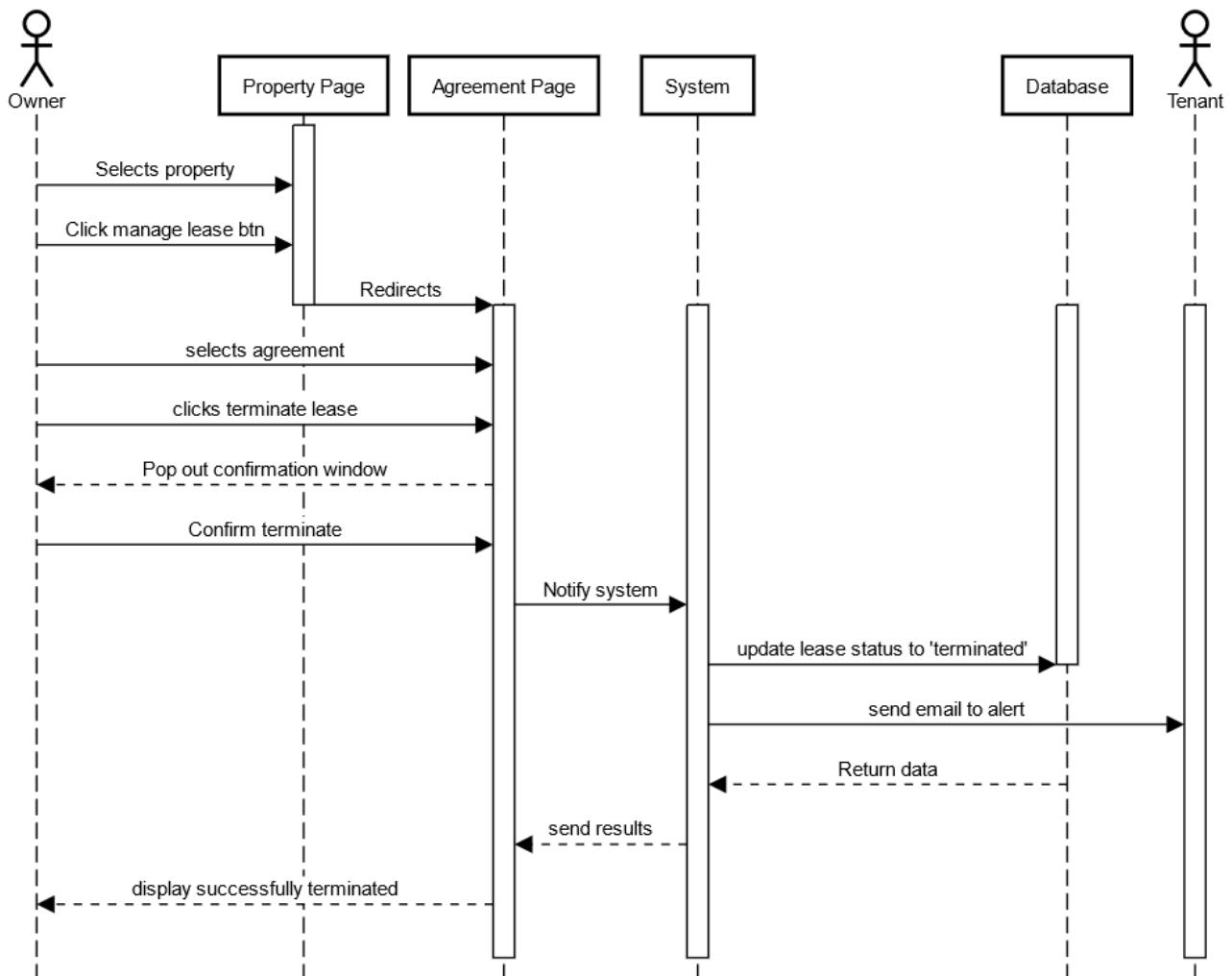
(renew lease)



(add lease)



(terminate lease)

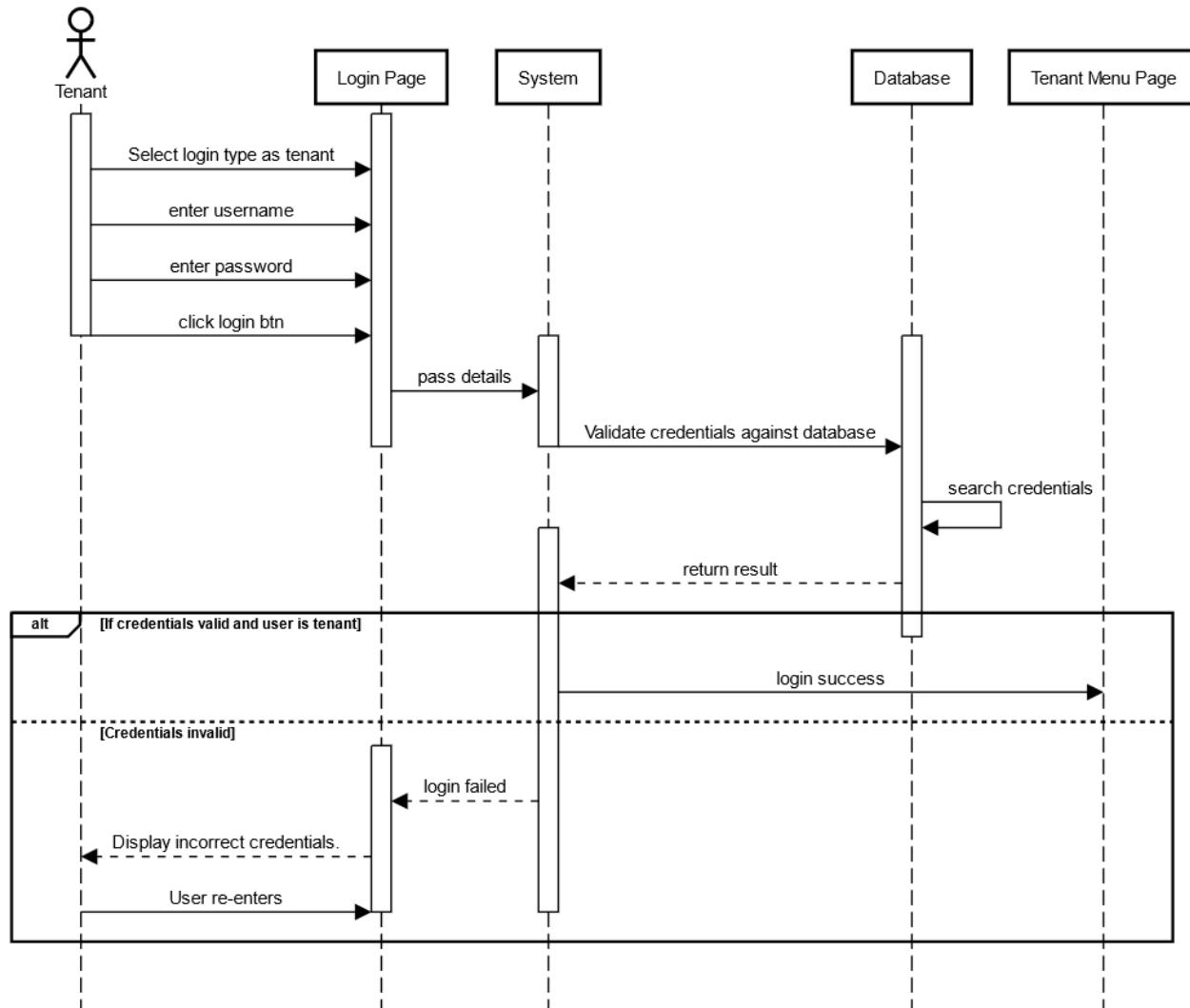


2.3 Tenant

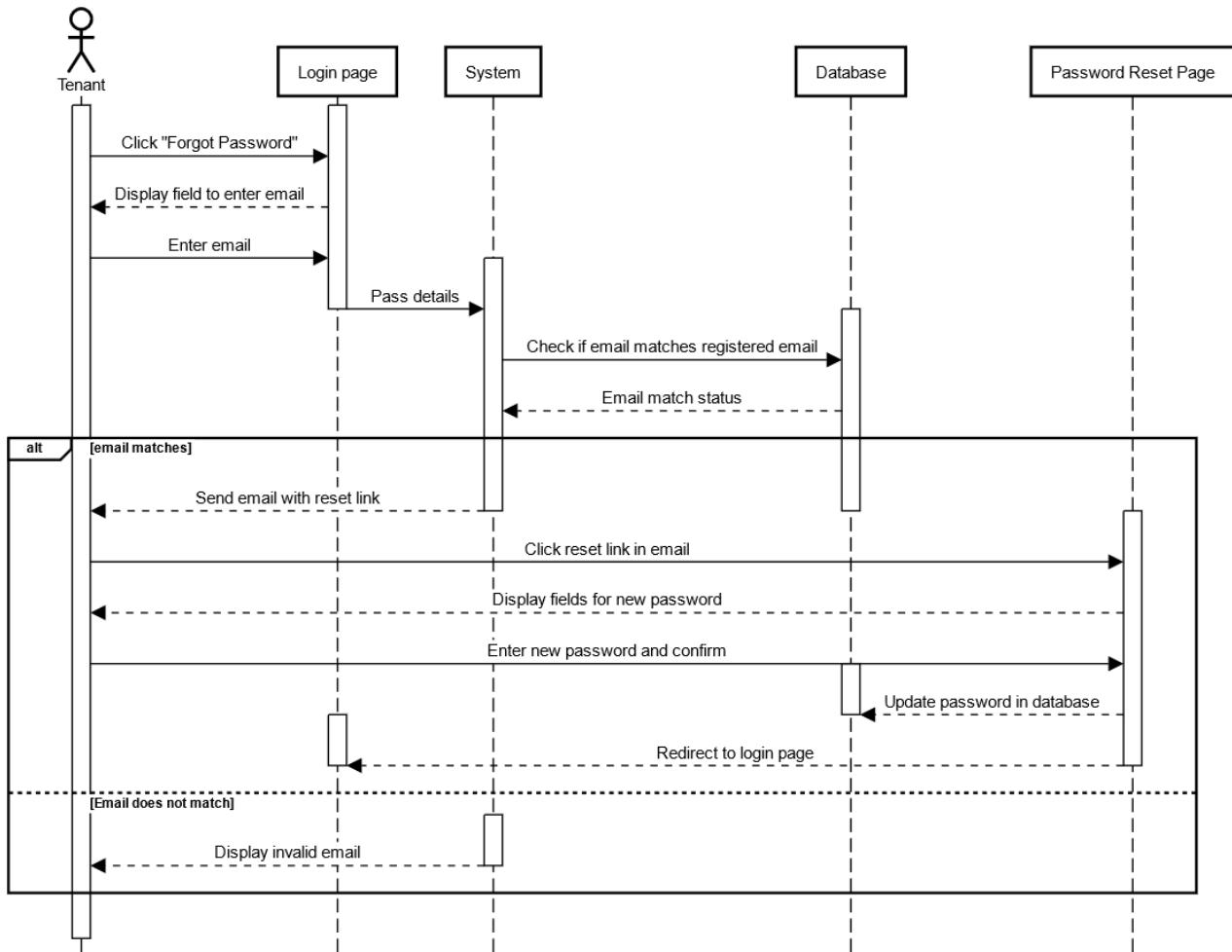
2.3.1 Login

Use Case Name:	Login
Description:	This use case outlines the process by which a tenant logs into the application, gains access to their account, and navigates to personalized features. It includes options for resetting password.
Primary Actor:	Tenant
Precondition	The tenant must already be registered with an account in the application's database.
Postcondition	The tenant is logged into their account and has access to personalized features and the tenant menu page.
Main Success Scenario:	<ol style="list-style-type: none"> 1. The tenant opens the login page. 2. The tenant selects the 'Tenant' user type. 3. The tenant enters their username and password into the appropriate fields. 4. The tenant clicks the login button. 5. The app validates the entered credentials against the stored user data in the database. 6. Upon successful validation, the app directs the tenant to the tenant menu page. 7. The tenant now has access to personalized features.
Alternative Scenario:	<ol style="list-style-type: none"> 1. If the entered credentials do not match, the system displays an incorrect username/password message. The tenant can attempt to log in again. 2. If the tenant selects the "forgot password" option, the system presents a field to enter their email. 3. The tenant enters their email and clicks the reset password button. 4. The system checks if the email matches the registered email. 5. If it matches, the system sends a password reset link to the user's email. 6. The tenant clicks on the link received in their email, which leads to a page to enter and confirm a new password. 7. The tenant enters the new password details and clicks submit.

- | | |
|--|--|
| | 8. The tenant is redirected to the login page to log in with the new password. |
|--|--|
8. The tenant is redirected to the login page to log in with the new password.



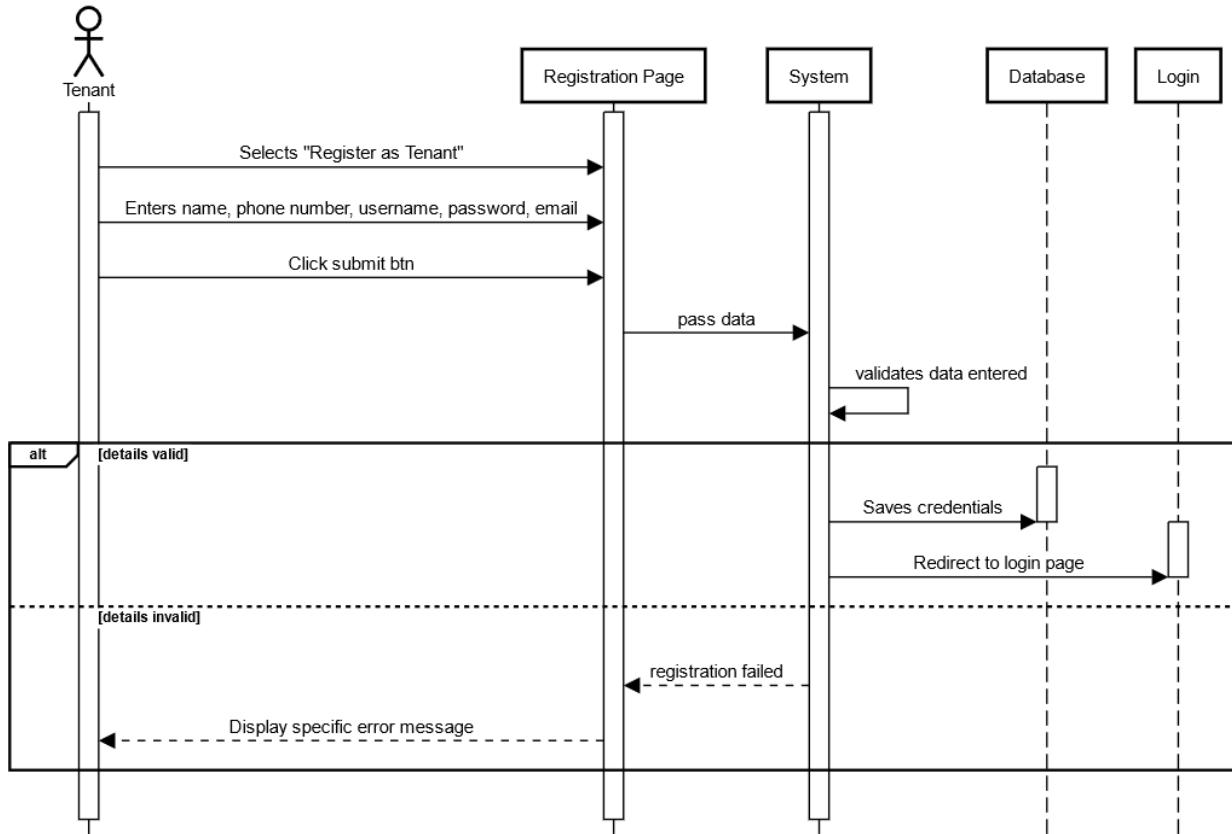
(forgot password)



2.3.2 Register

Use Case Name:	Register
Description:	This use case outlines the process for registering a new tenant account within the system.
Primary Actor:	Tenant
Precondition	No existing account should exist for the prospective tenants credentials.
Postcondition	A new tenant account is successfully registered in the system, and the tenant can log in with the registered credentials.
Main Success Scenario:	<ol style="list-style-type: none"> 1. User selects register as tenant.

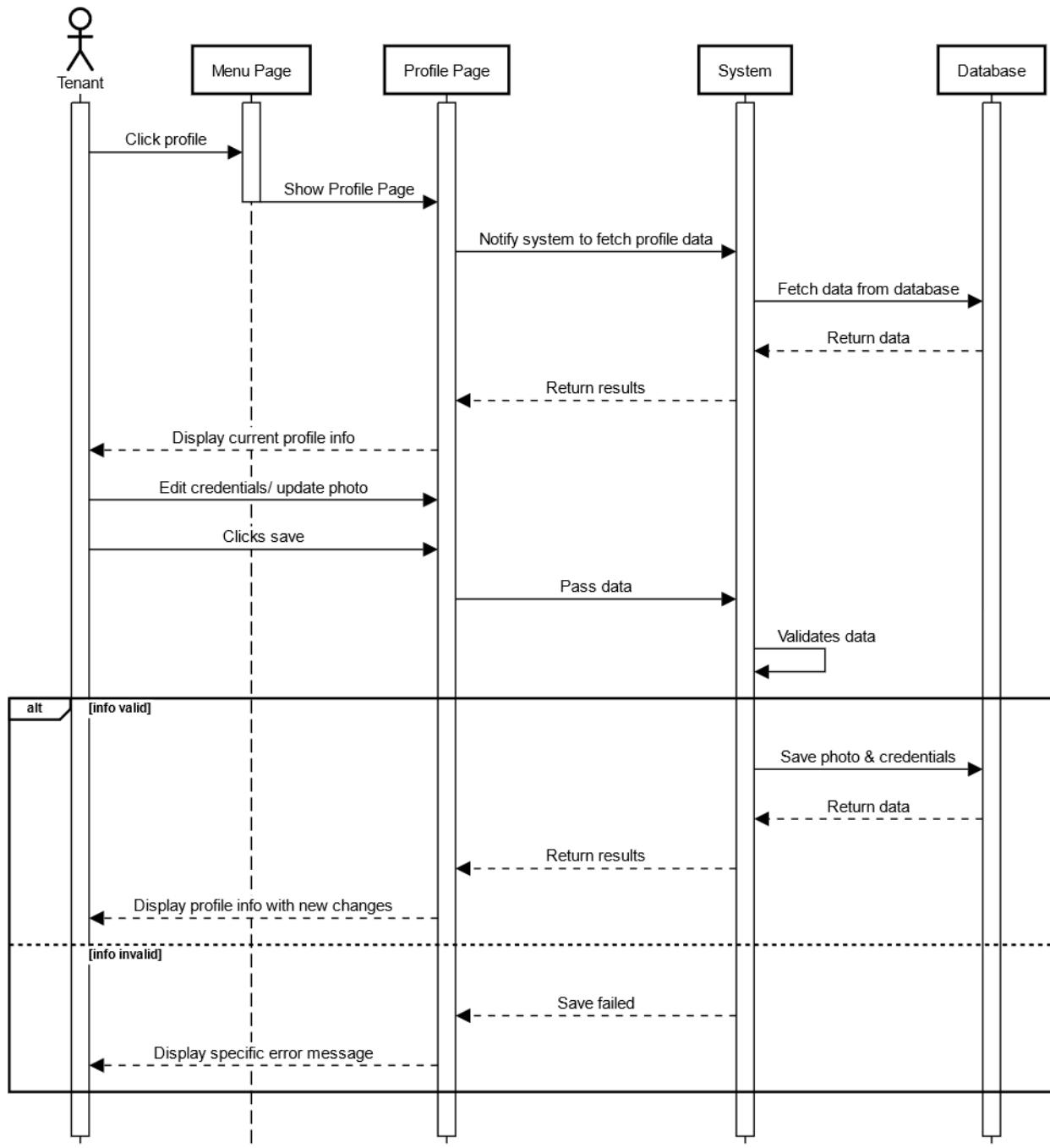
	<ol style="list-style-type: none">2. Tenant fills in details including name, phone number, username, password, email, then clicks create account to submit the registration details.3. System validates the credentials. If the information is complete with no errors, it proceeds to save the data to database. User is then redirect to login page. Else, if there are validation errors, the app displays the error messages respectively.4. Once the information is validated, the app creates a new account with the provided details.
Alternative Scenario:	<ol style="list-style-type: none">1. If the tenant tries to register with an email address that is already associated with an existing account, it will not proceed with registration and will display user already exists message and prompt the tenant to either log in with the existing account or use a different email address.2. If name or phone number contains special characters, it will display “special characters not allowed”.3. If phone number contains alphabets, it will display “phone number cannot contain alphabets”.4. If email address entered is invalid format, it will display “please enter a valid email”.



2.3.3 Create/Edit Profile

Use Case Name:	Create/Edit Profile
Description:	This use case describes the process by which a tenant can Create and edit their profile information within the system. The tenant has the option to edit their profile photo, credentials, and other personal details. The system ensures the integrity of the updates through validation before saving any changes to the database.
Primary Actor:	Tenant
Precondition	Tenant is logged into the system
Postcondition	<ol style="list-style-type: none"> 1. The user's profile is updated in the system with new information, if any changes are made. 2. The system's database reflects the latest state of the

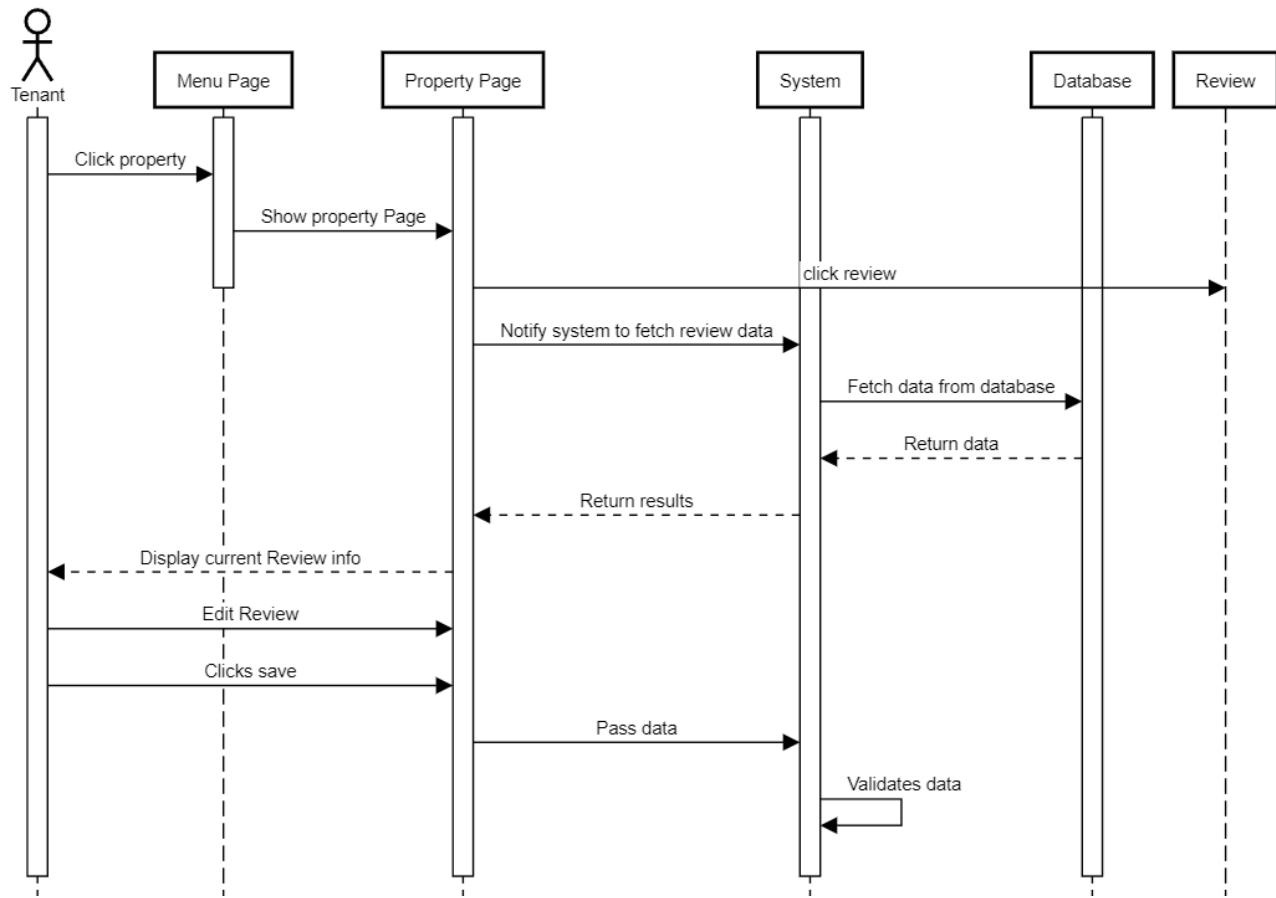
	<p>user's profile data after successful updates.</p>
Main Success Scenario:	<p>The user logs into the system and clicks on manage profile.</p> <p>2. The system displays the user's current profile information.</p> <p>3. The user selects the option to update their profile and they can choose to either update their profile photo or credentials.</p> <p>4. If updating the photo:</p> <ul style="list-style-type: none"> • The user uploads a new photo. • The system validates the photo format. • If valid, the system updates the profile photo in the database. <p>5. If removing the photo:</p> <ul style="list-style-type: none"> • A confirmation window is pop out for user to confirm the removal. • If user clicks confirm, the data gets updated in the database. <p>6. If updating credentials:</p> <ul style="list-style-type: none"> • The user edits their credentials (username, password, profile picture, email, name, phone number). • The user clicks 'save changes'. • The system checks the validity of the new credentials. • If valid, the system updates the credentials in the database.
Alternative Scenario:	<ul style="list-style-type: none"> • If user is not logged in, user cannot manage profile and will proceed to login page.



2.3.4 Review

Use Case Name:	Review
----------------	--------

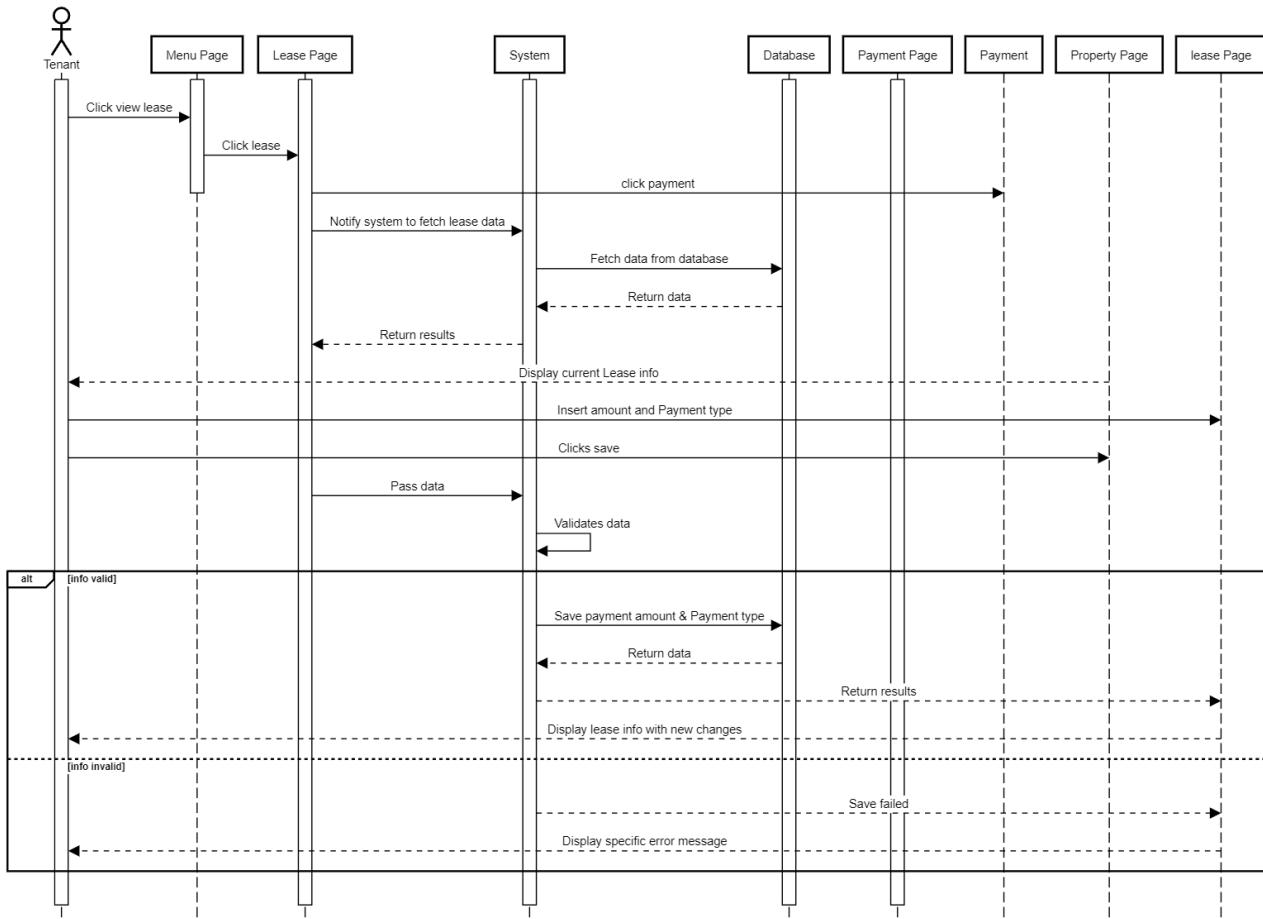
Description:	This use case describes the process by which an admin can update their profile information within the system. The admin has the option to update their profile photo, credentials, and other personal details. The system ensures the integrity of the updates through validation before saving any changes to the database.
Primary Actor:	Tenant
Precondition	<ul style="list-style-type: none"> ● The tenant must be logged in ● A property has to be rented by the tenant
Postcondition	<ul style="list-style-type: none"> ● The system's database reflects the latest state of the review data after successful updates.
Main Success Scenario:	<ul style="list-style-type: none"> ● The tenant is logged in ● The tenant will click on the property ● The system will validate if the property is rented by the owner. If it is valid, the tenant will click on review. If not then the tenant will have to choose another property ● Tenant will make a review with rating how many stars and a review description and update it in the database ● Tenant can edit or delete pre existing reviews and update it in the database. ●
Alternative Scenario:	<ul style="list-style-type: none"> ● There is no alternative scenario



2.3.5 Payment

Use Case Name:	Payment
Description:	This use case describes the process by which an admin can update their profile information within the system. The admin has the option to update their profile photo, credentials, and other personal details. The system ensures the integrity of the updates through validation before saving any changes to the database.
Primary Actor:	Tenant
Precondition	<ul style="list-style-type: none"> The tenant must be logged in A property has to be rented by the tenant
Postcondition	<ul style="list-style-type: none"> Payment is successfully registered into the system List of payment is shown and will be updated in the database

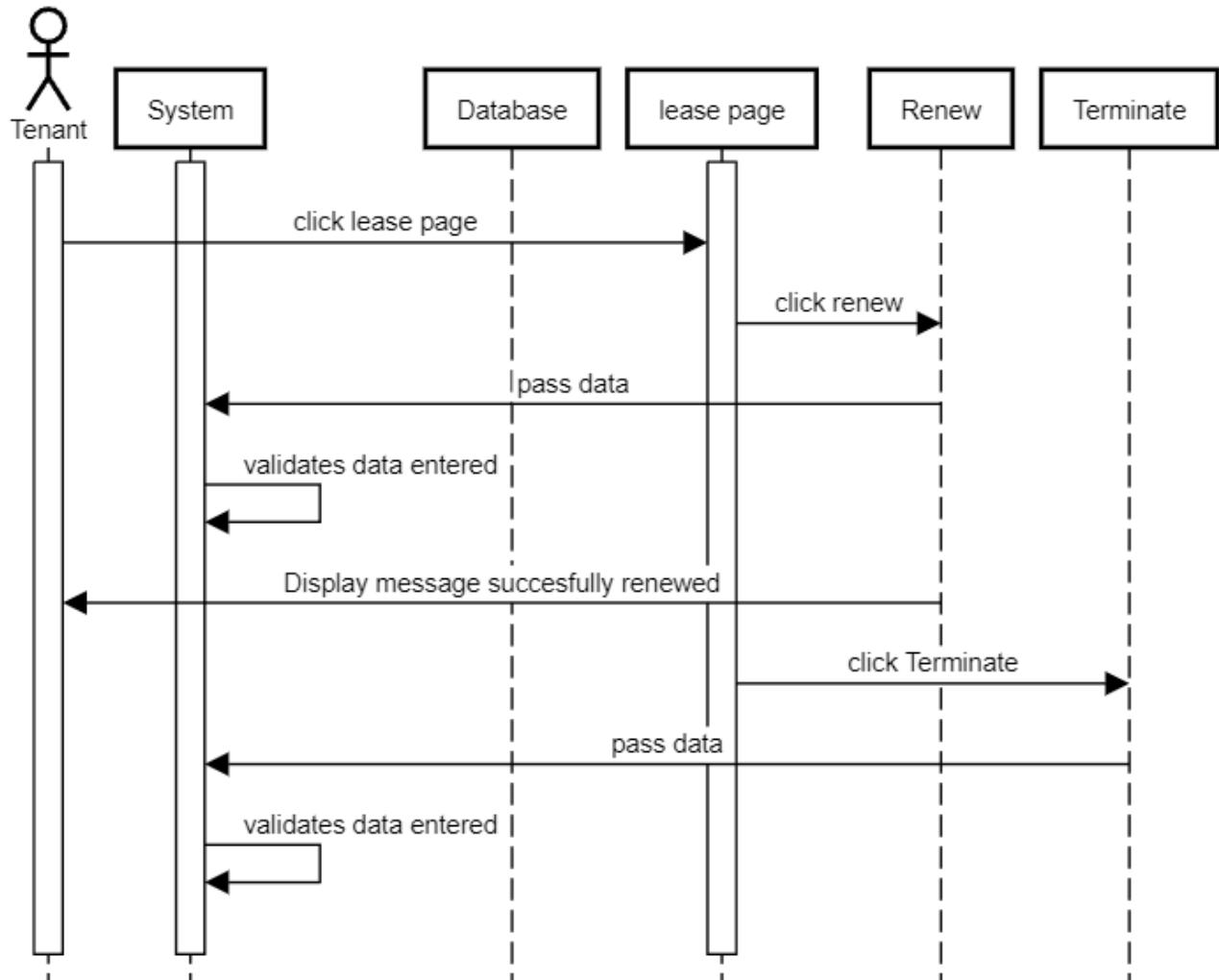
Main Success Scenario:	<ul style="list-style-type: none">● The tenant logs in and click on the payment● It will display lists of payments that is in the system● The tenant will click on the new payment entry and enter their name.● The system will check for validity and if true user will enter invoice and payment amount● The tenant can choose to edit or delete existing payments● The tenant can edit the invoice and amount and the system will validate it and it will update the payment status in the database● If the tenant deletes a payment, a confirmation window will pop up and update it in the database
Alternative Scenario:	<ul style="list-style-type: none">● If there is a system error while updating payment status in the database, the tenant, owner and admin will be notified, and the action is not completed.



2.3.6 Renew/Terminate Lease

Use Case Name:	Renew/Terminate Lease
Description:	This use case describes the process by which an admin can update their profile information within the system. The admin has the option to update their profile photo, credentials, and other personal details. The system ensures the integrity of the updates through validation before saving any changes to the database.
Primary Actor:	Tenant
Precondition	<ul style="list-style-type: none"> The tenant must be logged in A property has to be rented by the tenant
Postcondition	<ul style="list-style-type: none"> The updated lease status to 'pending' is reflected in the tenant's account.

	<ul style="list-style-type: none">● The tenant has the ability to view the updated agreement and take action.
Main Success Scenario:	<ul style="list-style-type: none">● The user is logged in.● The tenant clicks on 'manage lease agreement' and selects a property to manage the lease.● If a lease status is found, the tenant has the option to accept, renew or terminate the lease.● The system updates the lease status to 'pending' in the database and gives it to the owner and admin.
Alternative Scenario:	<ul style="list-style-type: none">● If tenant is not logged in, tenant cannot accept or terminate the lease agreement and will proceed to login page.● notified, and the action is not completed.

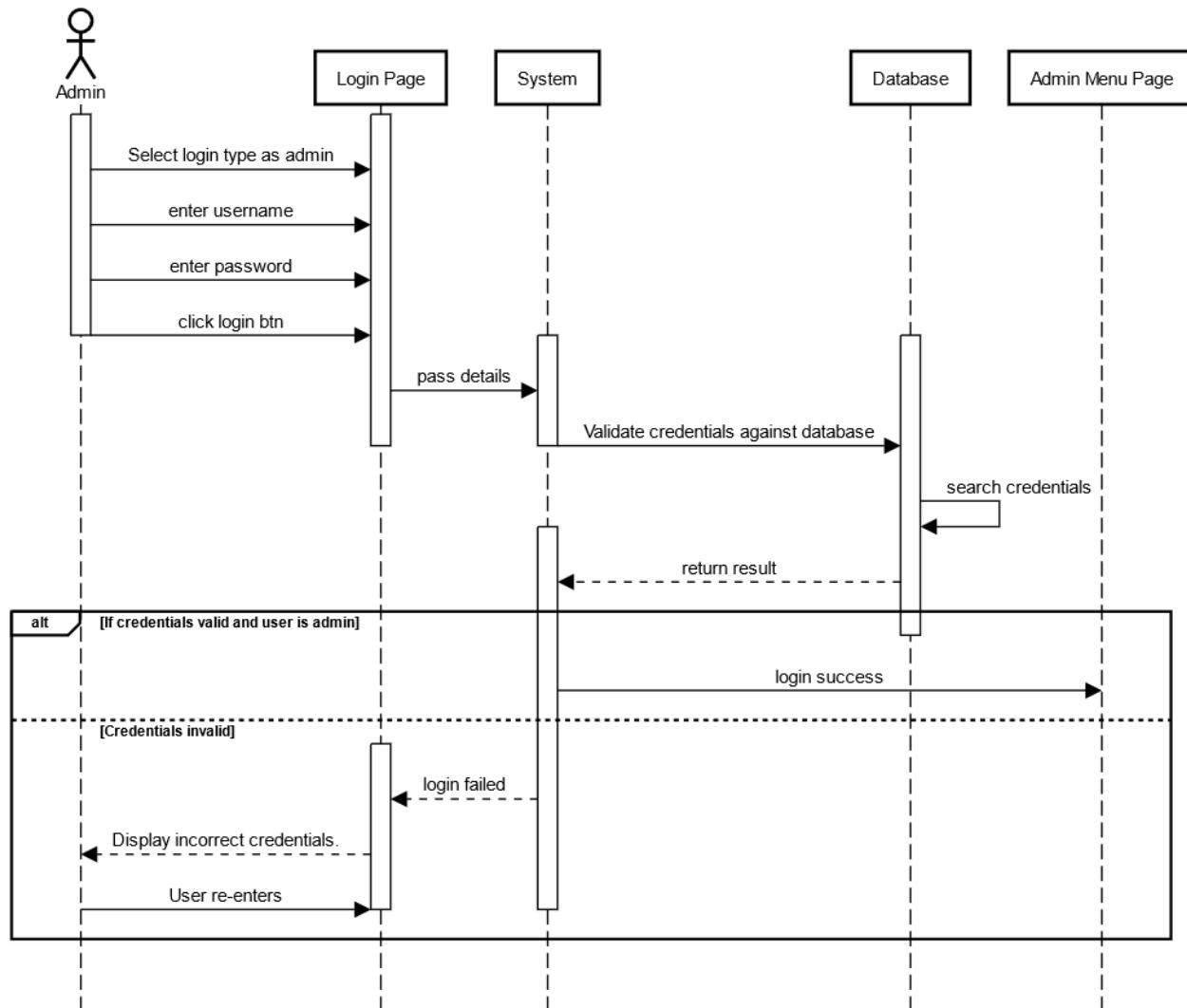


2.4 Admin

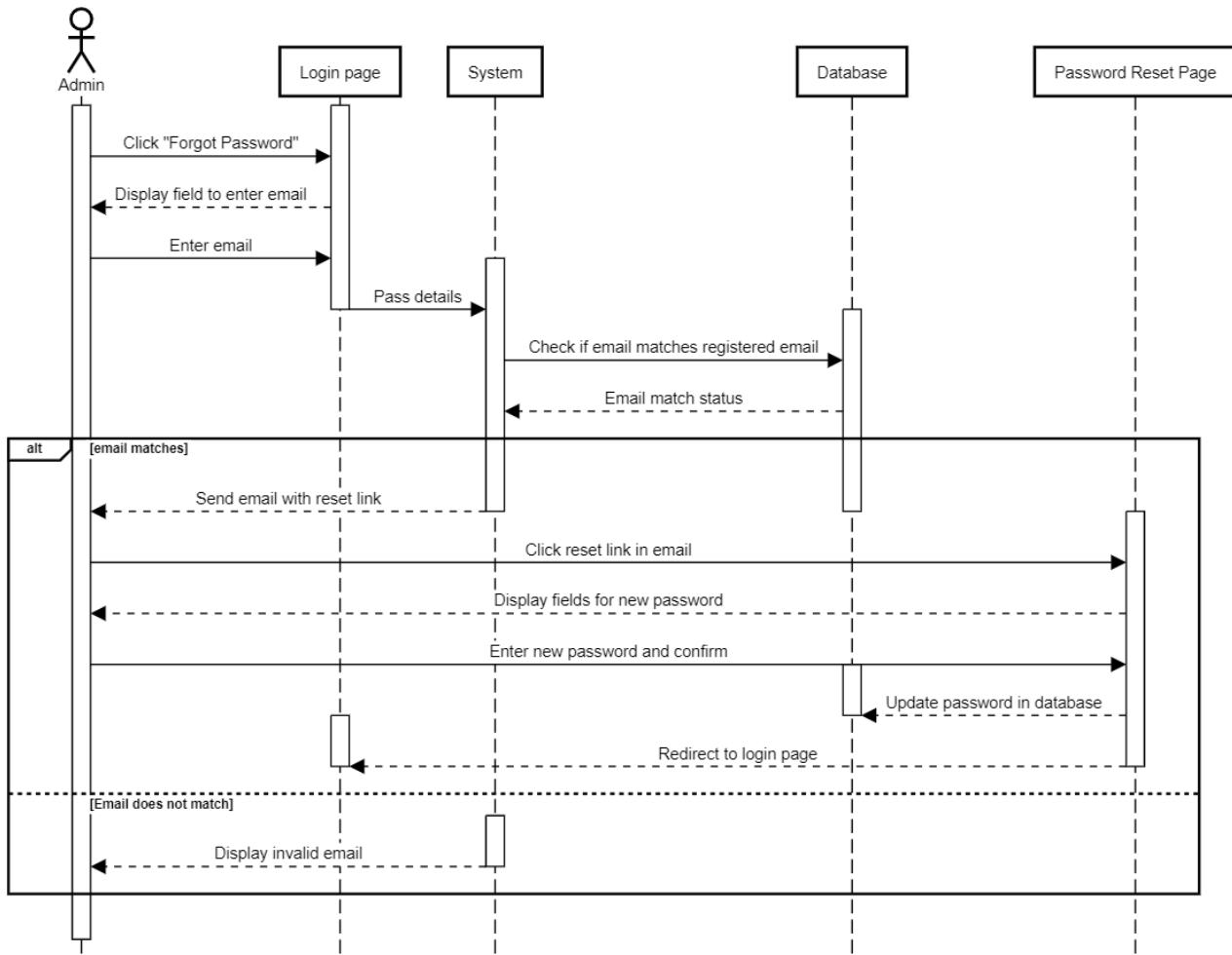
2.4.1 Login

Use Case Name:	Login
Description:	This use case outlines the process by which an admin logs into the application, gains access to their account, and navigates to personalized features. It includes options for handling password resets.
Primary Actor:	Admin

Precondition	The admin must already be registered with an account in the application's database.
Postcondition	The admin is logged into their account and has access to personalized features and the admin menu page.
Main Success Scenario:	<ol style="list-style-type: none"> 1. The admin opens the login page. 2. The admin selects the 'Admin' user type. 3. The admin enters their username and password into the appropriate fields. 4. The admin clicks the login button. 5. The app validates the entered credentials against the stored user data in the database. 6. Upon successful validation, the app directs the admin to the admin menu page. 7. The admin now has access to personalized features.
Alternative Scenario:	<ol style="list-style-type: none"> 1. If the entered credentials do not match, the system displays an incorrect username/password message. The admin can attempt to log in again. 2. If the admin selects the "forgot password" option, the system presents a field to enter their email. 3. The admin enters their email and clicks the reset password button. 4. The system checks if the email matches the registered email. 5. If it matches, the system sends a password reset link to the admin's email. 6. The admin clicks on the link received in their email, which leads to a page to enter and confirm a new password. 7. The admin enters the new password details and clicks submit. 8. The admin is redirected to the login page to log in with the new password.



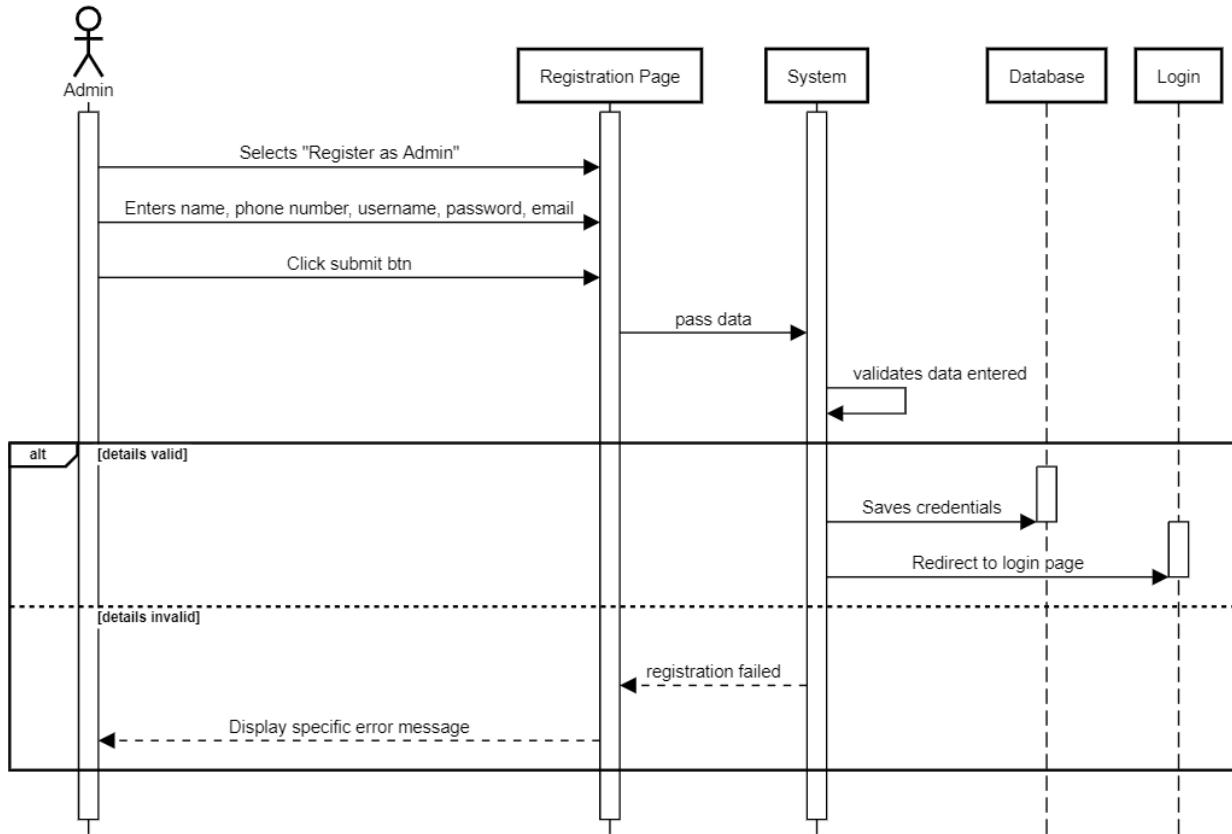
Forgot password



2.4.2 Register

Use Case Name:	Register
Description:	This use case outlines the process for registering a new admin account within the system. It ensures that the details provided are unique and that an existing admin approves the new account before it is activated and ready for use.
Primary Actor:	Admin
Precondition	No existing account should exist for the prospective admin's credentials.

Postcondition	A new admin account is successfully registered, activated in the system, and the admin has access to log in with the registered credentials.
Main Success Scenario:	<ol style="list-style-type: none">1. The user selects the option to register as an admin.2. The prospective admin fills in details including name, phone number, username, password, email, and admin ID, then clicks the 'create account' button to submit the registration details.3. The system validates the credentials. If the information is complete and contains no errors, the system creates a new admin account.4. Once approved, The system activates the new admin account and saves the new admin's details in the database.
Alternative Scenario:	<ul style="list-style-type: none">● If the existing admin rejects the registration request, the system notifies the prospective admin about the rejection.



2.4.3 Manage Profile

Use Case Name:	Manage Profile
Description:	This use case describes the process by which an admin can update their profile information within the system. The admin has the option to update their profile photo, credentials, and other personal details. The system ensures the integrity of the updates through validation before saving any changes to the database.
Primary Actor:	Admin
Precondition	Admin is logged into the system
Postcondition	3. The admin's profile details are updated according to their modifications.

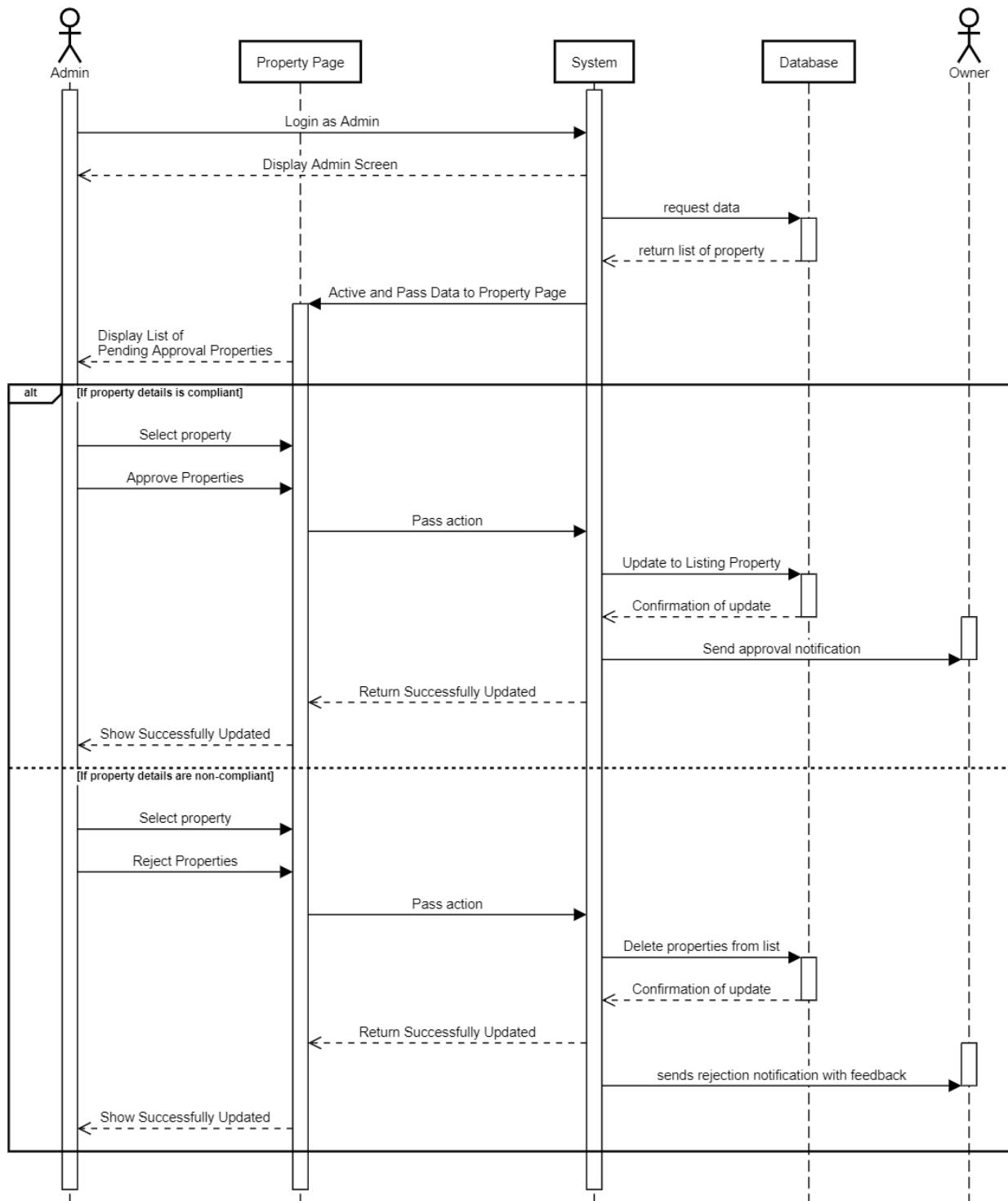
	<p>4. The system's database accurately reflects the current state of the admin's profile data.</p>
Main Success Scenario:	<ol style="list-style-type: none"> 1. The admin logs into the system and navigates to the profile section. 2. The system displays the admin's current profile information. 3. The admin selects the option to update their profile. 4. If updating the profile photo: <ul style="list-style-type: none"> • The admin uploads a new photo. • The system validates the photo format. • If valid, the system updates the profile photo in the database. 5. If removing the photo: <ul style="list-style-type: none"> • The system prompts a confirmation window. • The admin confirms the removal. • The system updates the database accordingly. 6. If updating credentials: <ul style="list-style-type: none"> • The admin edits their credentials such as username, password, email, name, and phone number. • The admin clicks 'save changes'. • The system checks the validity of the new credentials. • If valid, the system updates the credentials in the database.
Alternative Scenario:	If the admin is not logged in, they are redirected to the login page to access the manage profile feature.

2.4.4 Manage Property

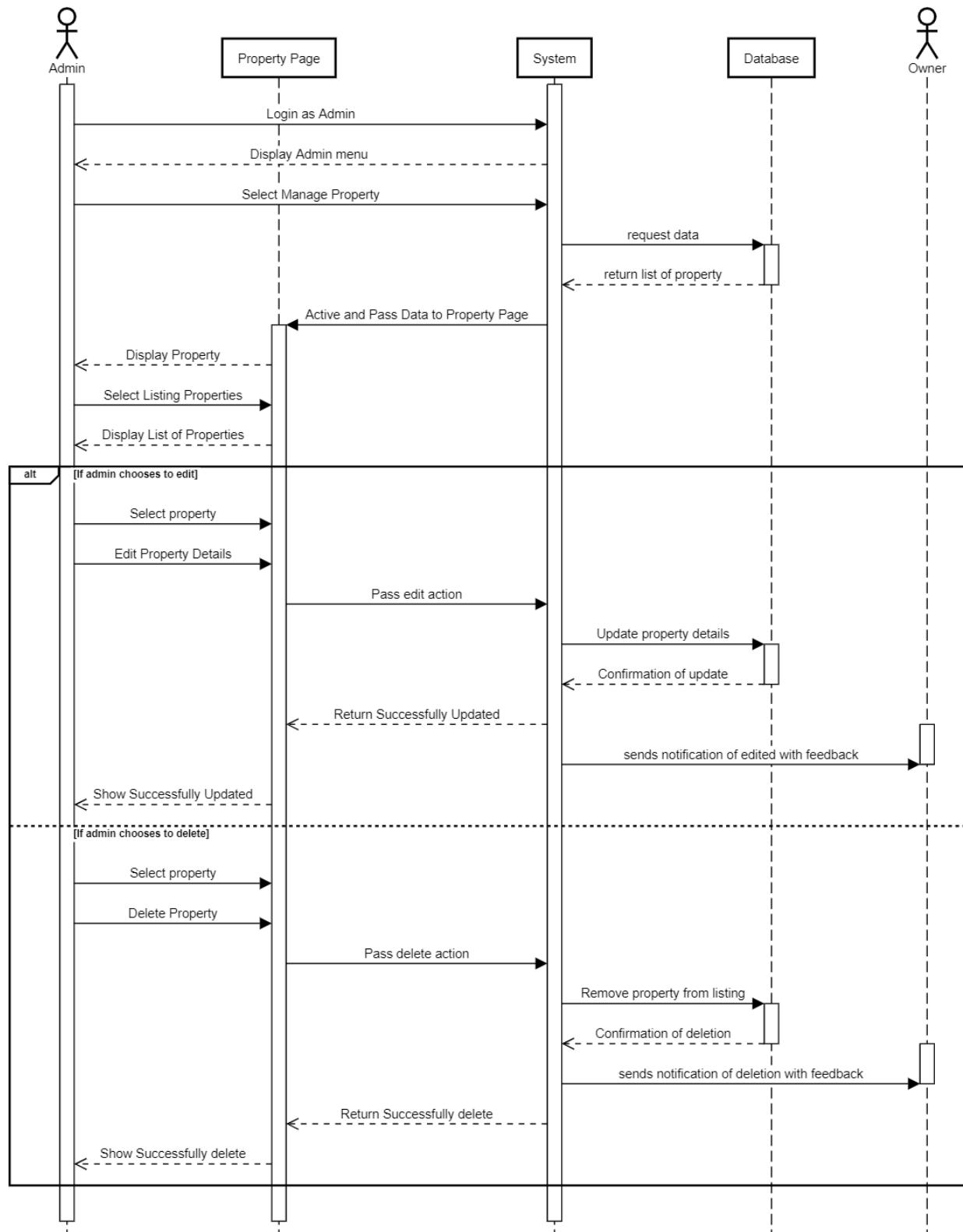
Use Case Name:	Manage Property
Description:	This use case involves the Admin managing property listings within the housepooling system. The Admin reviews, edits, and

	potentially removes properties based on compliance, completeness, and adherence to platform guidelines.
Primary Actor:	Admin
Precondition	The Admin is logged into the system
Postcondition	Property listings are managed and updated based on admin actions.
Main Success Scenario:	<ol style="list-style-type: none"> 1. Admin logs into the system and is presented with the admin menu. 2. Admin selects the 'Property' option from the Admin Screen. 3. System retrieves the list of properties from the database. 4. System displays the pending approval properties to the Admin. 5. Admin able to selects 'Manage Listing' to view Listing Properties'. 6. For pending properties, Admin reviews and either approves or rejects them: <ul style="list-style-type: none"> • If approved, the system updates the property status to listed and notifies the owner. • If rejected, the system deletes the property from the pending list and notifies the owner with feedback. 7. For listed properties, Admin can edit details or delete the property: <ul style="list-style-type: none"> • If edited, the system updates the property details and notifies the owner. • If deleted, the system removes the property from the listing and notifies the owner.
Alternative Scenario:	<ol style="list-style-type: none"> 1. If the property details are incomplete or violate guidelines, admin rejects the property listing with appropriate feedback.

Manage Pending Approval Property

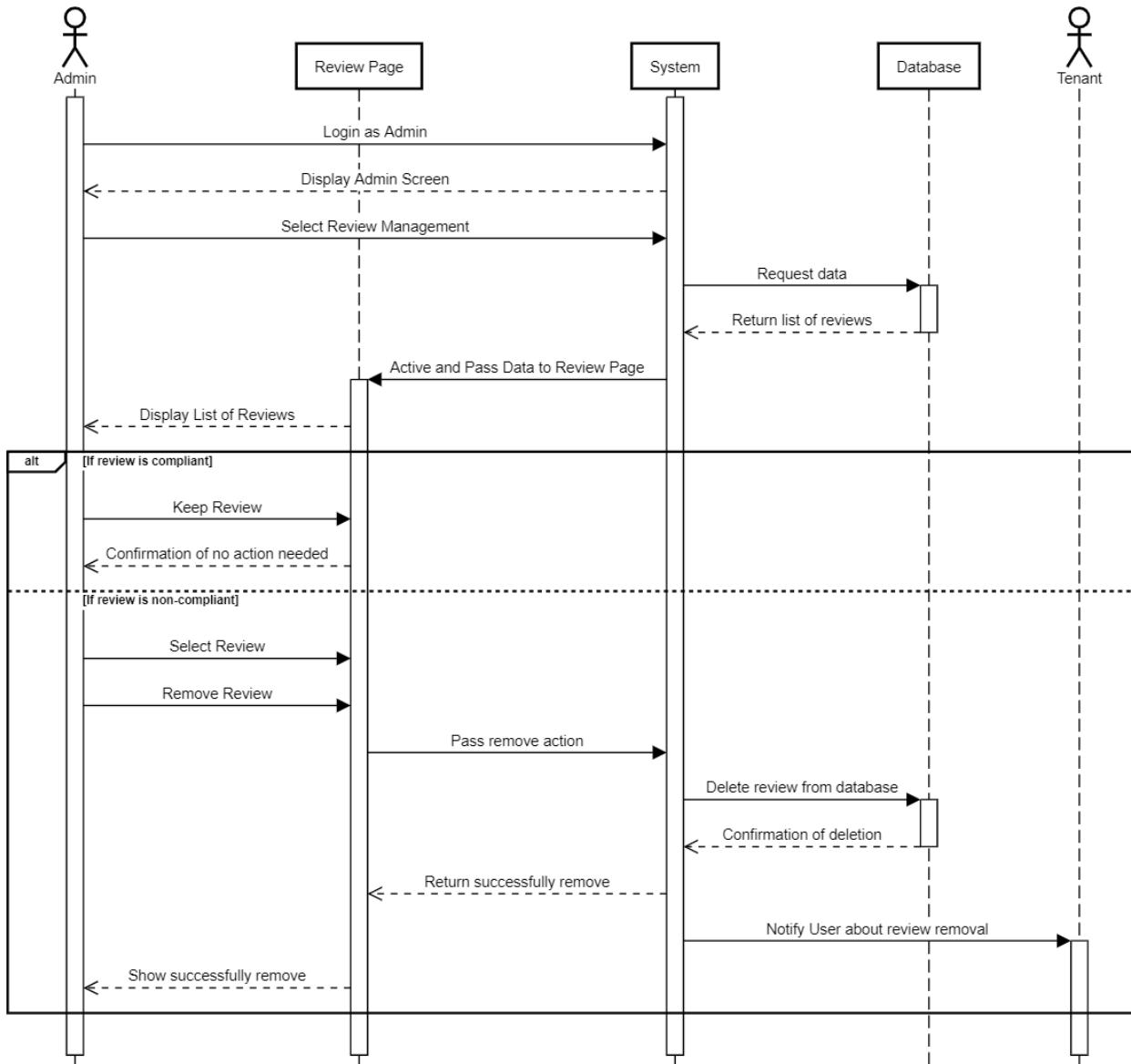


Manage Listing Property



2.4.5 Manage Reviews

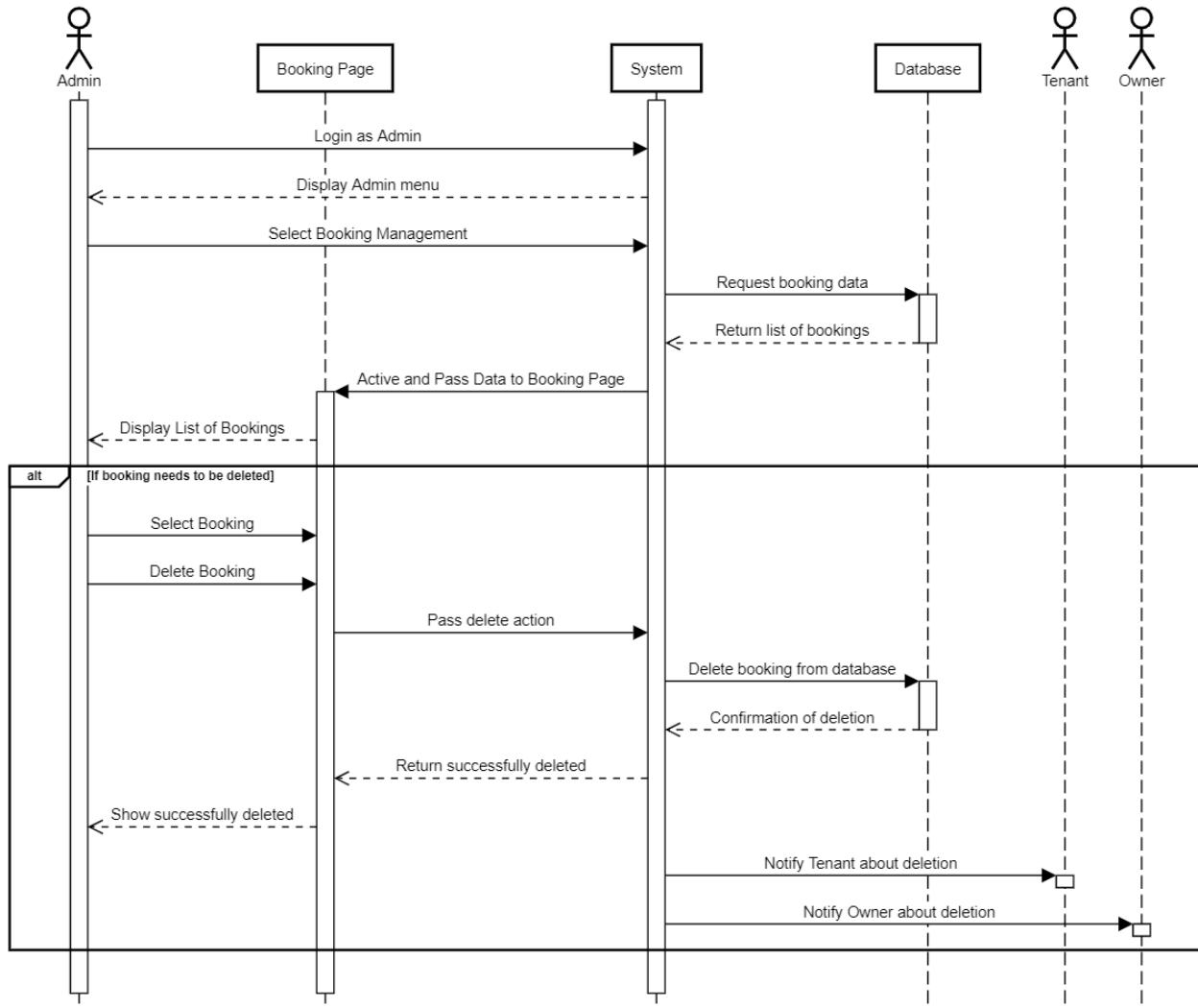
Use Case Name:	Manage Reviews
Description:	This use case involves the moderation of property reviews by the Admin. The Admin, after logging into the system, accesses the review management section. In this section, the Admin can choose to view a list of reviews submitted by tenants or remove reviews that do not comply with guidelines. All changes made by the Admin are saved in the database.
Primary Actor:	Admin
Precondition	The Admin is logged into the system
Postcondition	The Admin has successfully moderated property reviews.
Main Success Scenario:	<ol style="list-style-type: none"> 1. The Admin logs into the system. 2. The Admin accesses the review management section. 3. The Admin chooses to view a list of reviews submitted by tenants. 4. The Admin reviews the list of tenant-submitted reviews. 5. The Admin has the option to remove reviews that do not comply with guidelines. 6. The Admin saves the changes made to the reviews in the database.
Alternative Scenario:	<ol style="list-style-type: none"> 1. If any reviews are found to contain inappropriate content, the Admin removes such reviews and notifies the respective user.



2.4.6 Manage Booking

Use Case Name:	Manage Booking
Description:	This use case involves the admin managing the booking records for properties within the system. The admin can view and delete booking records as necessary to maintain accurate and up-to-date booking information.

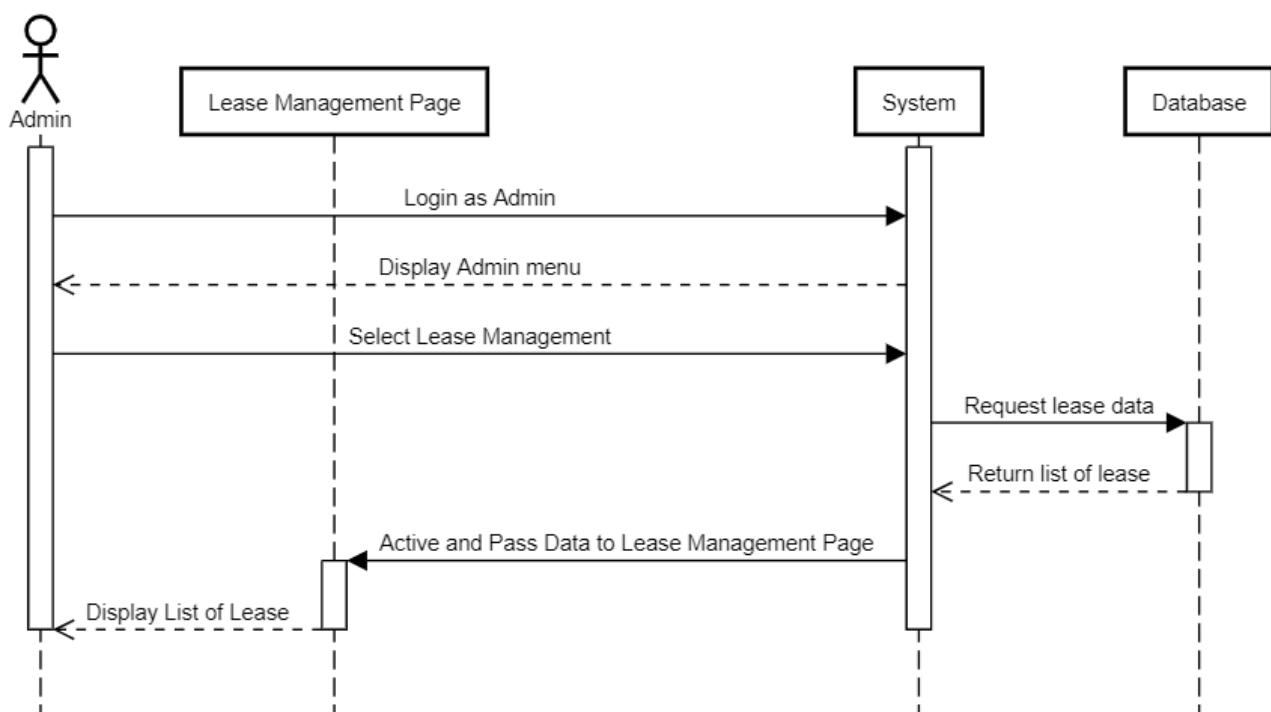
Primary Actor:	Admin
Precondition	Admin is logged into the system
Postcondition	Property bookings are accurately reflected in the system based on the admin's updates and management.
Main Success Scenario:	<ol style="list-style-type: none">1. The admin logs into the system.2. The admin navigates to the booking management section.3. The admin reviews the booking details for accuracy and completeness.4. If necessary, the admin deletes specific booking records.5. The system saves any changes made to the booking information in the database.
Alternative Scenario:	-



2.4.7 Manage Lease

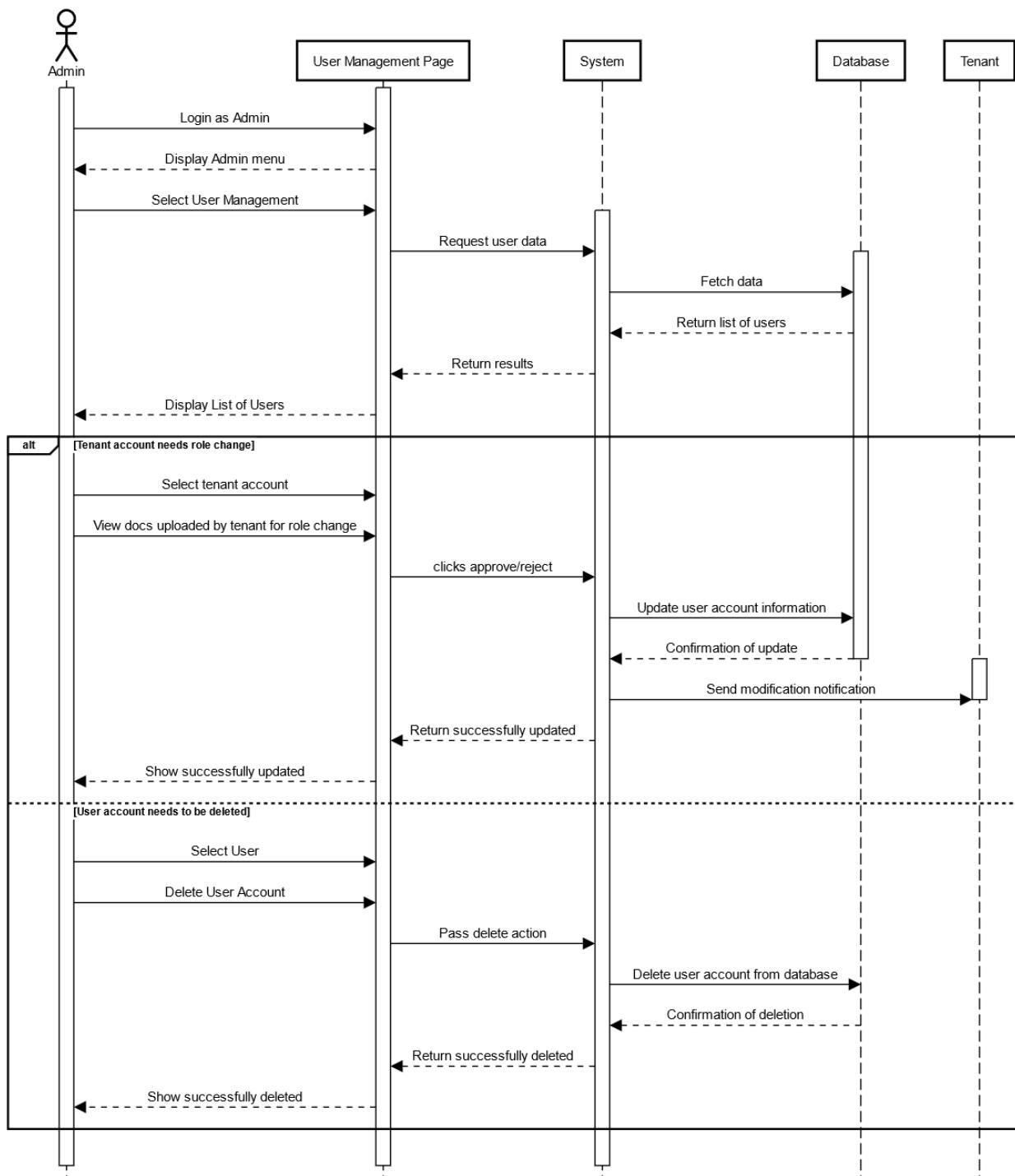
Use Case Name:	Manage Lease
Description:	This use case shows that the admin is able to view the lease between the owner and tenant in the system.
Primary Actor:	Admin
Precondition	Admin is logged into the system

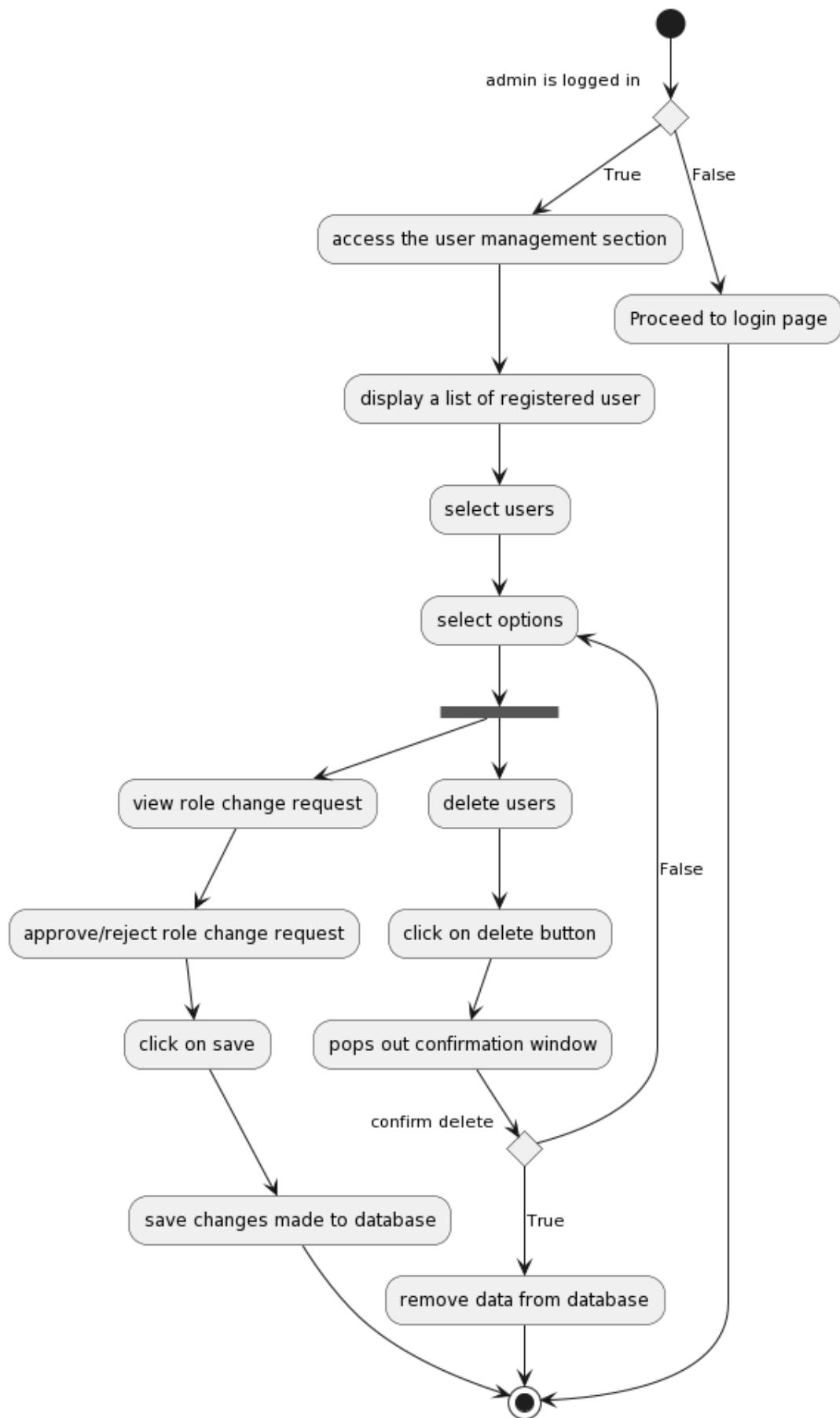
Postcondition	The Admin has successfully accessed and reviewed the lease agreements between Tenants and Owners.
Main Success Scenario:	<ol style="list-style-type: none"> 1. Admin logs into the system and is presented with the Admin menu. 2. Admin selects the 'Lease Management' option from the menu. 3. The system requests lease data from the database. 4. The Admin views a list of active lease agreements between Tenants and Owners displayed by the lease management page
Alternative Scenario:	-



2.4.8 Manage User

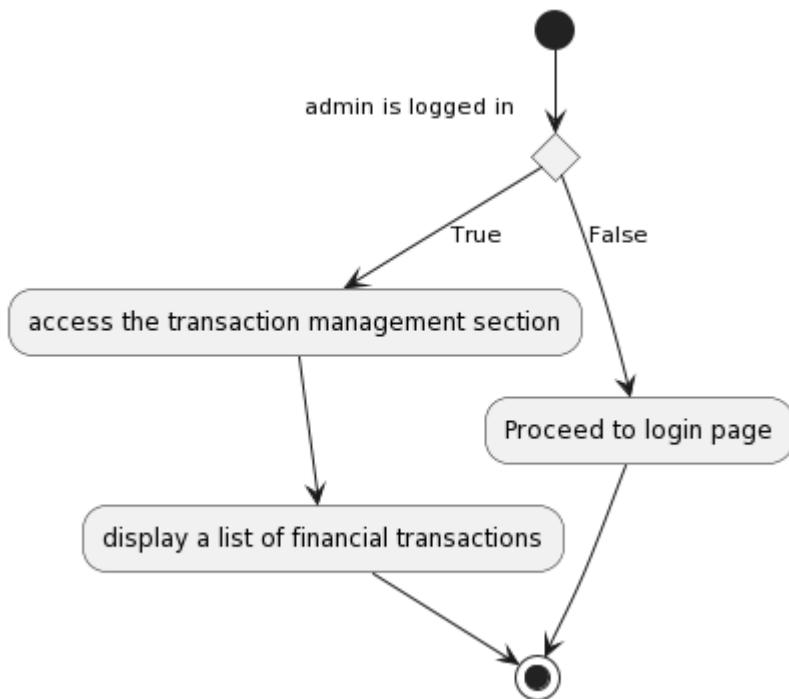
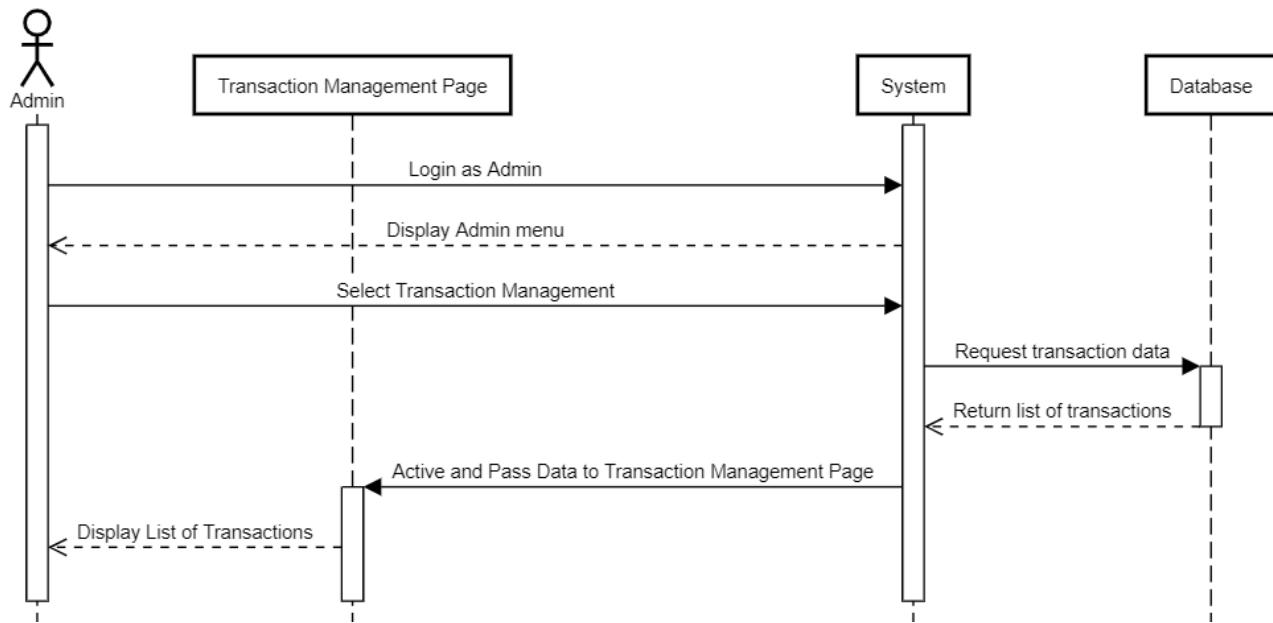
Use Case Name:	Manage User
Description:	This use case describes the admin's ability to manage user accounts within the system. The admin has the authority to manage roles change and delete user accounts to ensure proper user management and system security.
Primary Actor:	Admin
Precondition	Admin is logged into the system
Postcondition	The system's user accounts are updated to reflect the changes made by the admin.
Main Success Scenario:	<ol style="list-style-type: none"> 1. The admin logs into the system and navigates to the user management section. 2. The admin views the list of registered users within the system. 3. The admin can manage role change of tenant to owner by selecting a tenant account to modify roles. 4. The admin views the role change requests submitted by the tenant. The admin can approve or reject based on the tenant's uploaded application. The admin can reject tenant if the tenant has an active lease or the tenant has late payment history. 5. The admin can also choose to delete a user account if necessary. 6. The system saves the changes made by the admin to the database, updating the user accounts accordingly.
Alternative Scenario:	<ul style="list-style-type: none"> • If the admin cannot find a specific user in the list, they may use a search function to locate the user account. • If the admin needs to bulk update user accounts or permissions, they may select multiple users and apply the changes in a batch.





2.4.9 Manage Transactions

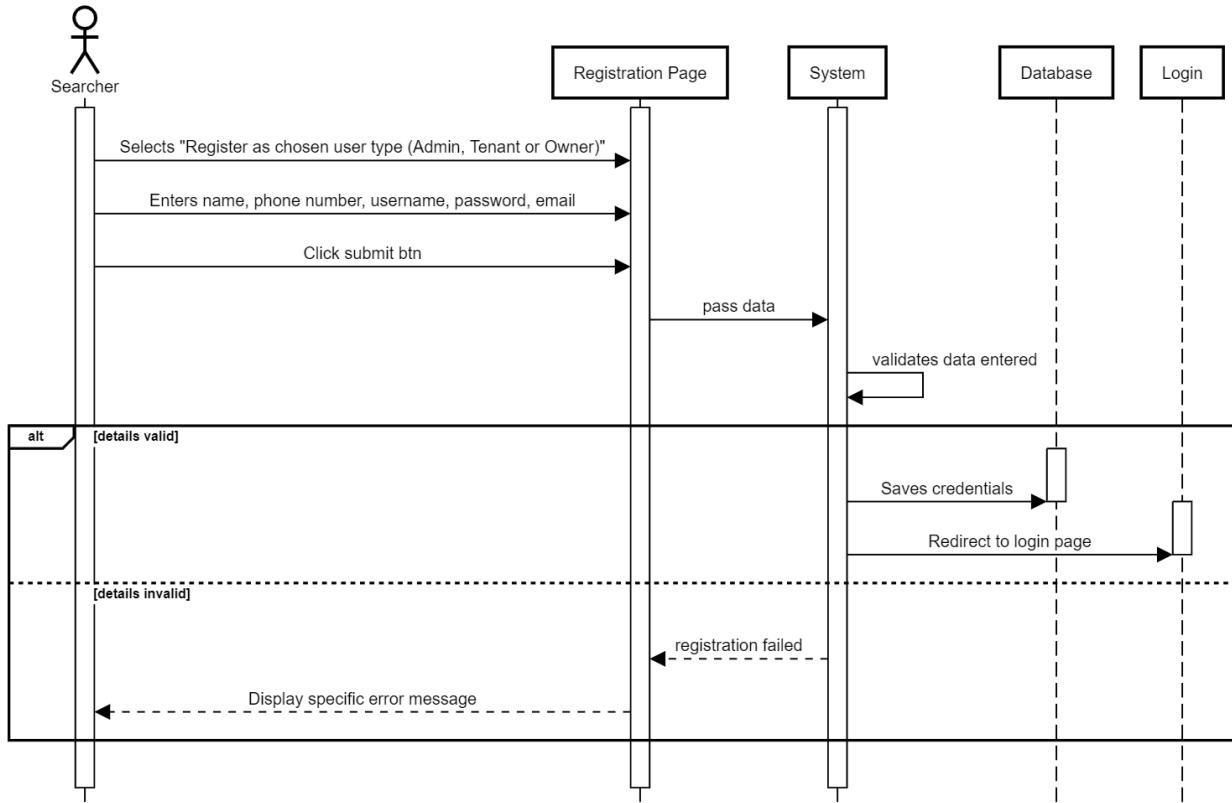
Use Case Name:	Manage Transactions
Description:	This use case details the process by which an Admin accesses and reviews a list of financial transactions in a system.
Primary Actor:	Admin
Precondition	Admin is logged into the system
Postcondition	The Admin has reviewed the transaction records.
Main Success Scenario:	<ol style="list-style-type: none">1. Admin logs into the system and is presented with the Admin menu.2. Admin selects the 'Transaction Management' option from the menu.3. The system requests transaction data from the database.4. The database returns a list of all financial transactions to the system.5. The system activates the Transaction Management Page and passes the transaction data to it.6. The Transaction Management Page displays the list of transactions to the Admin for review.
Alternative Scenario:	-



2.5 Searcher

2.5.1 Register

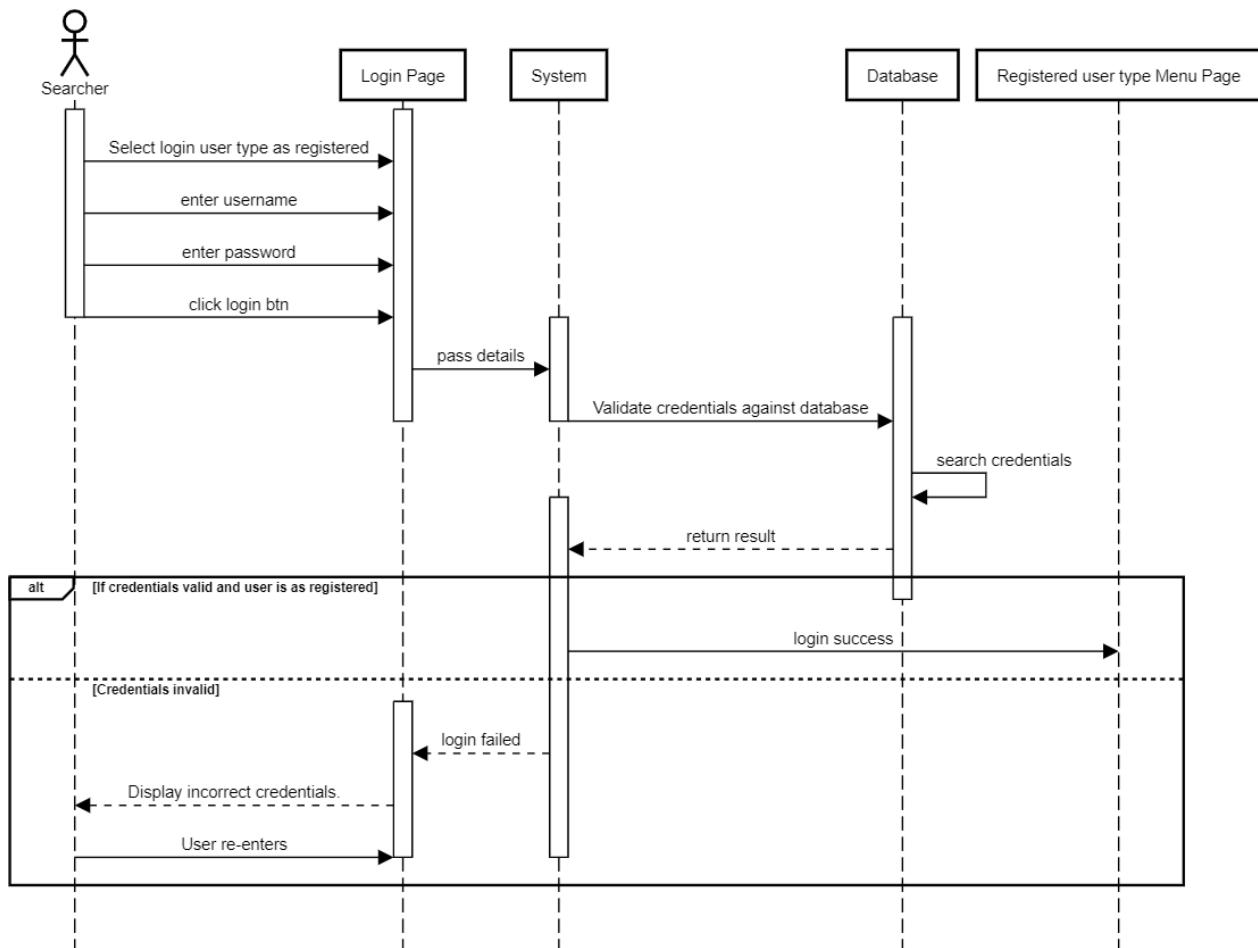
Use Case Name:	Register
Description:	This use case is for the process of registering a new account based on the searcher chosen type. It assures the details provided by the user are unique in the system.
Primary Actor:	Searcher
Precondition	No existing account should exist for the prospective searcher credentials.
Postcondition	The user has an account based on its chosen user type (Admin, Tenant or Owner)
Main Success Scenario:	<ol style="list-style-type: none"> 1. User selects register as chosen user type 2. User fills in details including name, phone number, username, password, email, then clicks create account to submit the registration details. 3. System validates the credentials. If the information is complete with no errors, it proceeds to save the data to the database. User is then redirected to the login page. 4. Once the information is validated, the app creates a new account for the user with the provided details.
Alternative Scenario:	<ol style="list-style-type: none"> 1. If the searcher tries to register with an email address that is already associated with an existing account, it will not proceed with registration and will display a user already existing message and prompt the owner to either log in with the existing account or use a different email address. 2. If a name or phone number contains special characters, it will display “special characters not allowed”. 3. If the phone number contains alphabets, it will display “phone number cannot contain alphabets”. 4. If the email address entered is invalid format, it will display “please enter a valid email”.

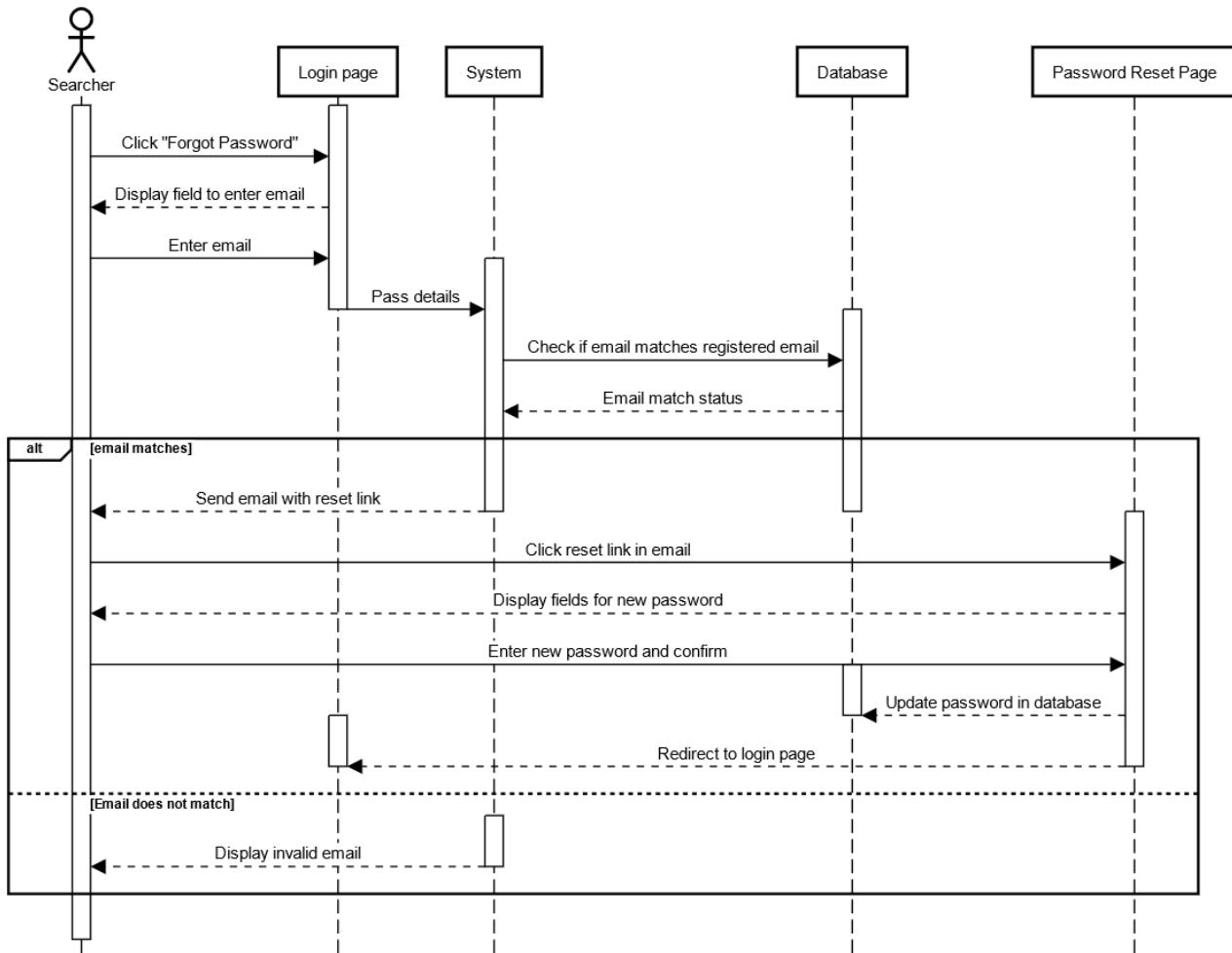


2.5.2 Login

Use Case Name:	Login
Description:	This use case outlines the process by which a searcher logs into the application, gains access to their account, and navigates to personalized features. It also includes an option for password resets.
Primary Actor:	Searcher
Precondition	User must have an existing account
Postcondition	User is logged in and has the accessed based on the user type that registered
Main Success Scenario:	<ol style="list-style-type: none"> 1. User opens the login page, chooses the type of user to be logged in, and enters their username and password.

	<ol style="list-style-type: none">2. User clicks the login button.3. The app validates the entered credentials against the stored user data in the database. If credentials are valid and login type is owner, it will proceed to the owner menu page. Else, if the login type is other users, it will proceed to the other users menu page.4. Upon successful login, the owner has access to personalized features such as manage their property listings, profile management, manage reviews, renew or terminate lease.
Alternative Scenario:	<ol style="list-style-type: none">1. If the credentials do not match, the system displays an incorrect username/password message. The user has the option to log in again.2. If the user selects the "forgot password" option, the system displays a field to enter the email.3. The user fills in their email and clicks the reset password button.4. The system checks if the email matches the registered email.5. If it matches, the system sends a link to the email for password reset.6. The user clicks on the link received in their email, which leads them to a page to enter a new password and confirm it.7. The user fills in the new password details and clicks submit.8. The user is redirected to the login page to log in with the new password.

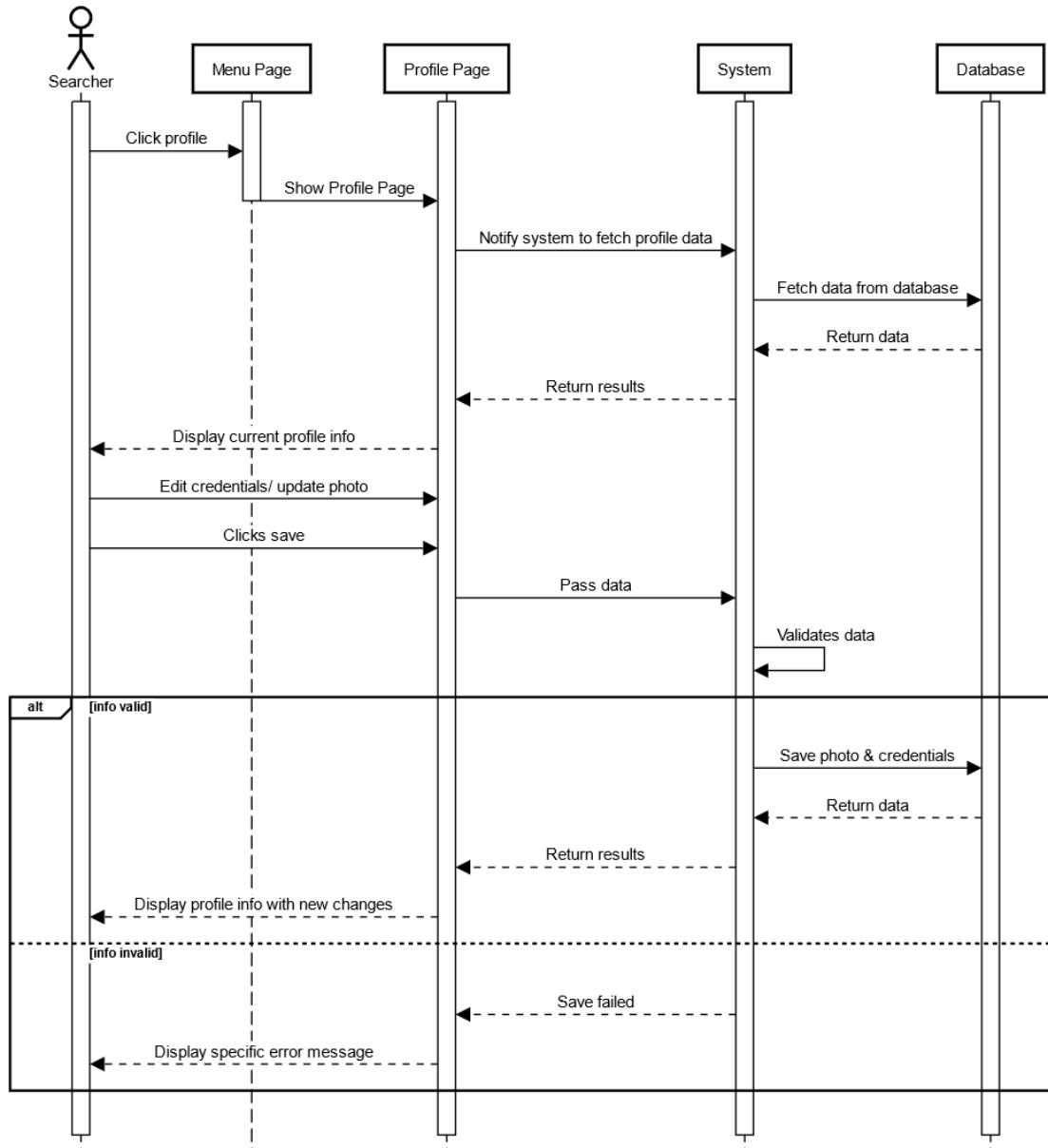




2.5.3 Manage Profile

Use Case Name:	Manage profile
Description:	This use case describes the process by which a searcher can update their profile information within the system. The searcher has the option to update their profile photo, credentials, and other personal details. The system ensures the integrity of the updates through validation before saving any changes to the database.
Primary Actor:	Searcher

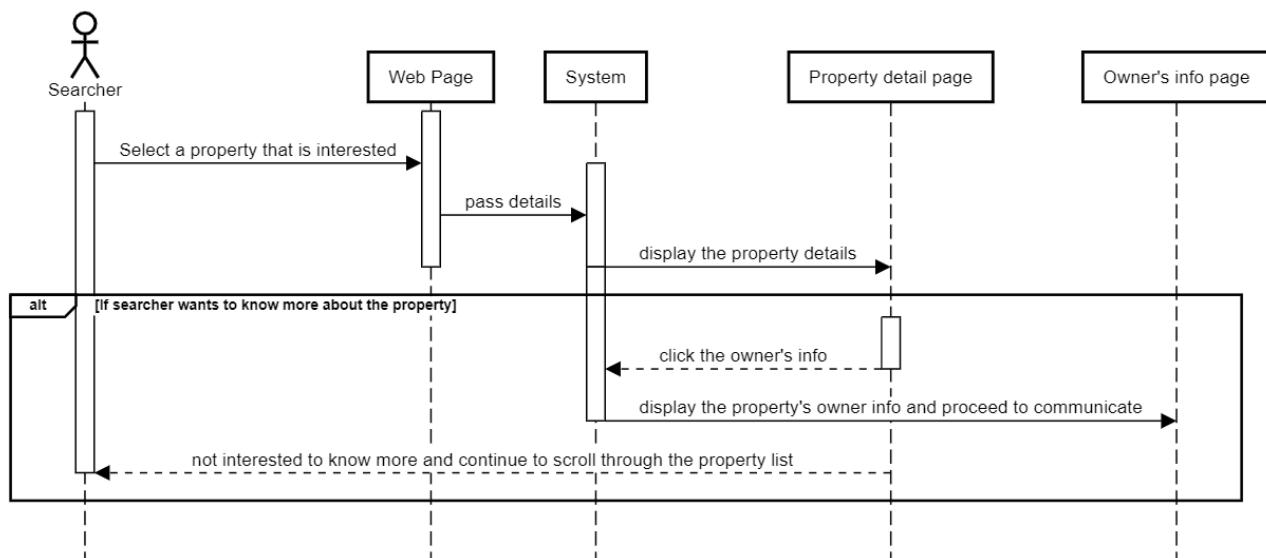
Precondition	The user is logged into their account
Postcondition	The user's profile is updated in the system with new information, if any changes are made and the system's database reflects the latest state of the user's profile data after successful updates.
Main Success Scenario:	<p>1. The user logs into the system and clicks on manage profile.</p> <p>2. The system displays the user's current profile information.</p> <p>3. The user selects the option to update their profile and they can choose to either update their profile photo or credentials.</p> <p>4. If updating the photo:</p> <ul style="list-style-type: none"> ● The user uploads a new photo. ● The system validates the photo format. ● If valid, the system updates the profile photo in the database. <p>5. If removing the photo:</p> <ul style="list-style-type: none"> ● A confirmation window pops out for user to confirm the removal. ● If user clicks confirm, the data gets updated in the database. <p>6. If updating credentials:</p> <ul style="list-style-type: none"> ● The user edits their credentials (username, password, profile picture, email, name, phone number). ● The user clicks 'save changes'. ● The system checks the validity of the new credentials. ● If valid, the system updates the credentials in the database.
Alternative Scenario:	If the user is not logged in, the user cannot manage a profile and will proceed to the login page.



2.5.4 View Property Details

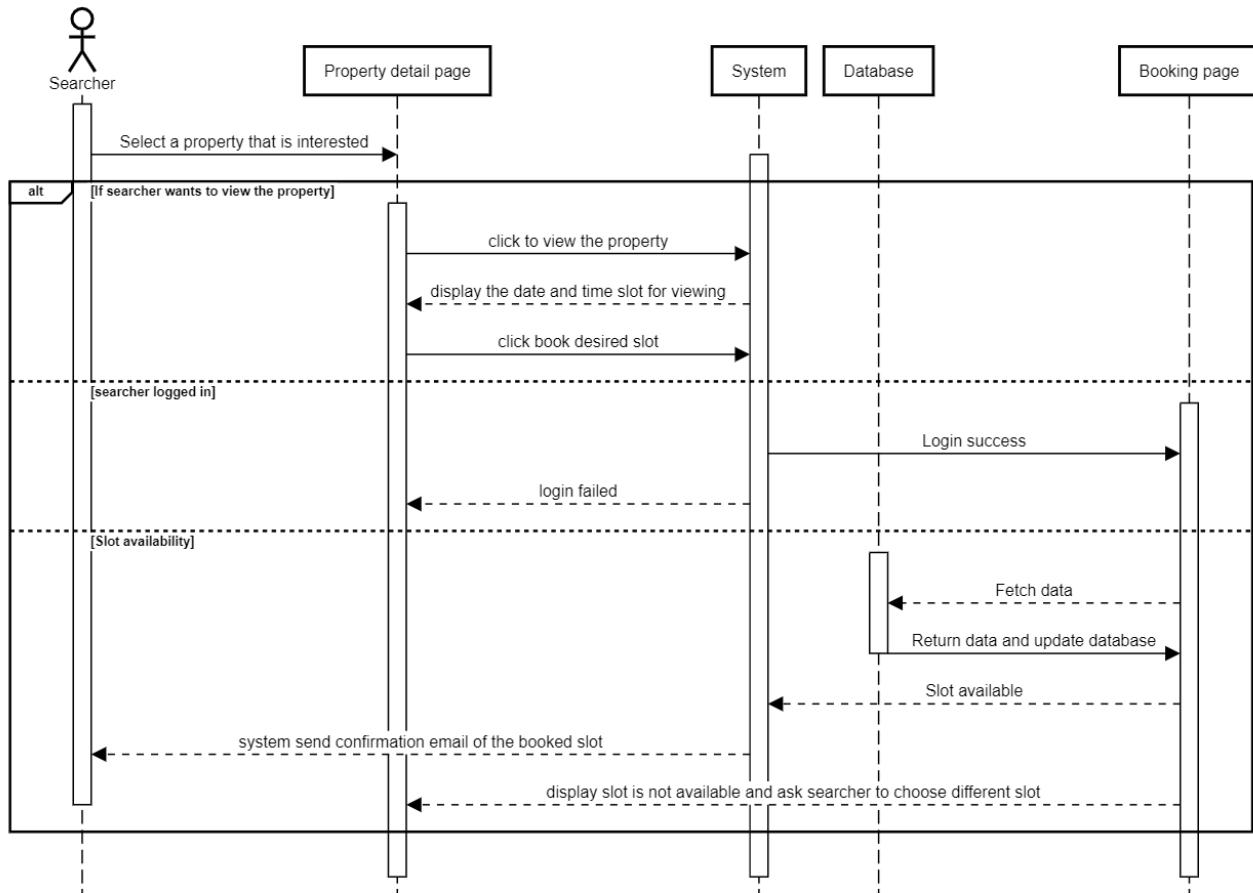
Use Case Name:	View property details
Description:	This use case is for the searcher to view the property details that they are interested in with a more detailed description of the property and display the owner's information to

	communicate if they decided to get to know more about the property.
Primary Actor:	Searcher
Precondition	-
Postcondition	The searcher can view the property details and can proceed with a detailed agreement and more detailed information regarding the property that the searcher is interested in through communication with the property's owner.
Main Success Scenario:	<ol style="list-style-type: none"> 1. The searcher is visiting the website to look for a property option that is available. 2. The searcher can view the property details if they are interested in the property. 3. The system will display the detailed description of the property and also with the owner's contact information. 4. If the searcher wants to know more about the property and have queries, they can communicate with the owner through email or phone. 5. The system will ask the searcher to create an account if they decide to officially become a tenant
Alternative Scenario:	-



2.5.5 Book timing to view Property

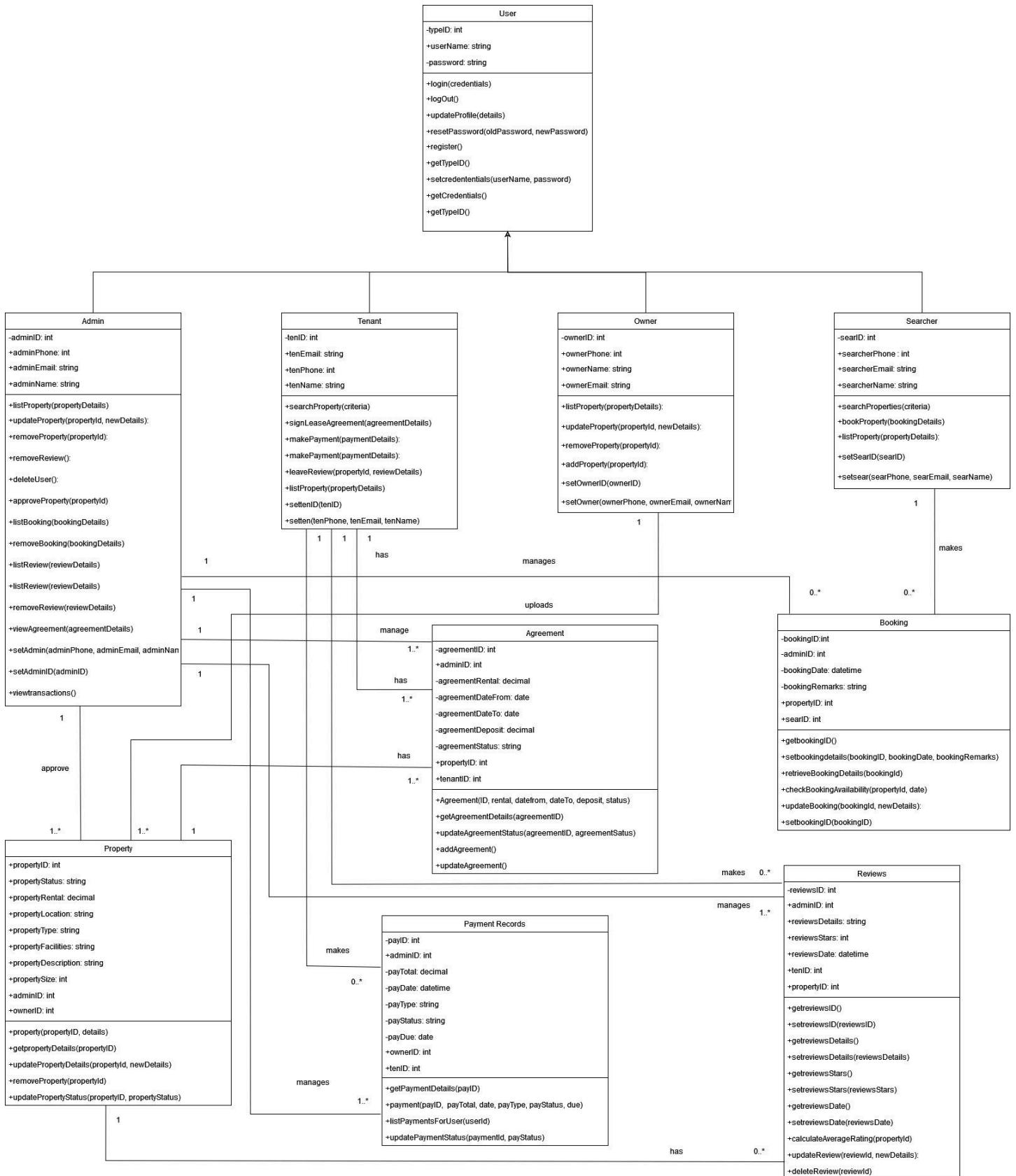
Use Case Name:	Book timing to view property
Description:	This use case is for the process of book timing to view the property by selecting date and time that is available and will receive a confirmation email of the booked slot
Primary Actor:	Searcher
Precondition	Searcher must be log in to make bookings
Postcondition	The confirmation email is sent to both tenant and owner for confirmation of bookings.
Main Success Scenario:	<ol style="list-style-type: none"> 1. The searcher is visiting the website to look for a property option that is available. 2. The searcher can view the property details if they are interested in the property. 3. The system will display the detailed description of the property and also with the owner's contact information. 4. If the searcher wants to view the property, searcher selects date and time slot and inputs any meeting remarks and clicks book. 5. System will check availability of the chosen time slot. If the slot is available, the system will send a confirmation email to both tenant and owner.
Alternative Scenario:	If the slot has been booked by someone, the system will prompt the searcher to choose another time slot.



3 Data Design

3.1 Design Class Diagram

User class is a parent class, and the actors searcher, admin, owner and tenant will inherit from the user class. All inherited classes share the same properties, with additional properties for each actor such as owner can add/update/delete property. The implementation of getters and setters allow encapsulation of data which ensures that the internal representation of an object is hidden from the outside.



3.2 Data Dictionary

3.3 User

User						
Field Name	Data Type	Length	PK/ FK	Required	Null/ Not Null	Description
typeID	integer	15	PK	Yes	Not Null	user type id
UserName	string	30		Yes	Not Null	Username
Password	String	12		Yes	Not Null	Password for user

3.3.1 Admin

Admin						
Field Name	Data Type	Length	PK/ FK	Required	Null/ Not Null	Description
adminID	integer	15	PK	Yes	Not Null	Id for each admin
adminEmail	string	15		Yes	Not Null	Password for login purposes
adminPhone	integer	12		Yes	Not Null	Admin's contact number
adminName	string	30		Yes	Not Null	Admin's full name

3.3.2 Searcher

Searcher						
Field Name	Data Type	Length	PK/ FK	Required	Null/ Not Null	Description
searID	integer	15	PK	Yes	Not Null	Id for each searcher

searEmail	string	30		Yes	Not Null	Searcher's primary email
searPhone	integer	12		Yes	Not Null	Searcher's contact number
searName	string	30		Yes	Not Null	Searcher's full name

3.3.3 Owner

Owner						
Field Name	Data Type	Length	PK/ FK	Required	Null/ Not Null	Description
ownerID	integer	15	PK	Yes	Not Null	Id for each owner
ownerEmail	string	30		Yes	Not Null	Owner's primary email
ownerPhone	integer	12		Yes	Not Null	Owner's contact number
ownerName	string	30		Yes	Not Null	Owner's full name

3.3.4 Tenant

Tenant						
Field Name	Data Type	Length	PK/ FK	Required	Null/ Not Null	Description
tenID	integer	15	PK	Yes	Not Null	Id for each tenant
tenEmail	string	30		Yes	Not Null	Tenant's primary email
tenPhone	integer	12		Yes	Not Null	Tenant's contact number

tenName	string	30		Yes	Not Null	Tenant's full name
---------	--------	----	--	-----	----------	--------------------

3.3.5 Property

Property							
Field Name	Data Type	Length	PK/FK	Required	Null/Not Null	Description	FK Referenced Table
propertyID	integer	15	PK	Yes	Not Null	Id for each property	
ownerID	integer	15	FK	Yes	Not Null	Id for each owner	Owner
adminID	integer	15	FK	Yes	Not Null	Id for each admin	Admin
propertyRental	decimal	4,2		Yes	Not Null	Rental price for the property	
propertyLocation	string	100		Yes	Not Null	Address of where the property located	
propertyType	string	30		Yes	Not Null	Detail on the property type	
propertyFacilities	string	100		Yes	Not Null	Available facility that is offered with the property	
propertyDescription	string	100		Yes	Not Null	Description about the	

						property in detail (furnished etc.)	
propertySize	integer	20		Yes	Not Null	The detail size of the property	
propertyStatus	string	30		Yes	Not Null	The status of the property whether it has been approve/reject by the admin	

3.3.6 Booking

Booking							
Field Name	Data Type	Length	PK/FK	Required	Null/Not Null	Description	FK Referenced Table
bookingID	integer	15	PK	Yes	Not Null	Id for each booking	
propertyID	integer	15	FK	Yes	Not Null	Id for each property	Property
searID	integer	15	FK	Yes	Not Null	Id for each searcher	Searcher
adminID	integer	15	FK	Yes	Not Null	Id for each admin	Admin
bookingDate	Date	15		Yes	Not Null	Format dd/mm/yyyy	
bookingRemarks	String	15		Yes	Not Null	Any special remarks of the	

Booking							
Field Name	Data Type	Length	PK/FK	Required	Null/Not Null	Description	FK Referenced Table
bookingID	integer	15	PK	Yes	Not Null	Id for each booking	
propertyID	integer	15	FK	Yes	Not Null	Id for each property	Property
searID	integer	15	FK	Yes	Not Null	Id for each searcher	Searcher
adminID	integer	15	FK	Yes	Not Null	Id for each admin	Admin
						booking	

3.3.7 Agreement

Agreement							
Field Name	Data Type	Length	PK/FK	Required	Null/Not Null	Description	FK Referenced Table
agreementID	integer	15	PK	Yes	Not Null	Id for each Agreement	
propertyID	integer	15	FK	Yes	Not Null	Id for each property	Property
tenID	integer	15	FK	Yes	Not Null	Id for each tenant	Tenant
adminID	integer	15	FK	Yes	Not Null	Id for each admin	Admin
agreementRental	decimal	4,2		Yes	Not Null	Agreed rental price	

Agreement							
agreementDateFrom	Date	15		Yes	Not Null	Format dd/mm/yyyy start date of agreement	
agreementDateTo	Date	15		Yes	Not Null	Format dd/mm/yyyy end date of agreement	
agreementDeposit	decimal	4,2		Yes	Not Null	The deposit amount	
agreementStatus	String	15		Yes	Not Null	Status to show whether the agreement is accepted/terminated by tenant	

3.3.8 Payment records

Payment Records							
Field Name	Data Type	Length	P K/ F K	Required	Null/ Not Null	Description	FK Referenced Table
PaymentID	integer	15	PK	Yes	Not Null	Id for each payment	
ownerID	integer	15	FK	Yes	Not Null	Id for each owner	Owner
tenID	integer	15	FK	Yes	Not Null	Id for each tenant	Tenant
adminID	integer	15	FK	Yes	Not Null	Id for each admin	Admin

Payment Records							
	er					admin	
PaymentTotal	integer	100		Yes	Not Null	The total amount of each payment	
PaymentDate	Date	15		Yes	Not Null	Format dd/mm/yyyy	
PaymentType	String	15		Yes	Not Null	Type of payment used(Cash,Credit Card)	
PaymentStatus	String	15		Yes	Not Null	Status to show whether the payment is available or not	
PaymentDue	Date	15		Yes	Not Null	Payment due date for deposit	

3.3.9 Reviews

Reviews							
Field Name	Data Type	Length	P K/ F K	Required	Null/ Not Null	Description	FK Referenced Table
reviewsID	integer	15	PK	Yes	Not Null	Id for each payment	
tenID	integer	15	FK	Yes	Not Null	Id for tenant	Tenant
propertyID	integer	15	FK	Yes	Not Null	property Id	Property

Reviews							
adminID	integer	15	FK	Yes	Not Null	Id for each admin	Admin
reviewsDetails	String	100		Yes	Not Null	The detailed review of the property	
reviewsStars	integer	15		Yes	Not Null	how many stars given to the property	
reviewsDate	Date	15		Yes	Not Null	Format dd/mm/yyyy	

3.4 Data Structures

3.4.1 Admin Array

Admin	
Field Name	Data Type
adminID	integer
Username	string
Password	string
adminEmail	string
adminPhone	integer
adminName	string

3.4.2 Searcher Array

Searcher

Field Name	Data Type
searID	integer
Username	string
Password	string
searEmail	string
searrPhone	integer
searName	string

3.4.3 Owner Array

Owner	
Field Name	Data Type
ownerID	integer
Username	string
Password	string
ownerEmail	string
ownerPhone	integer
ownerName	string

3.4.4 Tenant Array

Tenant	
Field Name	Data Type
tenID	integer
Username	string

Password	string
ownerEmail	string
ownerPhone	integer
ownerName	string

3.4.5 Property Array

Property	
Field Name	Data Type
propertyID	integer
propertyRental	decimal
propertyLocation	string
propertyType	string
propertyFacilities	string
propertyDescription	string
propertySize	integer
propertyStatus	string

3.4.6 Booking Array

Booking	
Field Name	Data Type
bookingID	integer
bookingDate	datetime

bookingRemarks	string
----------------	--------

3.4.7 Agreement Array

Agreement	
Field Name	Data Type
agreementID	integer
agreementRental	decimal
agreementDateFrom	datetime
agreementDateTo	datetime
agreementStatus	string
deposit	decimal

3.4.8 Payment Records Array

Payment Records	
Field Name	Data Type
payID	integer
payTotal	decimal
payDate	datetime
payType	string
payStatus	string
payDue	date

3.4.9 Reviews Array

Reviews	
Field Name	Data Type
reviewsID	integer
reviewsDetails	string
reviewsStars	string
reviewsDate	datetime

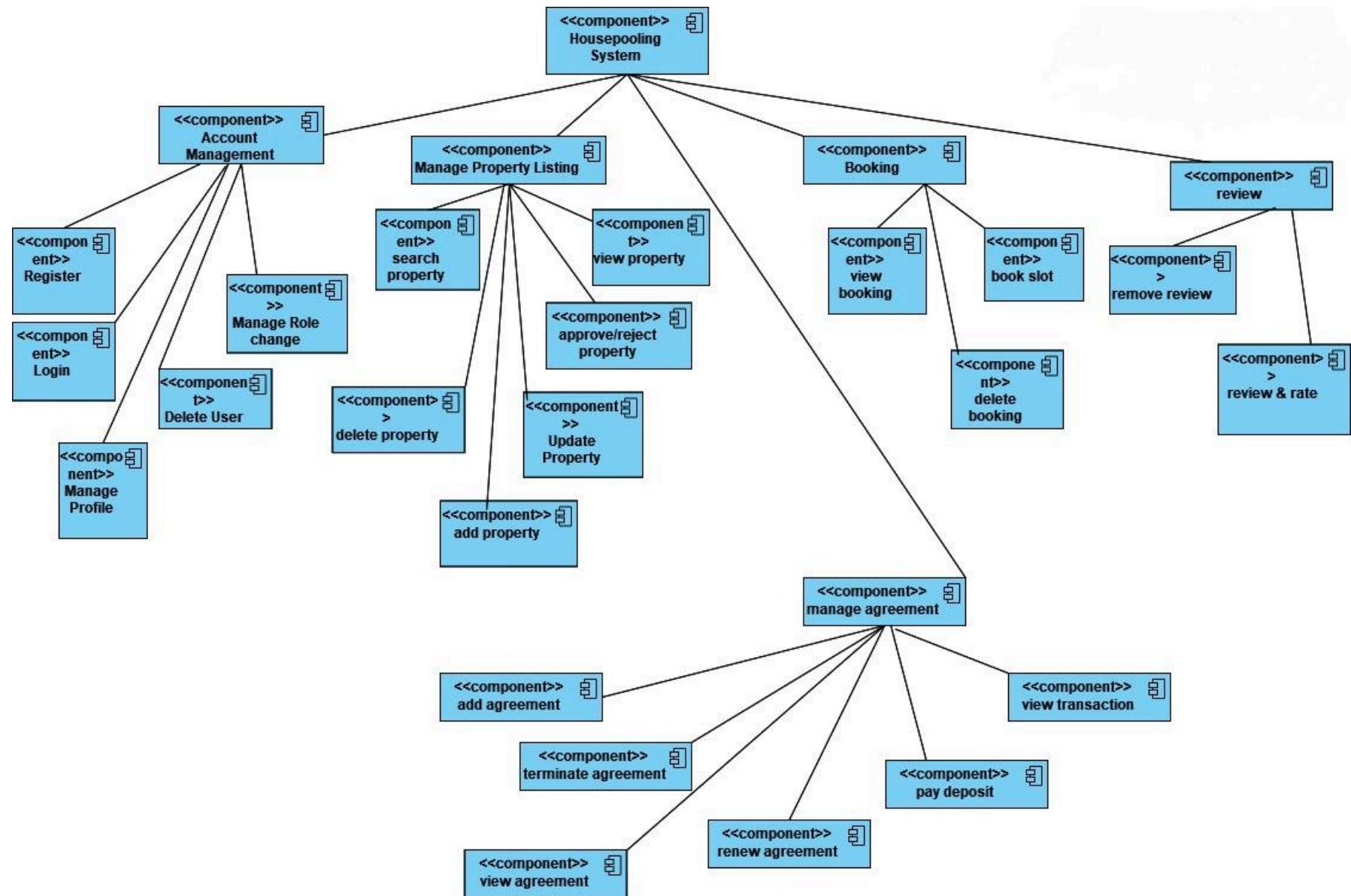
4 Architecture Design

4.1 Software Architecture

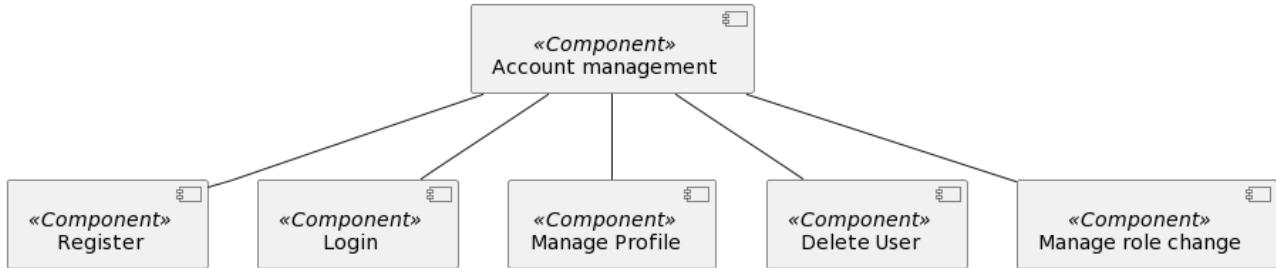
<TO DO: Describe the software architecture and place the architecture diagram here.

<TO DO: Describe the separation of the system into subsystems and how the subsystems are assigned to team members.>

In our housepooling system, there are a total of 5 different subsystems, that includes account management subsystem, manage property listing subsystem, booking subsystem, review subsystem and manage agreement subsystem. For major subsystems like manage property subsystem and manage agreement subsystem, 2 to 3 members could be assigned to it. For example, the member in charge of actor owner could be assigned to handle functionalities such as add property, delete property, and update property. At the same time, another member in charge of actor admin could be in charge of approve/reject property, which also falls under the manage property subsystem. As for manage agreement subsystem, member that is in charge of actor owner can be in charge of add/terminate/renew agreement, while the member in charge of actor tenant can also be assigned to handle pay deposit function based on the amount deposit stated in the agreement. Similarly, the member that is in charge of admin can handle functionalities such as view transaction activity. As for smaller subsystems like review subsystem, 1 member could be assigned to it. For example, member in charge of tenant can be assigned to review subsystem, as most likely the tenant will be able to make reviews on properties. This smooth delegation of task among team members ensures that the overall system achieves accuracy and consistency.

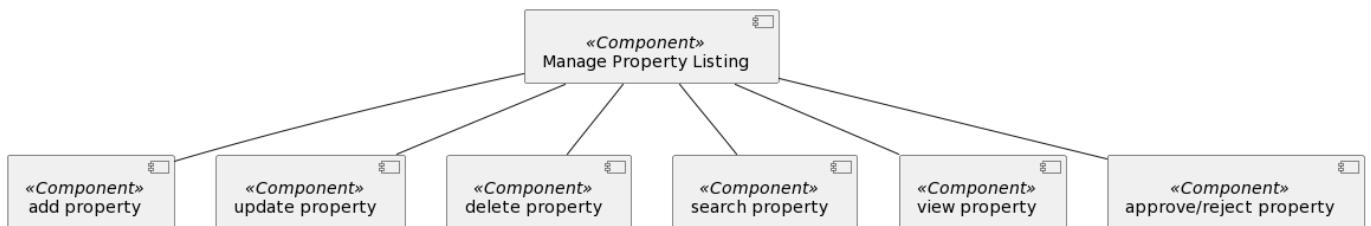


4.1.1 Account management Subsystem



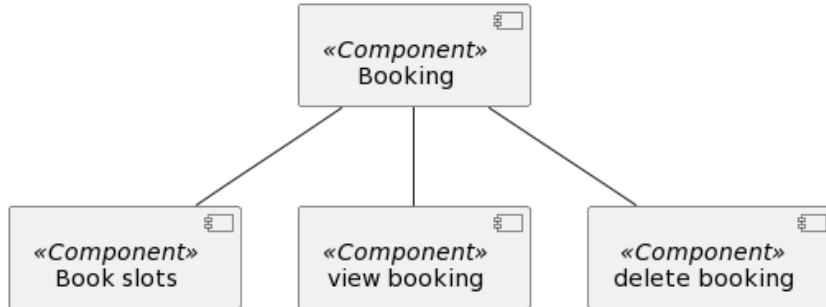
The account management subsystem is responsible to manage user accounts. It consists of registration and login component, where users will register for an account and login. Every user could manage their profile which includes modifying their photo and credentials. Delete user and manage role change is included for admin, where admin can delete user if the user violates the policy and manage role change applications for tenants who want to become owner.

4.1.2 Manage Property Listing Subsystem



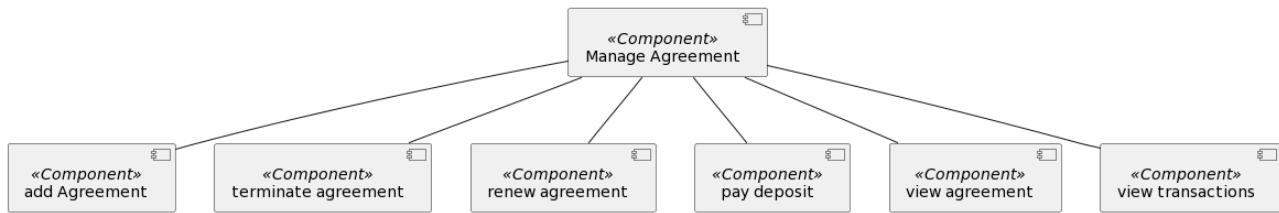
The manage property listing subsystem is responsible for managing property listings. It consist of components such as add, update and delete property for owners. It also has components like search property function for all users to understand the current market value of the property. Approve/reject property component is included for admin, where admin can decide whether to reject/approve property uploaded by admin.

4.1.3 Booking Subsystem



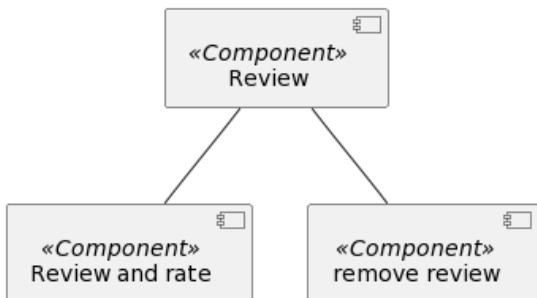
The booking subsystem is responsible for the process of booking slot processes. It consists of component such as book slots where users will choose their desired slot to view the property. There are also component of view booking and delete booking for the users to manage their booked slot.

4.1.4 Manage Agreement Subsystem



The manage agreement subsystem is responsible for managing agreements. It consists of components such as add, terminate, renew agreement for owner, and pay deposit component for tenant. It also includes a view transaction component for the admin to view the activity transaction made by tenant.

4.1.5 Review and rating Subsystem

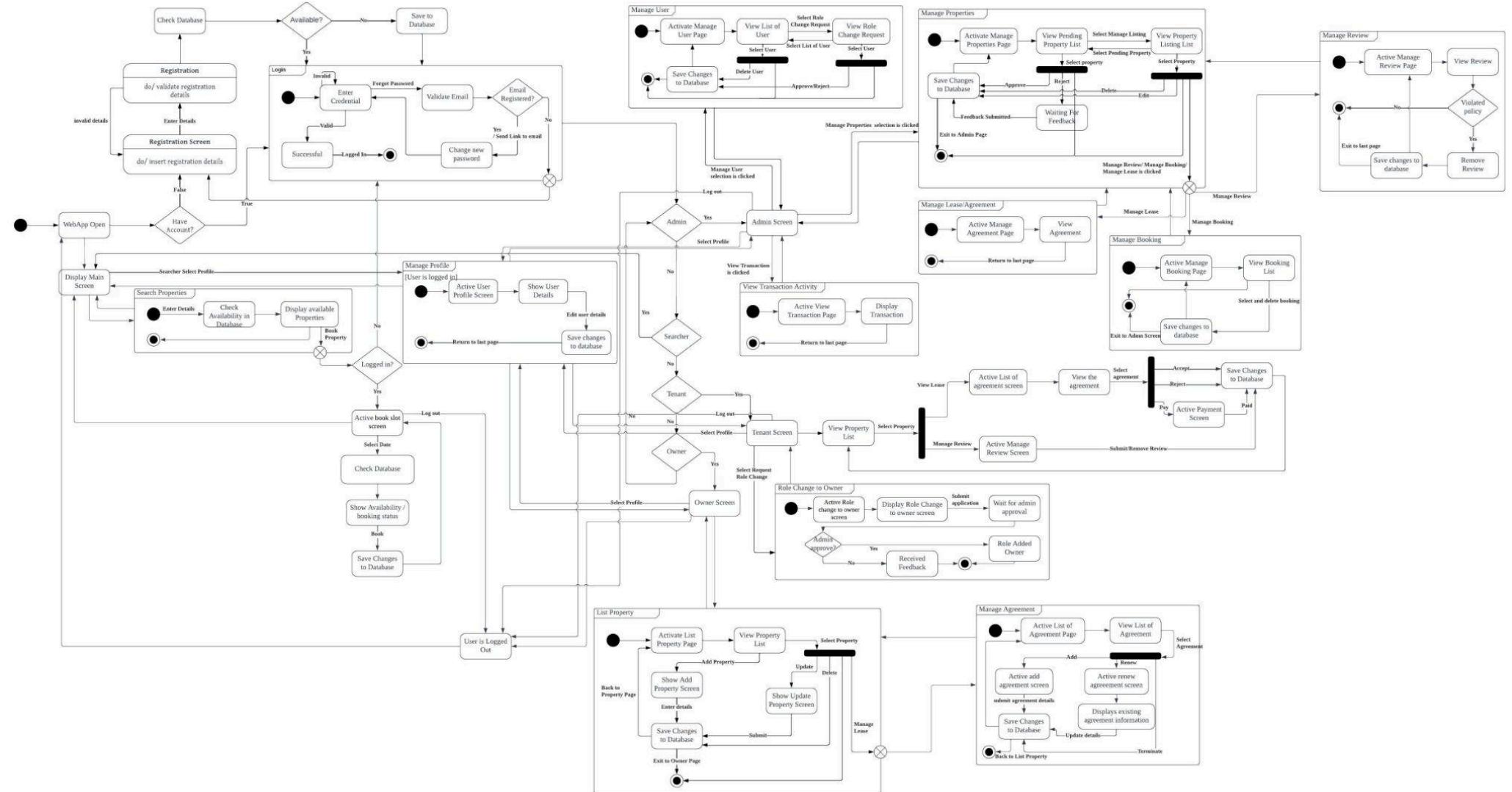


The review and rating subsystem is responsible for managing reviews. It consist of components such as review and rate and remove review, which will be performed by the tenant

4.2 State Transition Diagram

State diagram PDF format: https://drive.google.com/file/d/1ng7uC8jmoWYBd9vEZC0qOrzYsf_yKx76/view?usp=sharing

State diagram JPEG format: https://drive.google.com/file/d/1aj3HShnYsv6Vsa_xInRvqEulcCkWkhIR/view?usp=sharing



5 Interface Design

5.1 Index Screen / Menu Page (Main Screen)

This is the first page that is shown to the user. The searcher can perform a search for properties even without log in. The page also displays top-rated properties for the searcher to view. Owner, tenant and admin can also perform search for properties to know properties available /trending in the marketplace.

The screenshot shows a user interface for searching properties. At the top, there is a navigation bar with links for ".logo", "About Us", "Login", and "Sign Up". Below the navigation bar is a yellow header area containing the title "Search for Properties". The main content area is a white box containing a search form. The search form includes a text input field labeled "Search property...", a dropdown menu labeled "Monthly rental from", a dropdown menu labeled "Monthly rental to", a dropdown menu labeled "Location", a dropdown menu labeled "Property Type", and a black "Search" button at the bottom.

.logo About Us Login Sign Up

Top Rated Properties

Ratings: ★★★★★
Property Title
brief description
View

Ratings: ★★★★★
Property Title
brief description
View

Ratings: ★★★★★
Property Title
brief description
View

footer

5.2 Account Management Subsystem

The component includes: User registration, login, profile management, manage user, manage role change request.

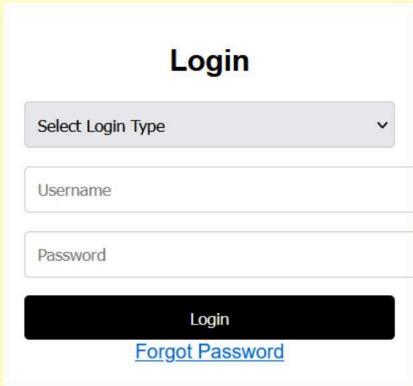
(login screen)
-this login screen is shared by all users.

.logo

About Us

Login

Sign Up



The login screen features a central white rectangular form with a black header containing the word "Login". Above the form is a dropdown menu labeled "Select Login Type". Below the form are two input fields: "Username" and "Password", each with a placeholder text ("Username" and "Password" respectively). At the bottom of the form is a black rectangular button labeled "Login" and a blue link labeled "Forgot Password". The entire form is set against a light yellow background.

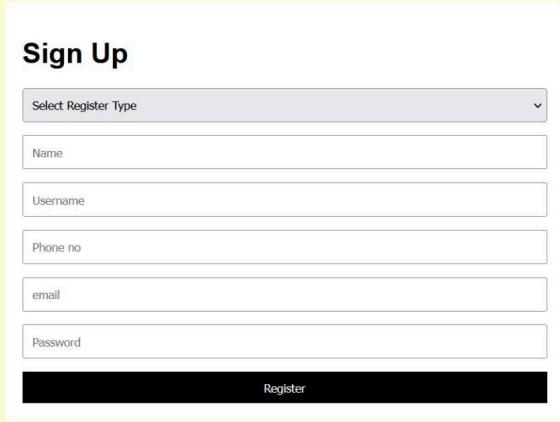
(Register screen)
-shared by all users

.logo

About Us

Login

Sign Up



The sign up screen features a central white rectangular form with a black header containing the word "Sign Up". Above the form is a dropdown menu labeled "Select Register Type". Below the form are five input fields: "Name", "Username", "Phone no", "email", and "Password", each with a placeholder text ("Name", "Username", "Phone no", "email", and "Password" respectively). At the bottom of the form is a black rectangular button labeled "Register". The entire form is set against a light yellow background.

(User profile screen)
-shared by all users

Home **Properties** **Profile** **Log Out**

Manage profile



Name:	<input type="text" value="James"/>
Username:	<input type="text" value="jame123"/>
Phone no:	<input type="text" value="017-1234567"/>
Email:	<input type="text" value="james@gmail.com"/>
Password:	<input type="password" value="***"/>

Save

(Admin manage user screen)
-admin can delete account for each user.

[Home](#)[Manage Transactions](#)[Manage User](#)[Profile](#)[Log Out](#)**Manage User****Role Change Request**

Select	Username	Usertype	Email	phone num	name
<input type="radio"/>	jane1234	owner	jne@gmail.com	012-3456789	Jane

[Delete account](#)

(Tenant role change to owner screen)

-If tenant wants to become a owner, owner clicks on request for role change. Tenant downloads the form and fills in. Tenant has to fill in personal information, prove of property, and reasons for role change. Once submitted, tenants wait for application to be approved by admin.

[Home](#)[Properties](#)[Profile](#)[Log Out](#)**Become An Owner**[Click here to download form](#) Browse... No file selected. Submit

(admin manage role change screen)

-admin can view the list of role change. Admin downloads the agreement uploaded by tenant. Admin verify the details. They have the options to approve or reject, they can reject based on factors including late payment, active lease status, etc.

[Home](#) [Manage Transactions](#) [Manage User](#) [Profile](#) [Log Out](#)

List of Role Change

Select	Username	Payment Status	lease status	User email	Phone num	docs
<input type="radio"/>	leee1234	late	terminated	l@gmail.com	013-2345678	 Click to download
<input type="radio"/>	rty345	on time	pending	q@mgmai.l.com	012-2345678	 Click to download

Approve Reject

5.3 Manage Property Listing Subsystem

<TO DO: Describe the screens of subsystem 2 and place the screen designs here.>

The component includes: Property creation, update property, delete property, search property functionality, view list of properties, approve/reject property.

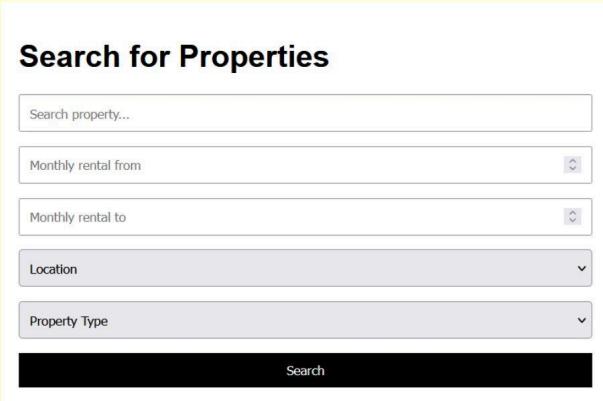
(search screen)

.logo

About Us

Login

Sign Up



The image shows a search interface titled "Search for Properties". It features a search bar labeled "Search property...". Below it are two dropdown menus: "Monthly rental from" and "Monthly rental to", both with downward arrows. There are also two dropdown menus: "Location" and "Property Type", each with a downward arrow. At the bottom is a large black rectangular button with the word "Search" in white.

(owner view property screen)

-After login, owner can view the list of properties owned. Owner has the options to add, delete, or update the property. Owner can also manage agreement for each property by ticking the property and click manage lease

Home **Properties** **Profile** **Log Out**

Welcome Back! Name

Property List

Select	Pic	Property Name	Property Address	Status	feedback
<input type="radio"/>		Property Name	kampung Mawar, Jalan Lengkok, Kedah	rejected	wrong address
<input type="radio"/>		Property name	kampung Mawar, Jalan Lengkok, Kedah	approved	-

Add **Manage lease** **Update** **Delete**

(owner add property screen)

-owner enter details of the property, uploads pic and submit

Home **Properties** **Profile** **Log Out**

Add Property

Location

Address

Rental

Description

Size (in sqft)

Furnished

Property Type

Facilities

Bedrooms

Bathrooms

No file selected.

(owner update property screen)

-owner update property details. The screen displays the existing information for property, owner click on textbox to edit and click submit.

Home

Properties

Profile

Log Out

The screenshot shows a web-based application interface. At the top, there is a navigation bar with four items: "Home", "Properties", "Profile", and "Log Out". Below the navigation bar, the main content area has a title "Update Property". Inside this title, there is a dropdown menu labeled "Location" which is currently set to "123 Main St". There are several input fields: one for the number "1500", one for the description "Spacious 3-bedroom house", one for the square footage "1800 sq. ft.", one for the type "House", one for the amenities "Swimming pool, garage", and two for the number of bedrooms "3" and "2". Below these fields is a file upload input labeled "Browse..." followed by a file path "360_F_352117727_d5h8y15mn7mxzYkte15ThuDlHzRuGkN.jpg". At the bottom of the form is a single button labeled "Submit".

(Admin pending property screen)

-after login, it display the pending property screen. Admin ticks on the property and click approve/reject.

[Home](#)[Manage Transaction](#)[Manage User](#)[Profile](#)[Log Out](#)

Welcome Back! Name

[Manage Listing](#)

Pending Property List

Select	Pic	Property Name	Property Description	Owner name	Owner email
<input type="radio"/>		Property name	Property description	Lam	lam@gmail.com
<input type="radio"/>		Property Name	Property description	Lee	lee@gmail.com

[Approve](#)[Reject](#)*(Admin feedback after reject screen)**-if admin rejects, admin can enter feedback based on the reasons for rejection, example, invalid pic, blurred pic, invalid address, etc.*

[Home](#) [Manage Transaction](#) [Manage User](#) [Profile](#) [Log Out](#)

Status: rejected

Feedback:

Add your comment...

Submit

(Admin view & manage listing screen)

-admin can manage the property listing uploaded by owner. Admin can select the property, and has the options to manage lease, manage review, edit/ delete property.

[Home](#) [Manage Transaction](#) [Manage User](#) [Profile](#) [Log Out](#)

Manage Listing

Select	Pic	Property Name	Property Description	Owner name	Owner email
<input type="radio"/>		Property name	Property description	Lam	lam@gmail.com
<input type="radio"/>		Property Name	Property description	Lee	lee@gmail.com

Manage Lease

Manage Review

Manage Booking

Edit

Delete

(Tenant view property screen)

-After login, the screen displays the properties rented by the tenant. Tenant has the options to view the lease uploaded by owner by selecting the property. Tenant can also manage reviews for each property.

Home
Properties
Profile
Log Out

Welcome Back! Name
[Request for role change to owner](#)

Property List

Select	Pic	Property Name	Property Address
<input type="radio"/>		Property name	kampung Mawar, Jalan Lengkok, Kuala Lumpur
<input type="radio"/>		Property Name	kampung Mawar, Jalan Lengkok, Kuala Kuda

[View Lease](#)
[Manage Review](#)

5.4 Booking Subsystem

Components: Searcher book slots to view property, manage booking.

(Searcher book slot screen)

-After searcher perform the search for properties, if they are interested in the property, they can select the booking slot. Before they book the timing to view property, the searcher has to be logged in.

Home **Profile** **Log Out**



Property Title
description in detail

Select Booking slot:

<<
<
July 2019
>
>>

Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Remarks

Book

(admin manage booking screen)
 -admin can view the bookings booked by searcher.

Home **Manage Transactions** **Manage User** **Profile** **Log Out**

Manage Booking

Select	Searcher Name	Booking Date	Remarks
<input type="radio"/>	Onn	08/05/2006 03:05:15 PM	Meet me near the park
<input type="radio"/>	Lily	07/05/2006 03:05:15 PM	-

Delete

5.5 Manage Agreement Subsystem

Component includes: view list of agreements, add agreement, renew agreement, terminate agreement, pay deposit, view transactions.

(owner list of agreements screen)

-after selecting the property, owner can see the list of agreements for the selected property. Owner selects the agreement and has the options to renew, terminate or add.

List of Agreements									
Select	Tenant Name	Tenant no.	Rental	Start	End	deposit	Status	Payment Status	
<input type="radio"/>	leong	016-494 5678	467	01/31/202 5	01/31/202 6	500	pending	unpaid	
<input type="radio"/>	lee	013-234 5678	800	01/31/202 3	01/31/202 5	1000	accepted	paid	
Renew				Terminate				Add	

(Owner add agreement screen)

-owner input details for agreement

Home Properties Profile Log Out

Add Agreement

Tenant Name:

Rental:

Agreement Start date:

Agreement End date:

Deposit:

(Owner renew agreement screen)

-owner renews agreement. The screen displays existing agreement information. Owner clicks on textbox field to update.

Home Properties Profile Log Out

Renew Agreement

Tenant Name:

Rental:

Agreement Start date:

Agreement End date:

Deposit:

Payment Status: Paid

Current Lease Status: Accepted

(tenant list of agreement screen)

-tenant view the agreement. Tenant can select the agreement, and click accept or reject. They can also select agreement and click pay to pay for deposit stated in the agreement.

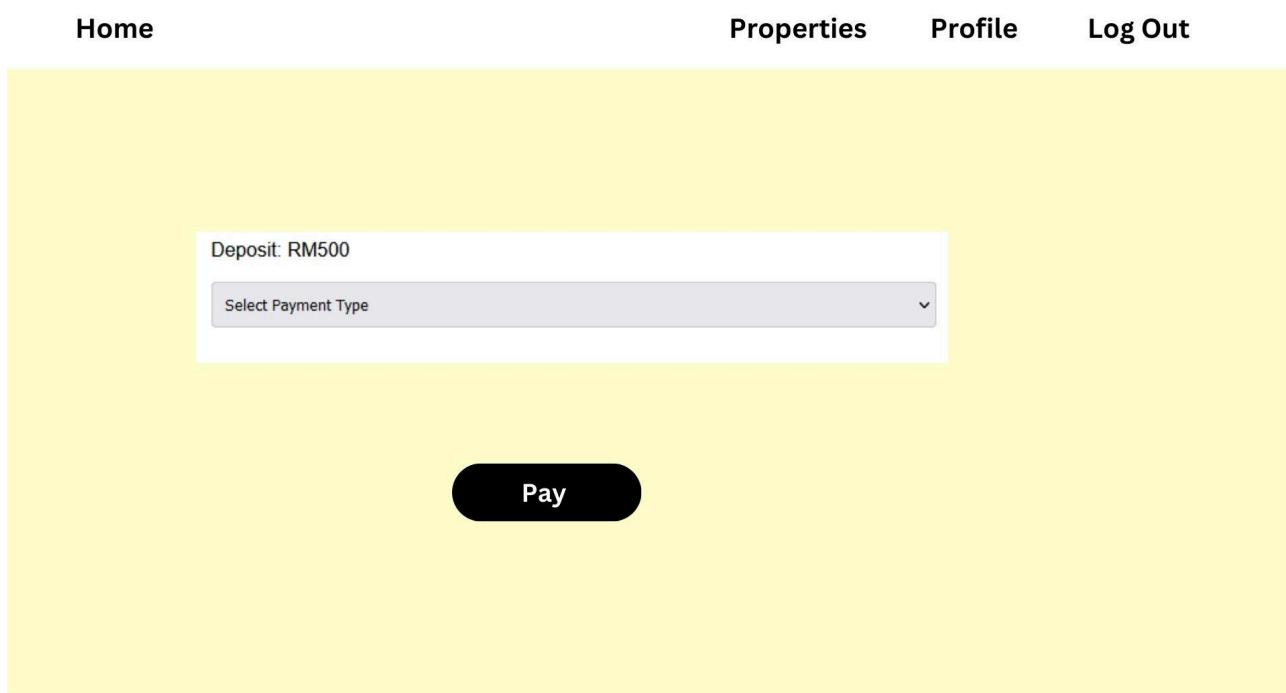
The screenshot shows a user interface for managing agreements. At the top, there is a navigation bar with links for Home, Properties, Profile, and Log Out. Below this is a section titled "List of Agreements". A table displays a single row of data:

Select	Tenant Name	Tenant no.	Rental	Start	End	deposit	Status	Payment Status	Payment Due
<input type="radio"/>	leong	016-4945678	467	01/31/2025	01/31/2026	500	pending	unpaid	02/12/2024

Below the table are three buttons: "Accept", "Reject", and "Pay".

(Tenant pay deposit based on deposit in agreement screen)

-tenant selects the payment type, example ewallet, online banking and click pay.



(admin view lease/agreement screen)

-admin can view the agreement and can filter agreement based on status like active, terminated and pending agreements between tenant and owner.

The screenshot shows an admin interface for managing leases. At the top, there are navigation links: Home, Manage Transaction, Manage User, Profile, and Log Out. Below this, a yellow header bar contains the text "Manage Lease". To the right of the header is a dropdown menu with the following options: Active, Active, Terminated, Pending. Below the header is a table with the following data:

Tenant Name	Owner Name	Agreement Date From	Agreement Date to	lease status
Leong	Lim	01/31/2025	01/31/2026	accepted
Kim	Jay	01/1/2023	01/2/2023	accepted

(admin view transaction activity screen)

-admin can view the transaction activity made by tenant. In this way, admin can easily track the payment if there are any issues in payment made.

Home	Manage Transactions	Manage User	Profile	Log Out																								
View Transactions																												
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Tenant Name</th><th>Tenant email</th><th>Owner Name</th><th>Owner email</th><th>property name</th><th>amount</th><th>pay date</th><th>pay type</th></tr> </thead> <tbody> <tr> <td>Leong</td><td>l@gmail.com</td><td>Esher</td><td>d@gmail.com</td><td>property name</td><td>1000</td><td>08/05/20 06 03:05:15 PM</td><td>ewallet</td></tr> <tr> <td>Kim</td><td>q@mgmail.com</td><td>John</td><td>e@gmail.com</td><td>property name</td><td>400</td><td>08/05/20 06 03:05:15 PM</td><td>online banking</td></tr> </tbody> </table>					Tenant Name	Tenant email	Owner Name	Owner email	property name	amount	pay date	pay type	Leong	l@gmail.com	Esher	d@gmail.com	property name	1000	08/05/20 06 03:05:15 PM	ewallet	Kim	q@mgmail.com	John	e@gmail.com	property name	400	08/05/20 06 03:05:15 PM	online banking
Tenant Name	Tenant email	Owner Name	Owner email	property name	amount	pay date	pay type																					
Leong	l@gmail.com	Esher	d@gmail.com	property name	1000	08/05/20 06 03:05:15 PM	ewallet																					
Kim	q@mgmail.com	John	e@gmail.com	property name	400	08/05/20 06 03:05:15 PM	online banking																					

5.6 Review and Rating Subsystem

Component includes: review and rate property, remove review, manage review.

(Tenant rate and review property screen)

-tenant review property by entering number of stars and comments. They can choose to remove the comments for each property.

Home

Properties

Profile

Log Out

Number of stars: 

Comments:

Add your comment...

Submit

Remove

(admin manage review screen)

-admin can view the reviews uploaded by tenants. They can remove the review if the review has violated the policy/ terms and conditions.

Home **Manage Transaction** **Manage User** **Profile** **Log Out**

Manage Review

Select	Tenant Name	Review Stars	Review details
<input type="radio"/>	leong	5	Nice house!
<input type="radio"/>	James	4	good!

Delete

6 Component Design

6.1 Main Components

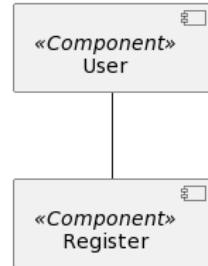
<TO DO: Describe the main components (modules, classes, packages, etc.) and the table with the components and related subsystems here.>

Subsystem	Component
Account Management Subsystem	-register -login -manage profile -manage role change -delete user
Booking subsystem	-view booking -book slot -delete booking
Review Subsystem	-remove review -review and rate

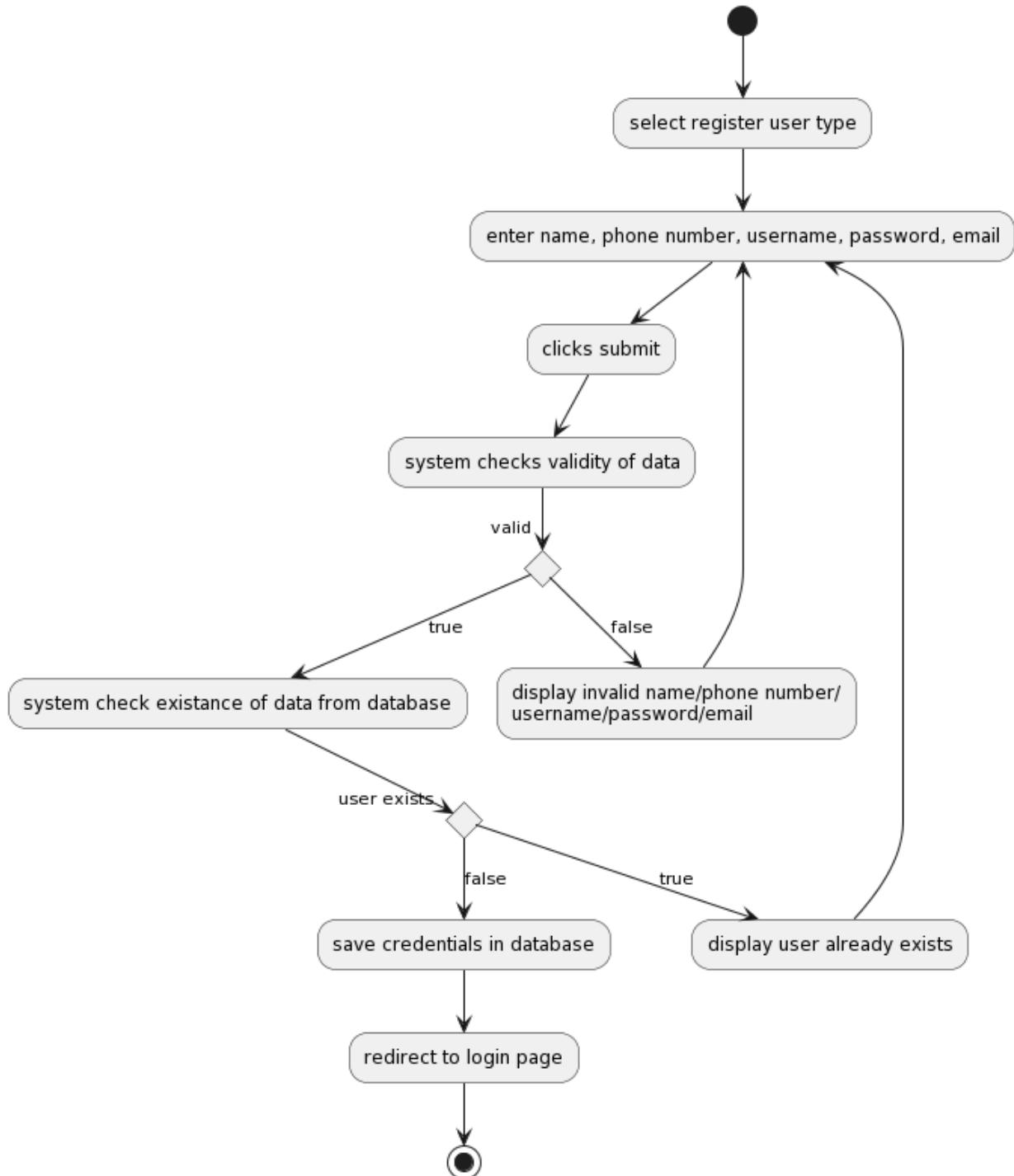
Manage Property Subsystem	-search property -view property -add property -approve/reject property -delete property -update property
Manage Agreement Subsystem	-add agreement -terminate agreement -view agreement -renew agreement -pay deposit -view transaction

6.1.1 Registration (all users)

The user navigates to the registration page and selects the type of account they wish to create (owner, tenant, searcher, or admin). After successful registration, the user will be redirected to the login page.



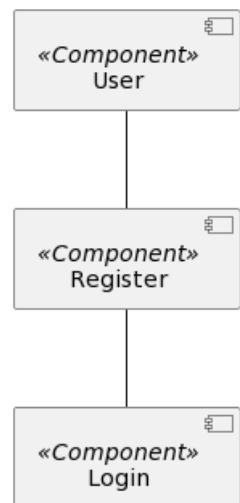
(Registration activity diagram)



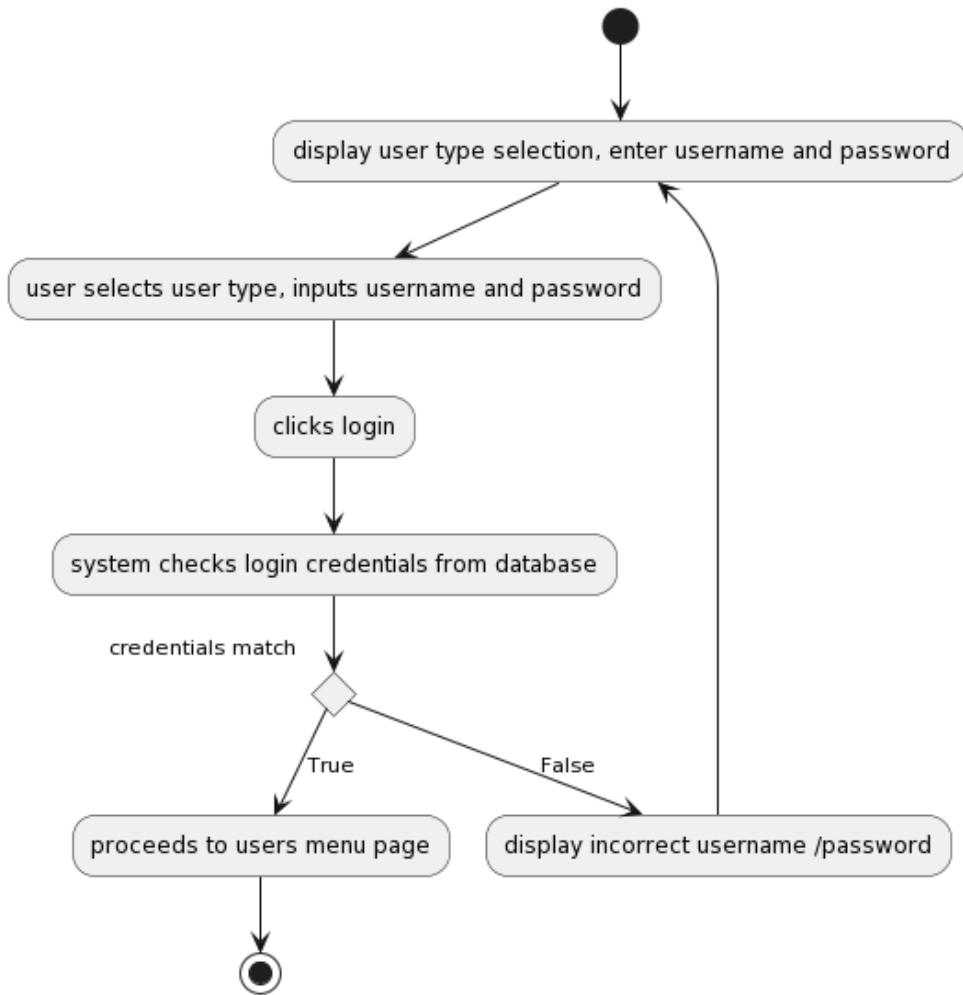
6.1.2 Login (all users)

After the user created an account via registration, it will navigate to login. Login function is able to check for user input credentials. If the credentials are correct, the system allows access. If the

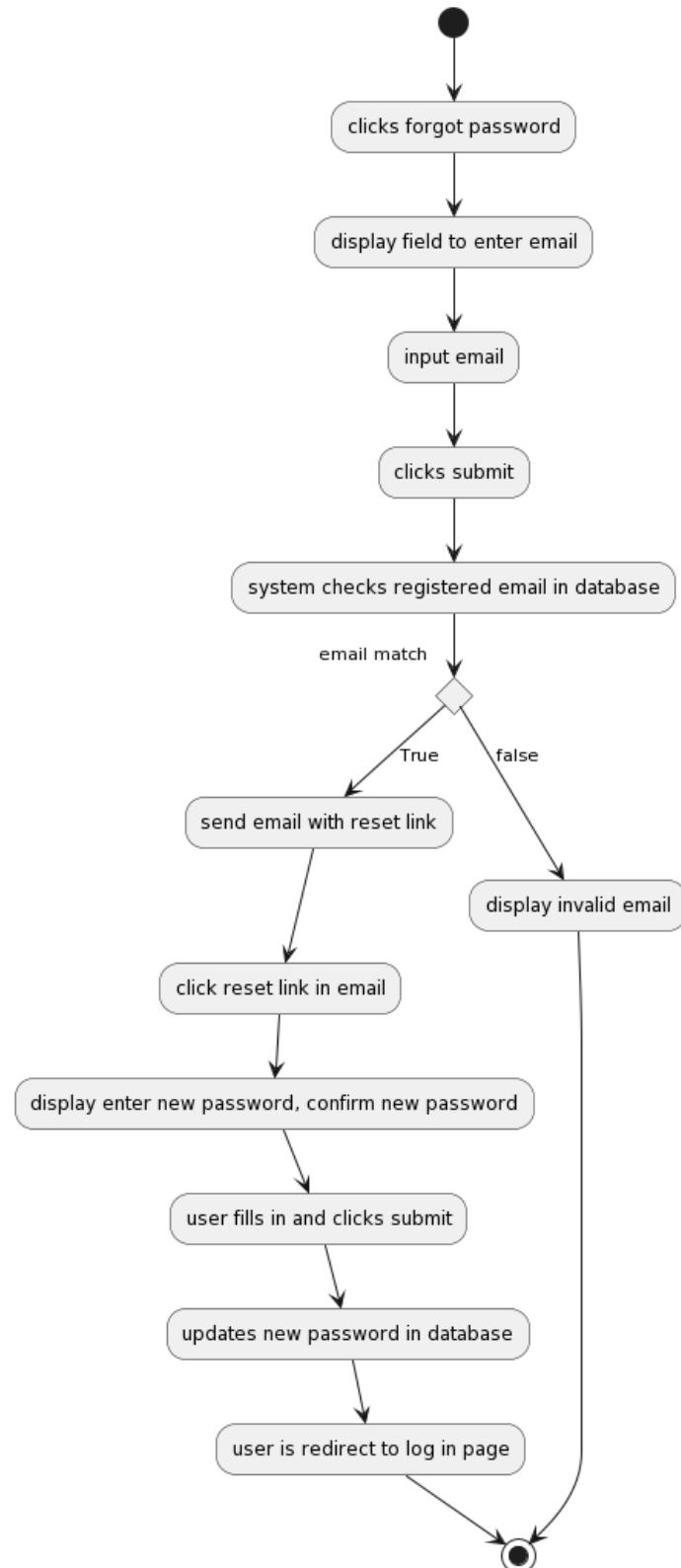
credentials are incorrect, the system generates an error message which is displayed to the user, prompting them to try again. The user can click forgot password if they wish to reset the password, which sends them a reset link to their email.



(login activity diagram)

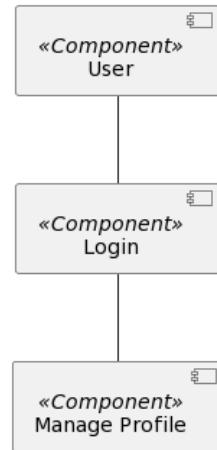


(forgot password activity diagram)

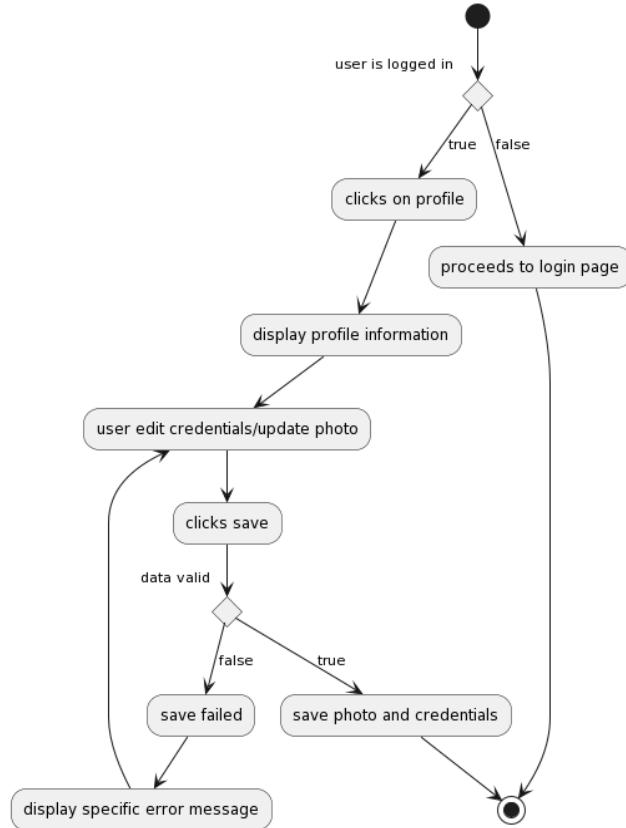


6.1.3 Manage Profile (all users)

<TO DO: Describe the component and place the diagram here. There should be algorithm, pseudocode, flowchart, activity diagram to support the processing in the component.>
The user proceeds to login first, then navigate to profile section to change their credentials.



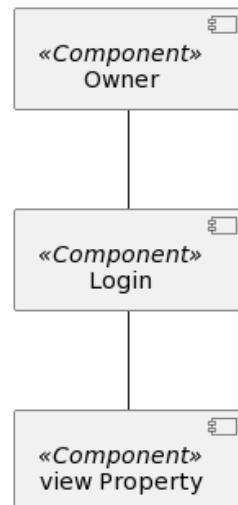
(manage profile activity diagram)



Owner Component Diagram:

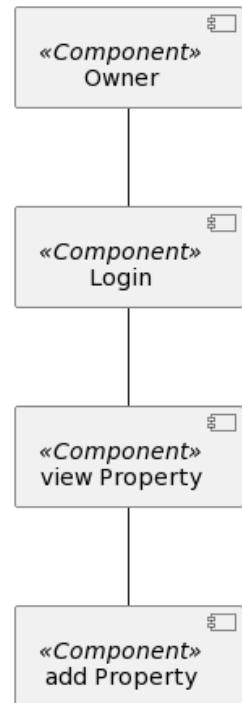
6.1.4 Owner view Property

The owner proceeds to login with the registered account, then heads over to the property action to view property details.



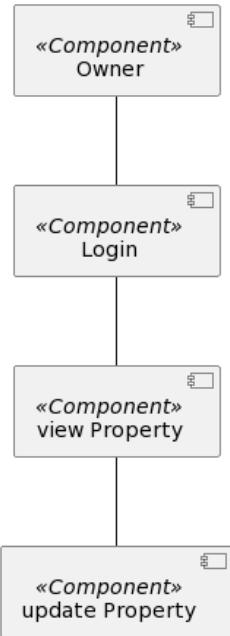
6.1.5 Owner Add Property

The owner proceeds to login with the registered account, then heads over to the property action to add property.



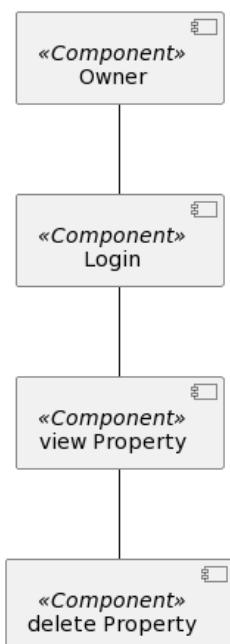
6.1.6 Owner Update Property

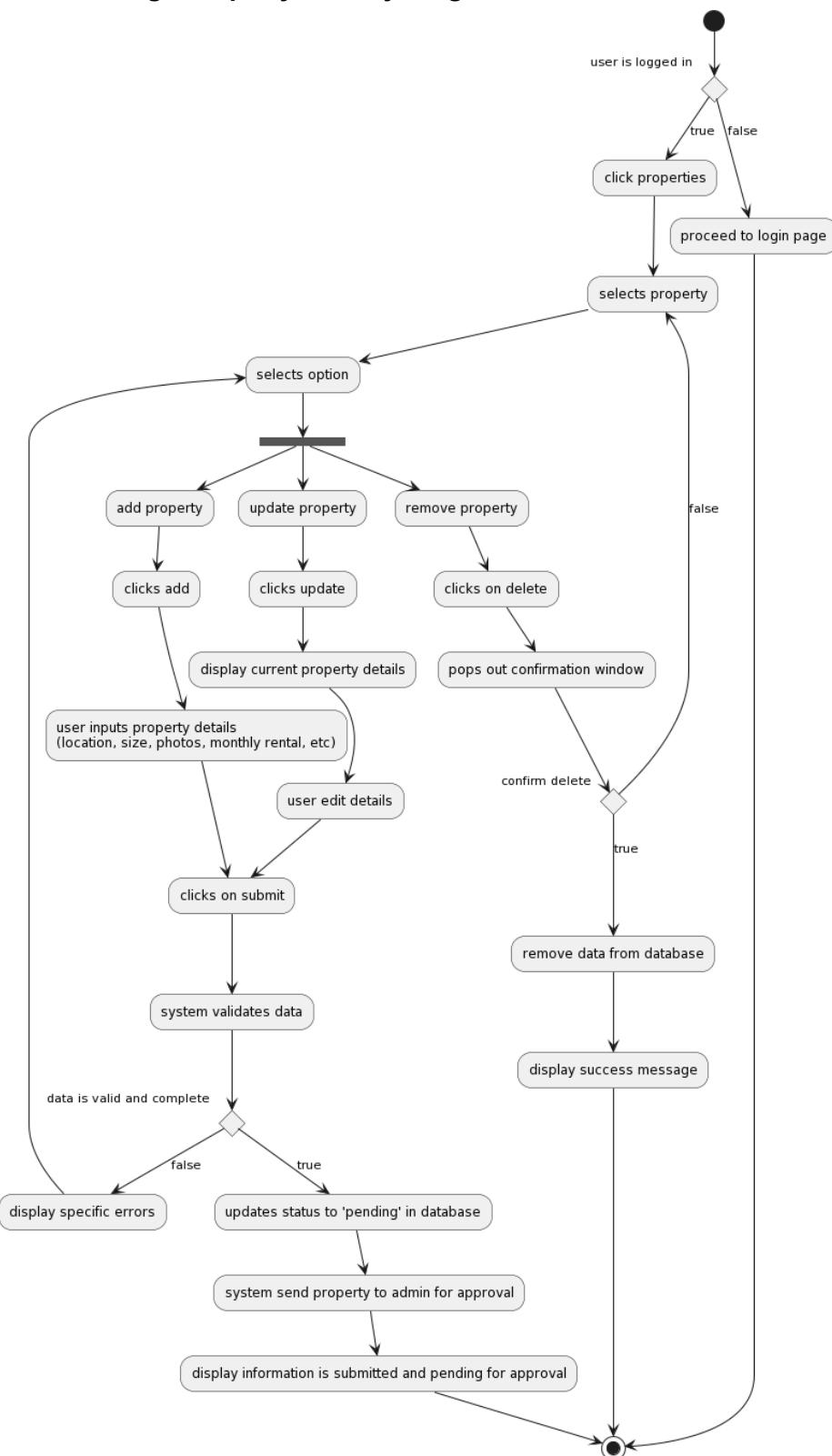
The owner proceeds to login with the registered account, then heads over to the property action to update property.



6.1.7 Owner Delete Property

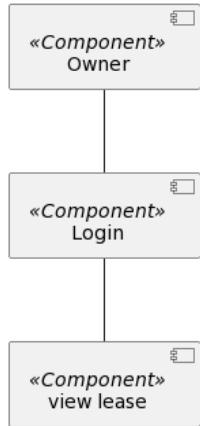
Owner can choose to delete the property at the property section.



Owner Manage Property Activity Diagram:

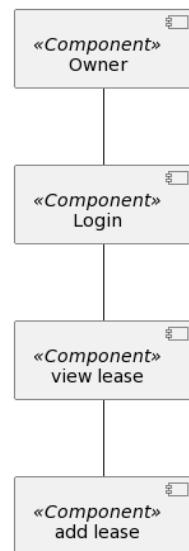
6.1.8 Owner View lease

After login, owner can navigates to the agreement section to view lease.



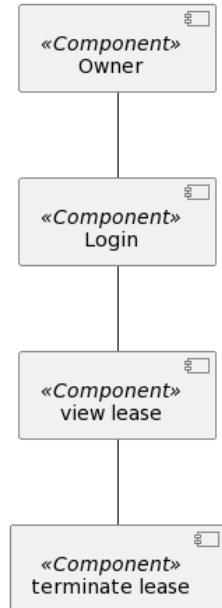
6.1.9 Owner Add Lease

Owner can chose to add lease for new tenants.



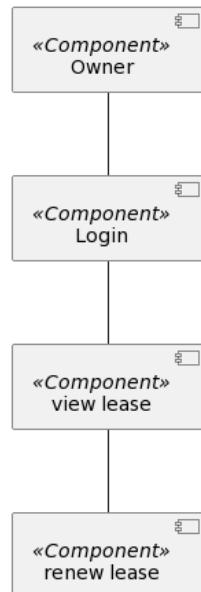
6.1.10 Owner Terminate Lease

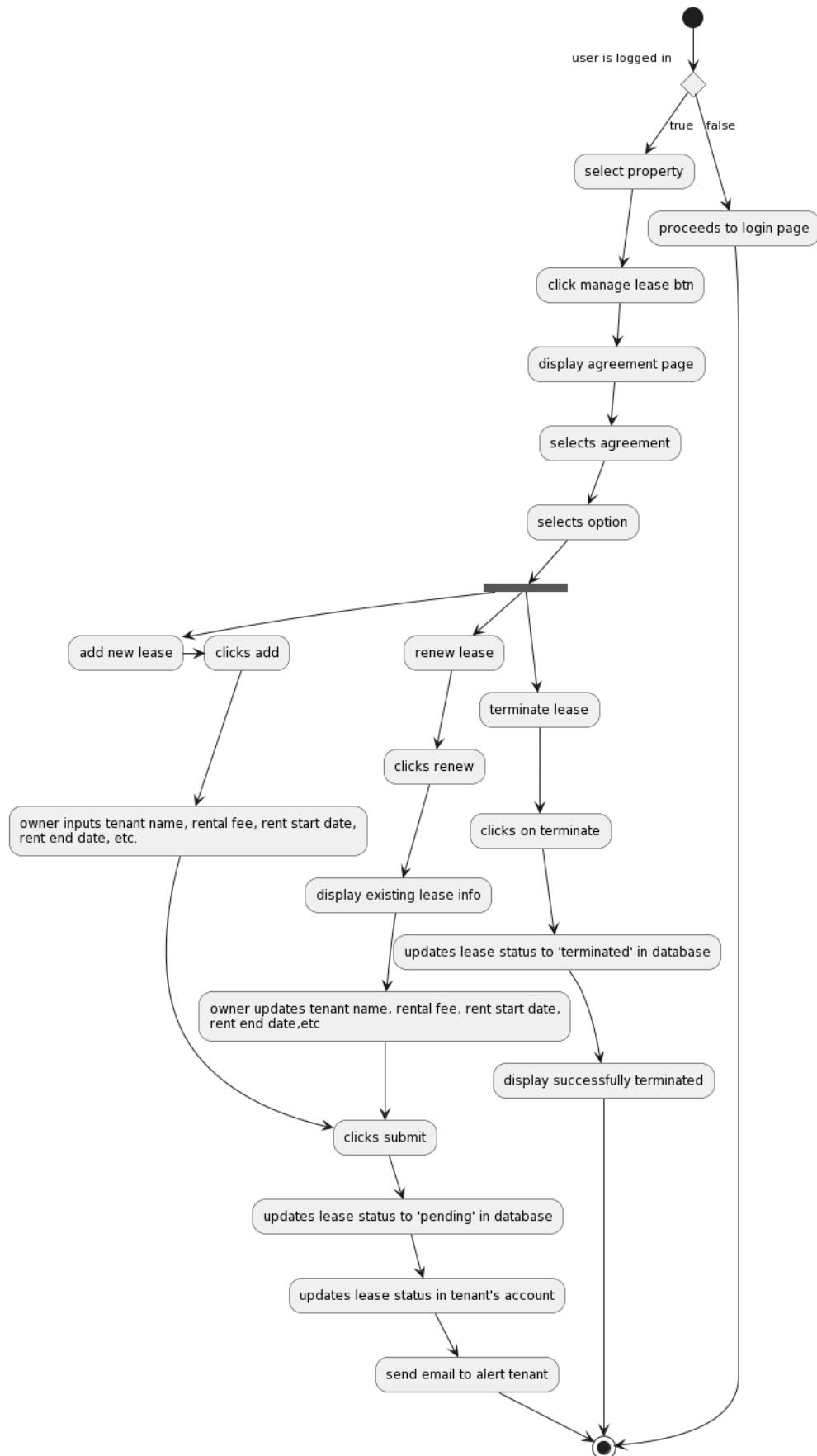
If owner wants to terminate lease, owners terminates the lease, where tenant will also be notified about the terminated status.



6.1.11 Owner Renew Lease

If owner wants to renew lease, owners renews the lease, where tenant will also be notified about the renewed status, then has the option to accept or reject the lease.

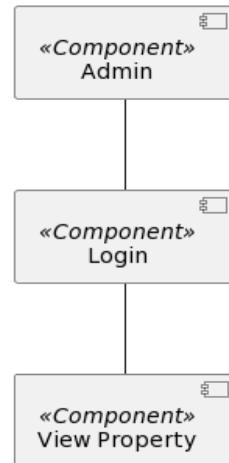


Owner Manage Lease Activity Diagram:

Admin Component Diagram:

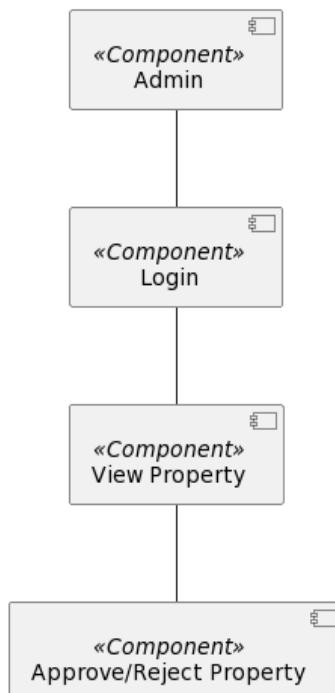
6.1.12 Admin view property

The view property is for the admin to review the property by viewing the pending approval properties and listing properties.



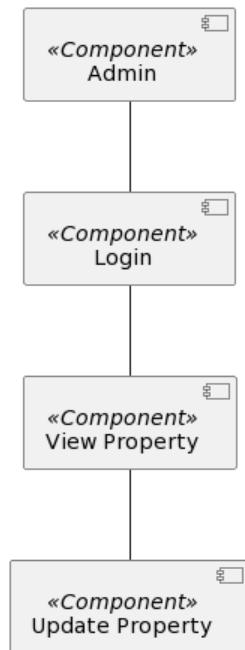
6.1.13 Admin approve property

The approve property is for the admin to review the property by viewing the pending approval properties and decide whether to approve or reject the property submitted by the owner for listing purposes.



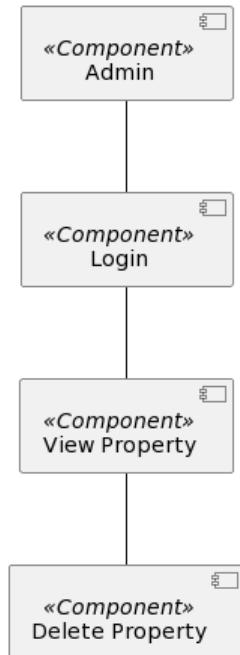
6.1.14 Admin update property

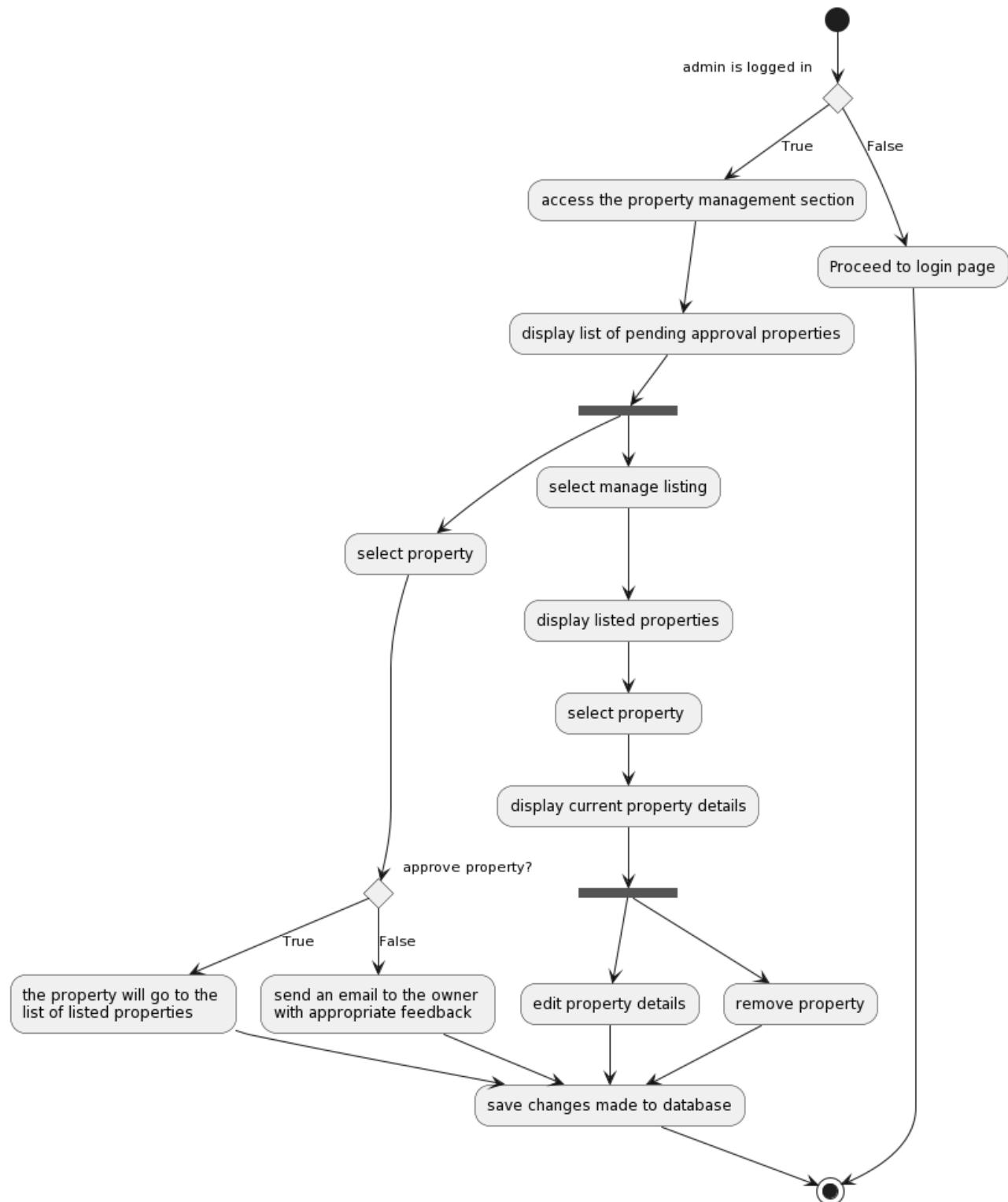
The update property function enables the admin to update the property details.



6.1.15 Admin delete property

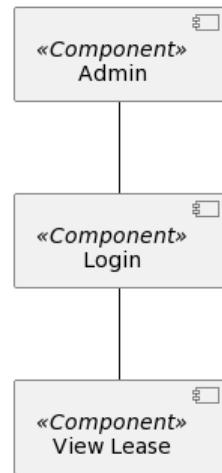
The delete property function enables the admin to delete the property listing if the property violated the rules.

**Admin Manage Property Activity Diagram**

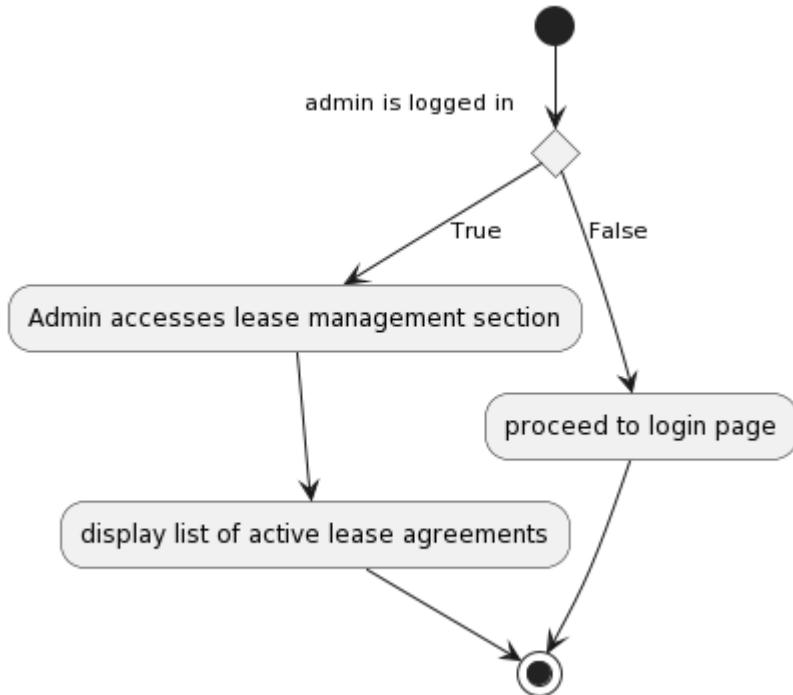


6.1.16 Admin view lease

The view lease is for the admin to review the lease by viewing the lease or agreement between the owner and tenant

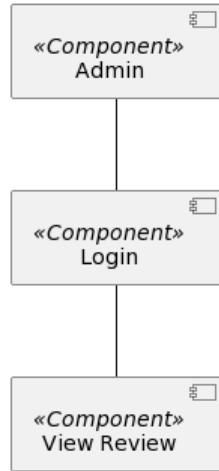


Admin Manage Lease Activity Diagram



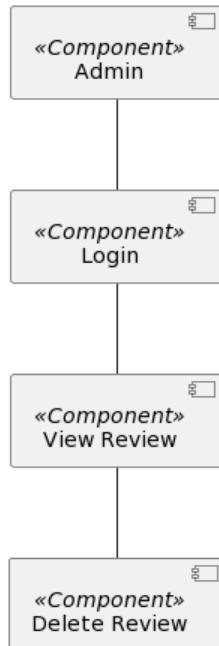
6.1.17 Admin view review

The view review is for the admin to view a list of reviews of the properties submitted by the tenant.

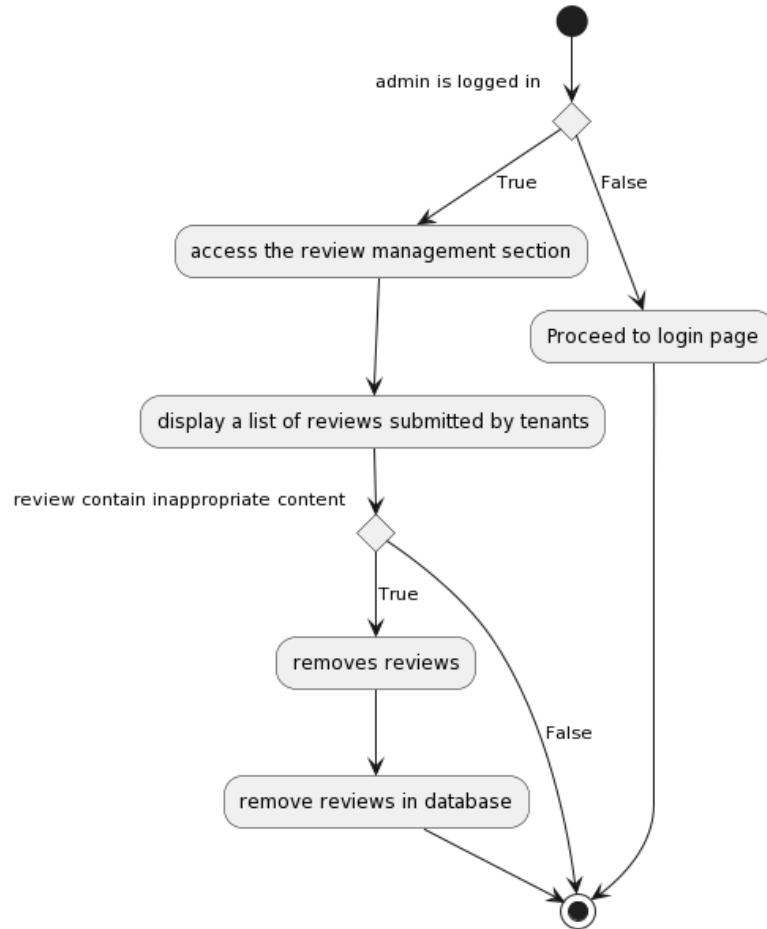


6.1.18 Admin delete review

The delete review is for the admin to delete reviews of the properties submitted by the tenant if the review violated the rules.

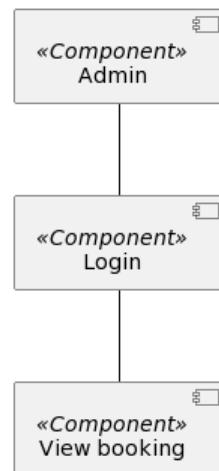


Admin Manage Review Activity Diagram



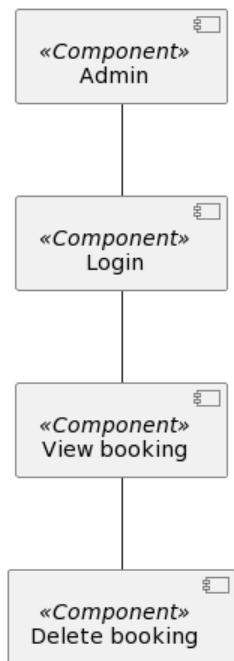
6.1.19 Admin view booking

The view booking is for the admin to view a list of booking booked by the searcher to physically view the property.

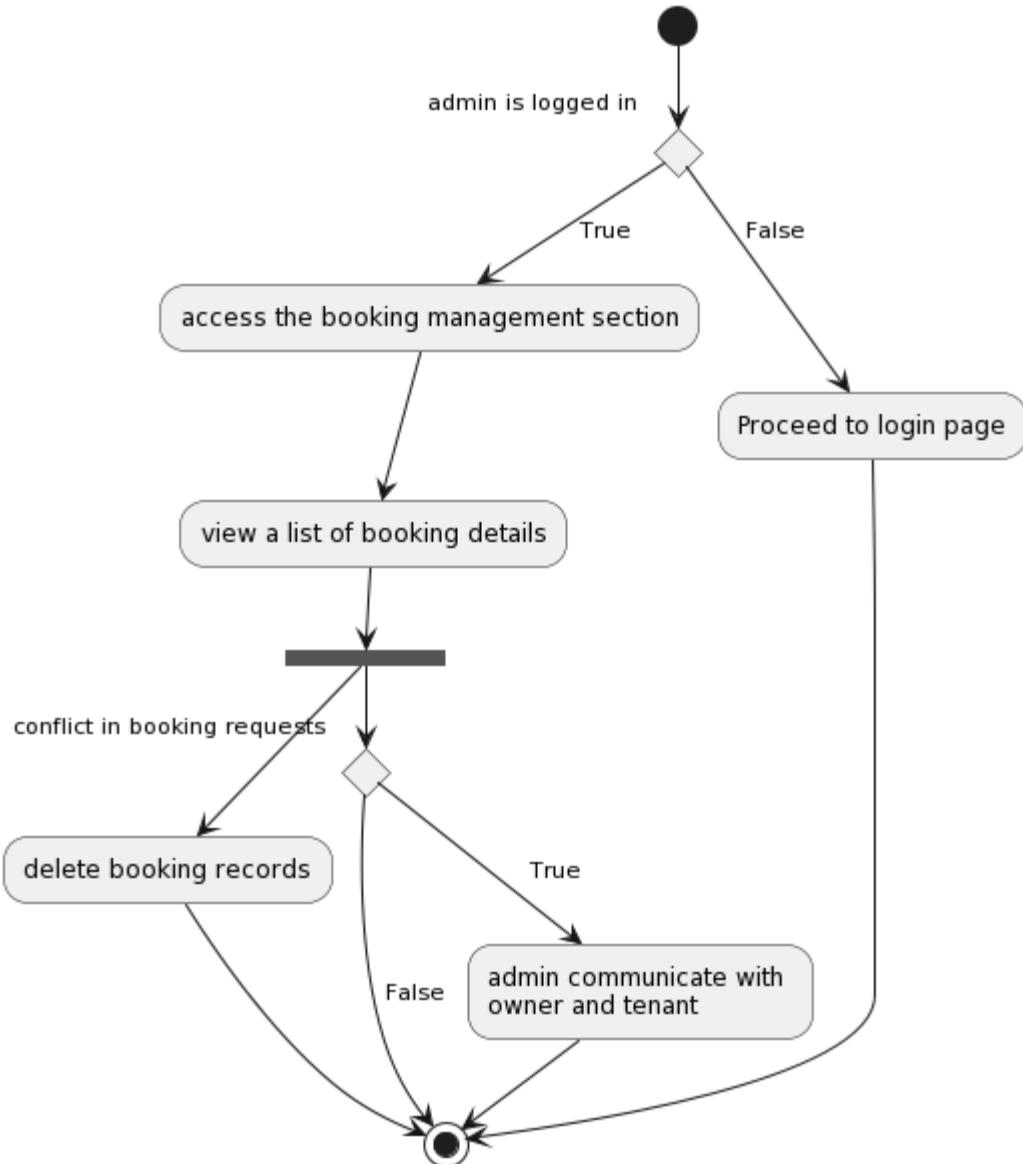


6.1.20 Admin delete booking

The delete booking is for the admin to delete the booking booked by the searcher.

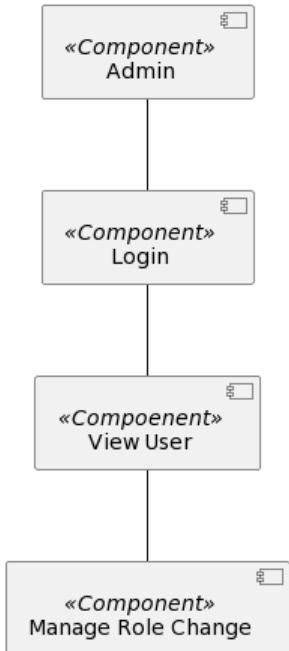


Admin Manage Booking Activity Diagram



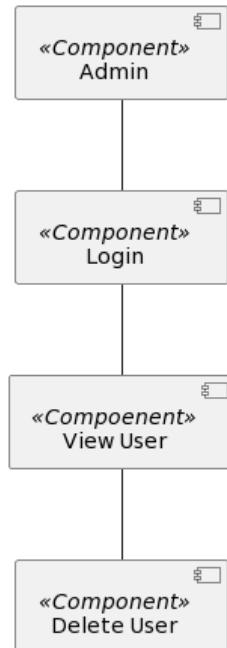
6.1.21 Admin manage role change

The admin is able to view the role change request and decide whether to approve or reject the role change request by the tenant.

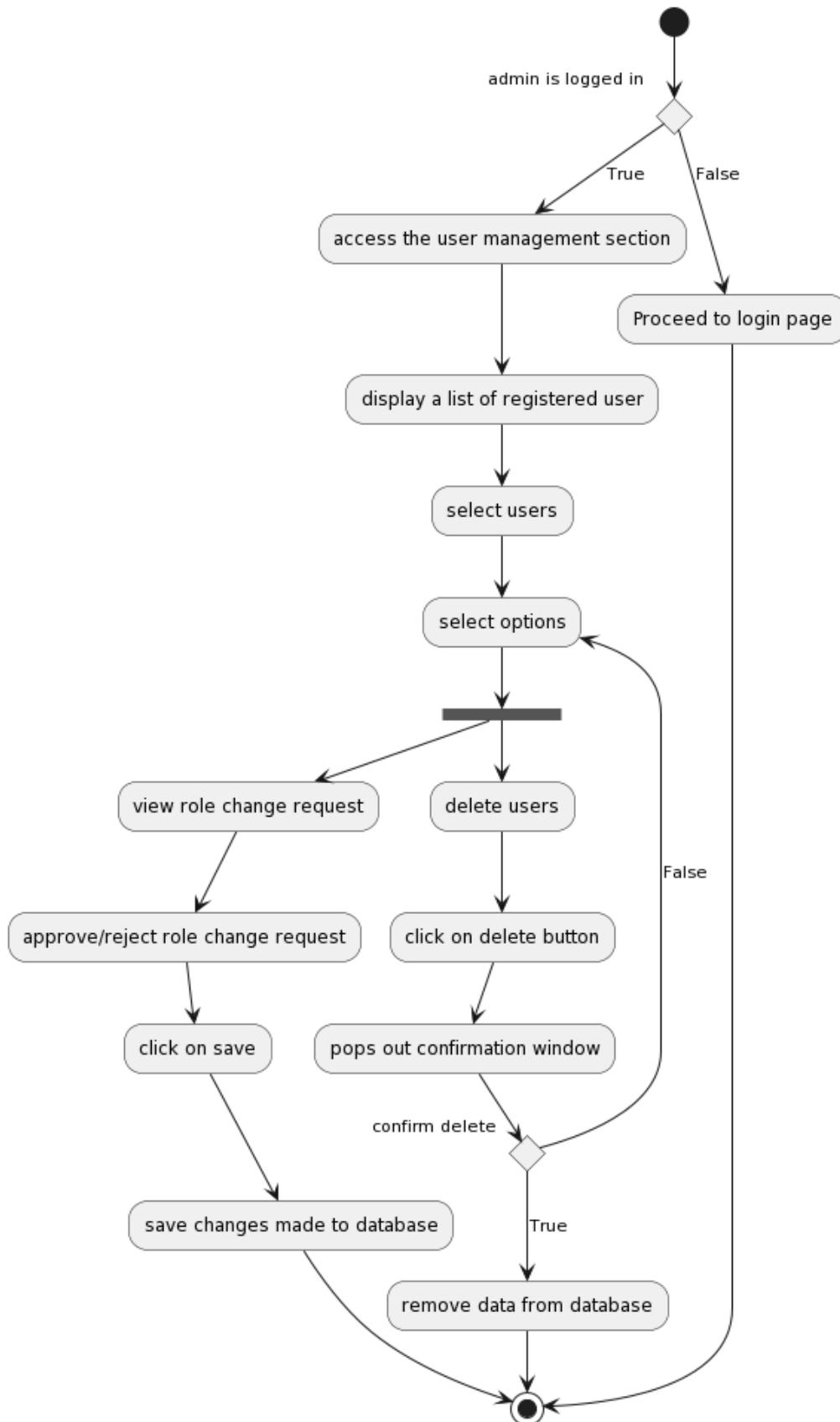


6.1.22 Admin delete user

The delete user is for the admin to delete a user account permanently in the database.

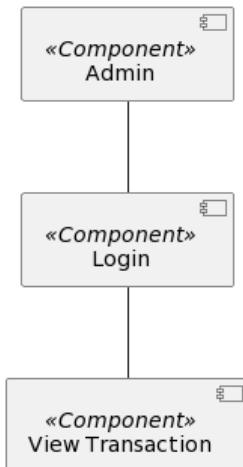


Admin Manage User Activity Diagram

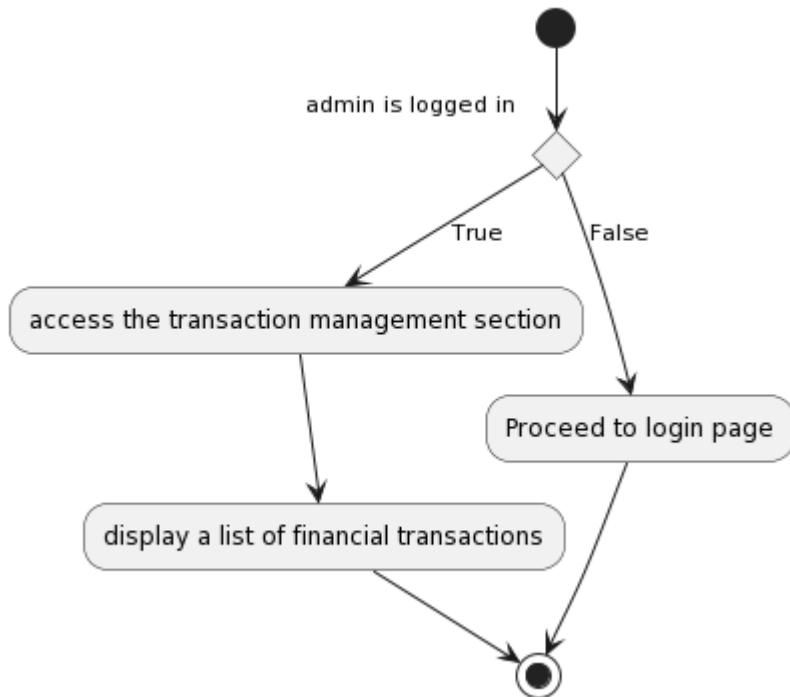


6.1.23 Admin view transaction

The view transaction is for the admin to review the transaction or payment paid by the tenant.



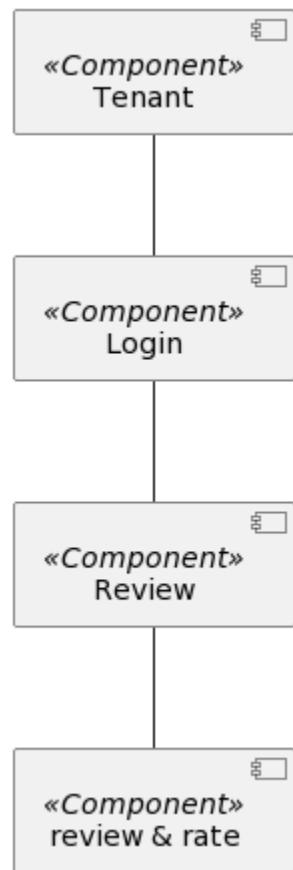
Admin Transaction Activity Diagram

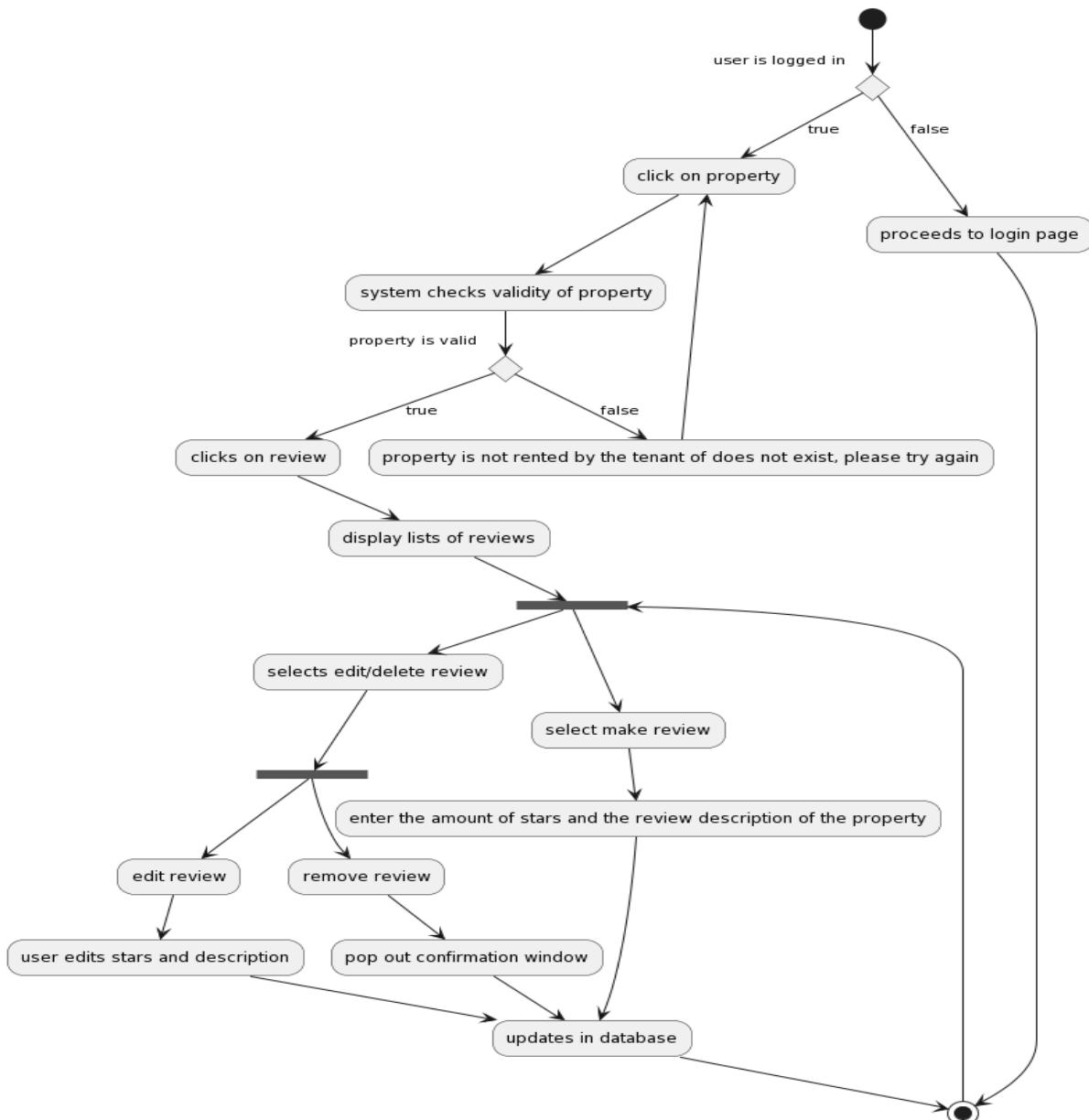


Tenant Component Diagram:

Tenant Review & Rate

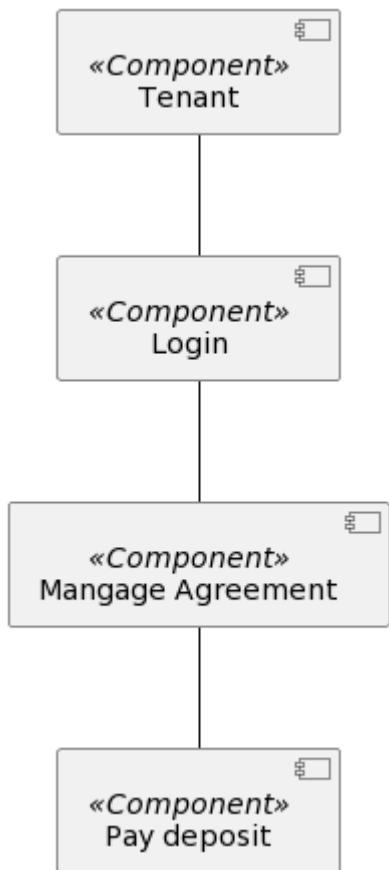
The review component where tenant can give a detailed review and rate how many stars to give to a rented property.



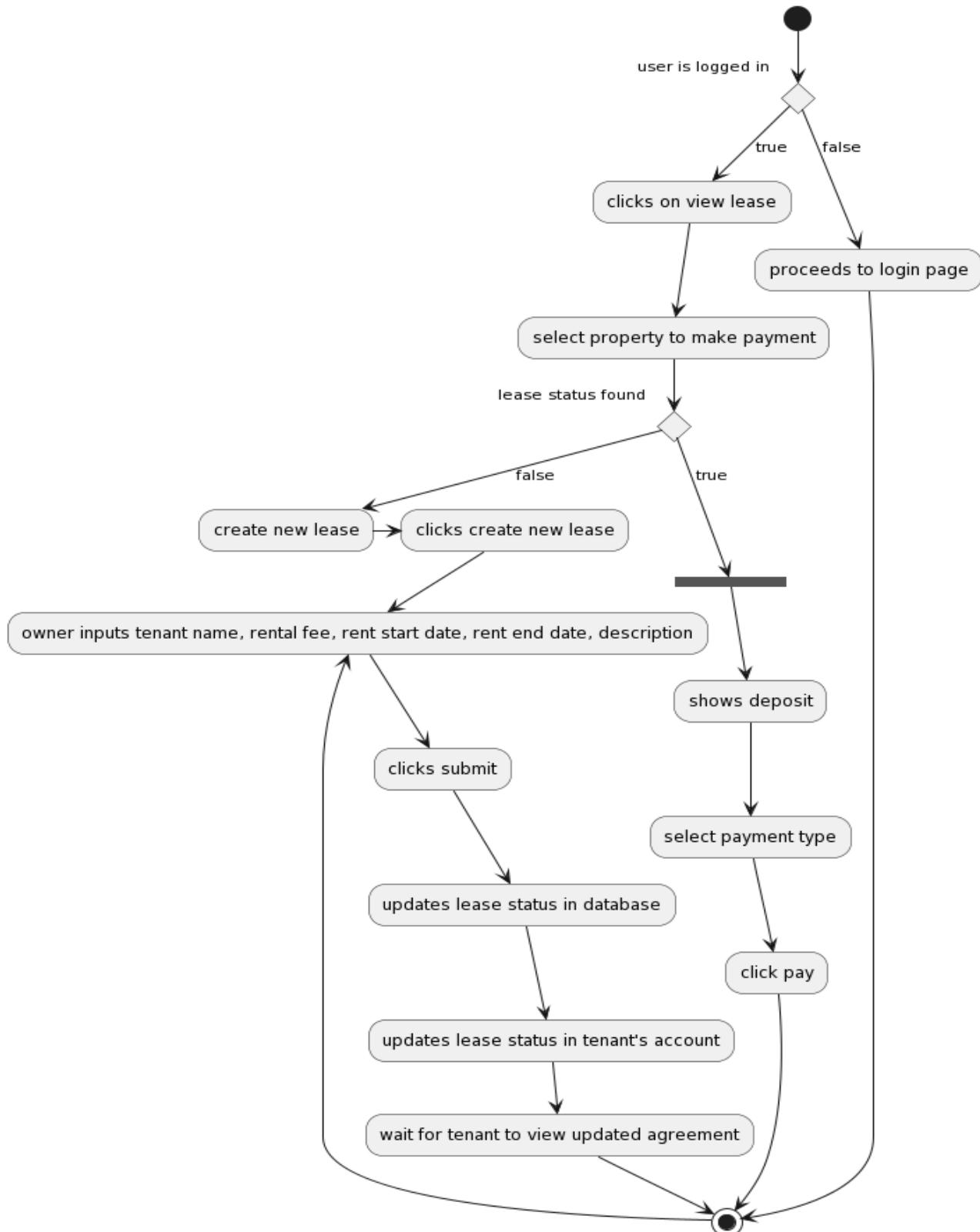
Activity Diagram for Review & Rate

Tenant Pay Deposit

The payment deposit component where tenant can pay the deposit of their rent by selecting a payment method

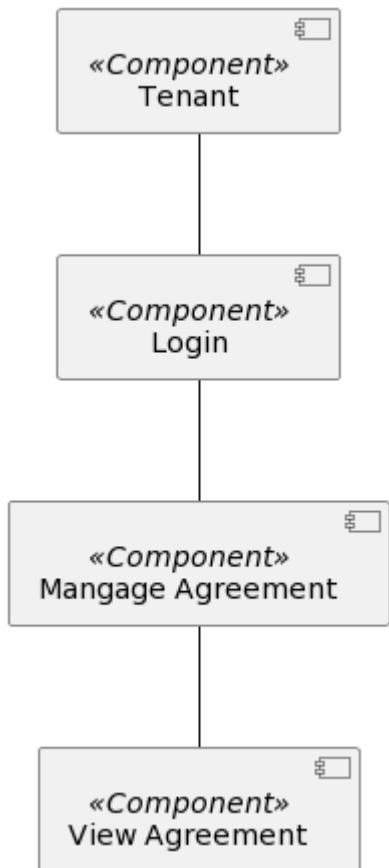


Activity Diagram for Pay Deposit



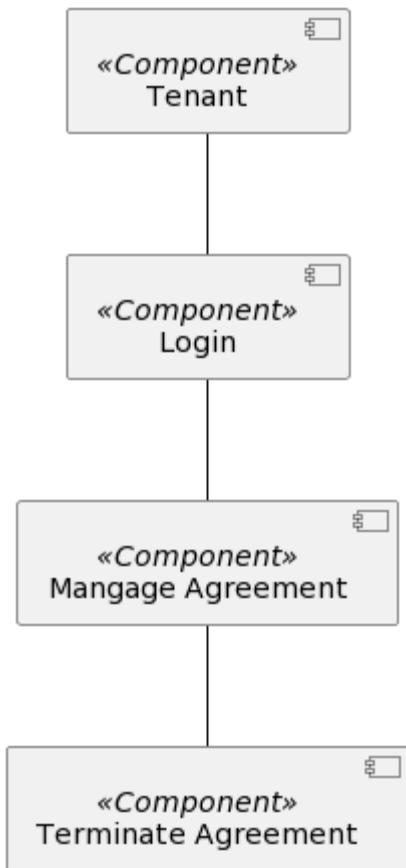
Tenant View Agreement

The view agreement component can make tenant view at the agreement of their rent



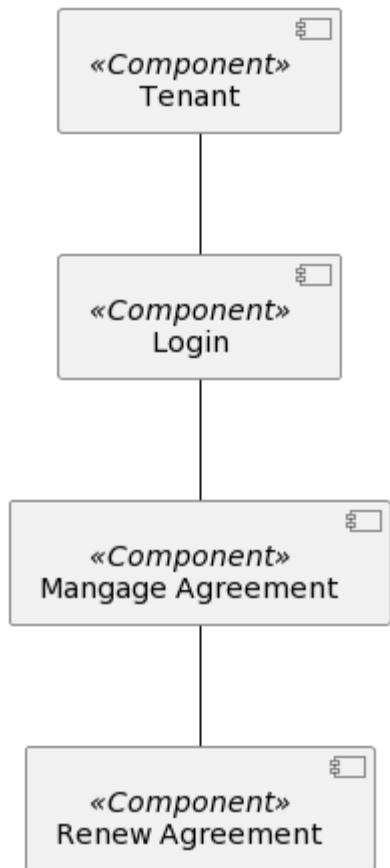
Tenant Terminate Agreement

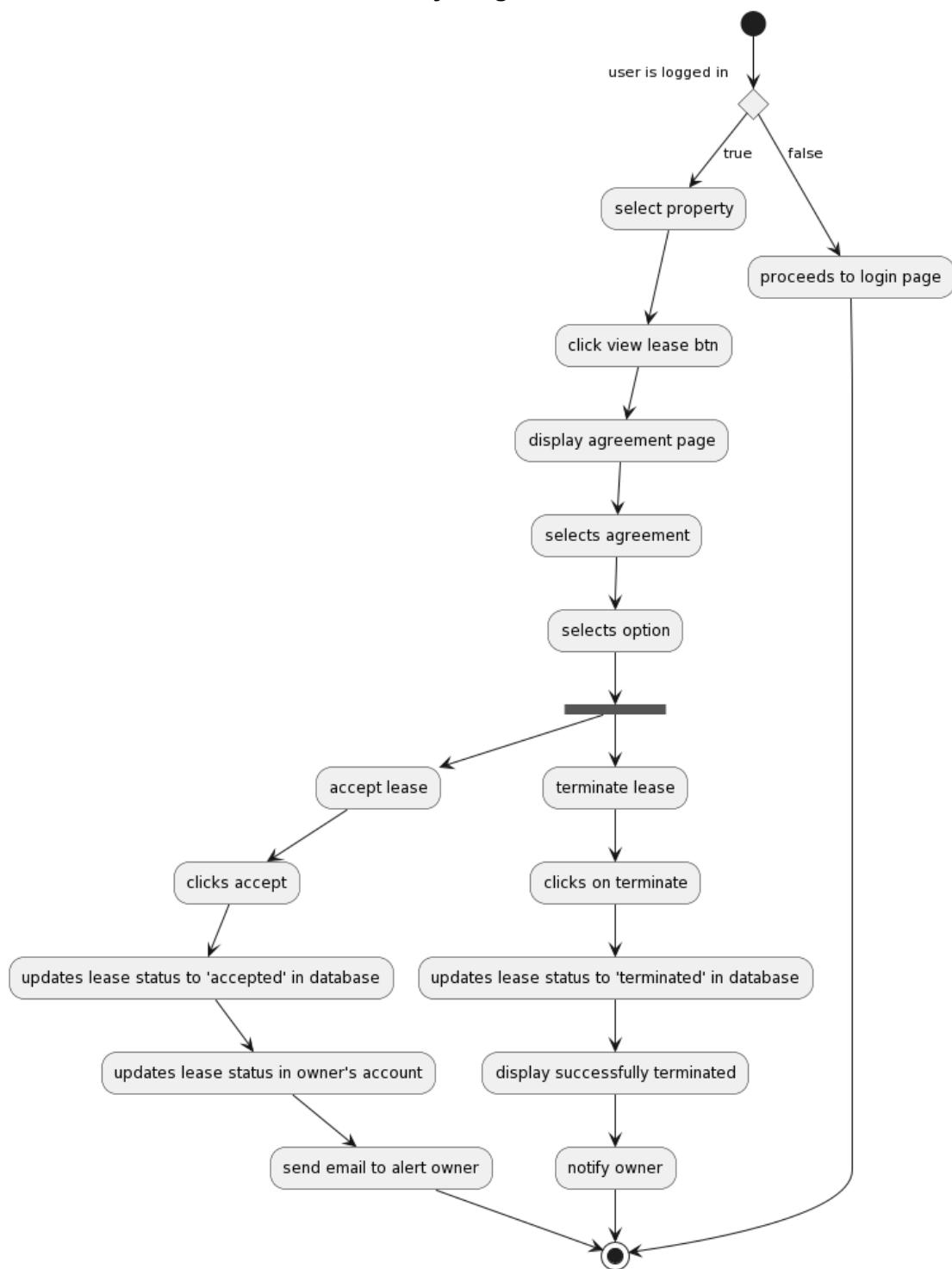
The terminate agreement can terminate an agreement if they wish to not stay in the property anymore



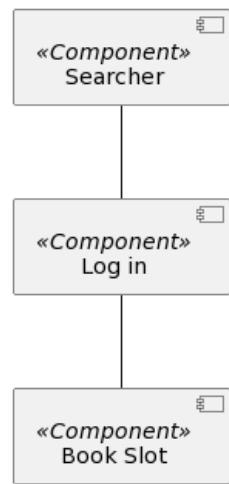
Tenant Renew Agreement

The renew agreement can renew the agreement so the tenant can stay longer in the property.

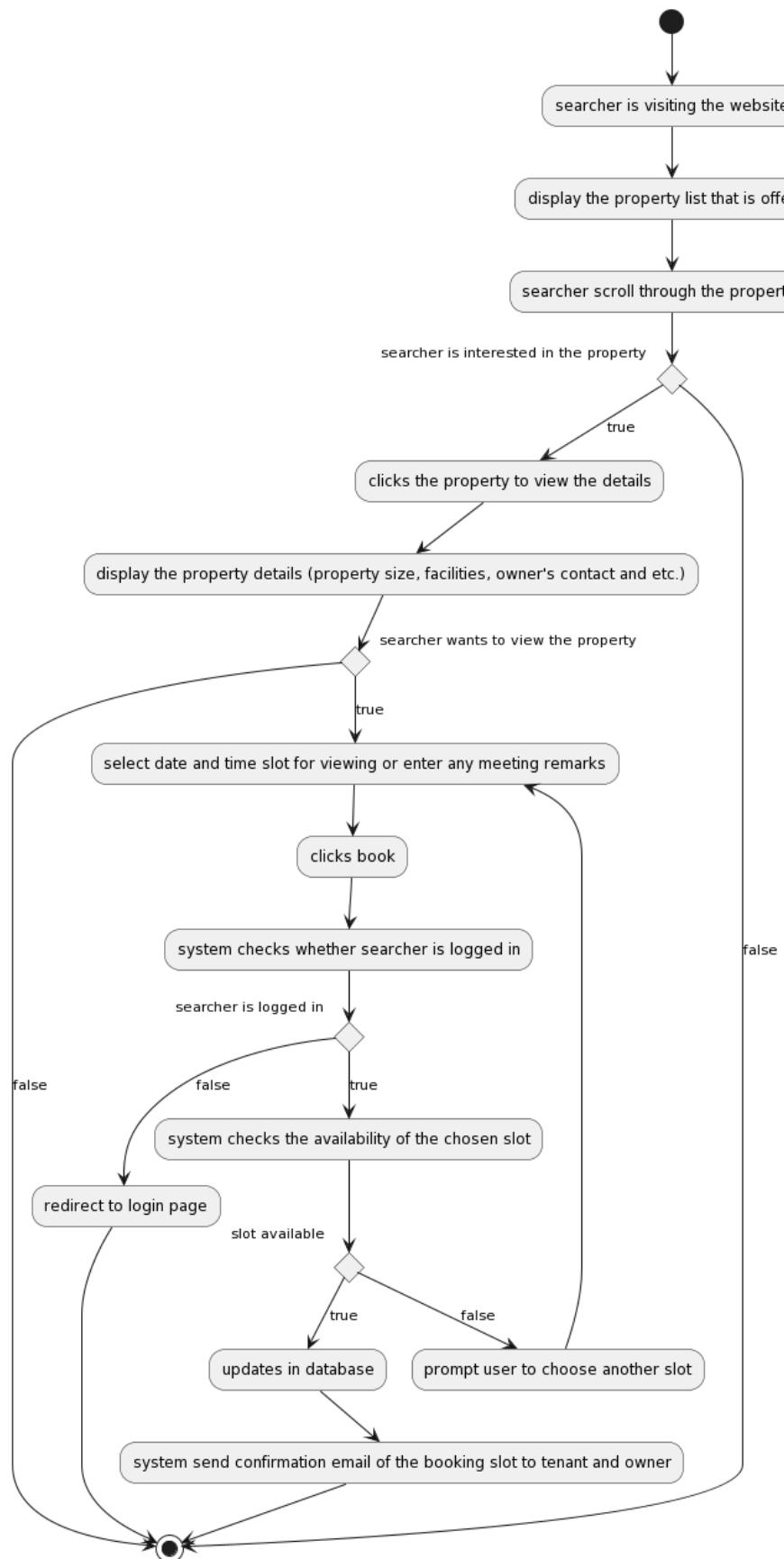


Tenant renew/terminate lease Activity Diagram

The searcher needs to login to gain access to book a slot to view the property that they are interested in.

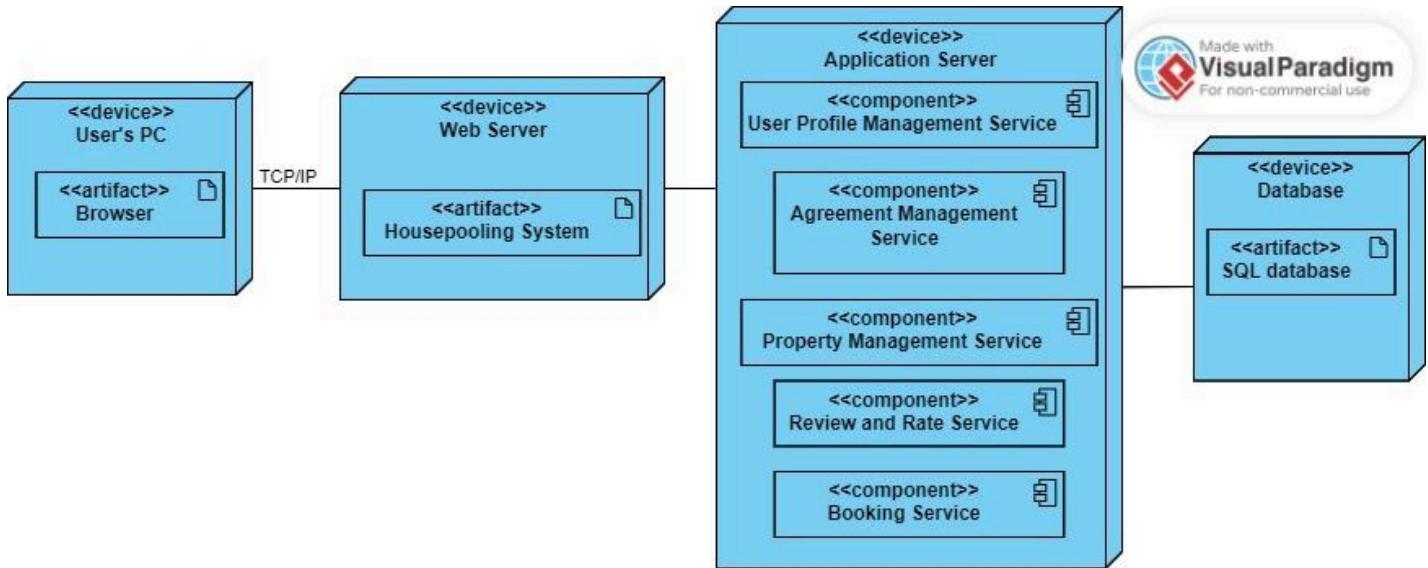


Searcher Booking Slot Activity Diagram



7 Deployment Design

7.1 Deployment Diagram



The UML deployment diagram for the Housepooling System illustrates the interactions between the system's software and hardware components. Users access the system through a browser, establishing a TCP/IP connection with the web server that hosts the Housepooling System application. The application server is responsible for managing various services integral to the Housepooling System, delivering functionalities to the users. Meanwhile, the database server is tasked with managing the SQL database that stores and retrieves data for the Housepooling System, ensuring data integrity and accessibility.

8 Updated Requirements

<This section is Optional. Place other requirements that have been updated from the SRS, particularly those that would have impact on the software design.>

...