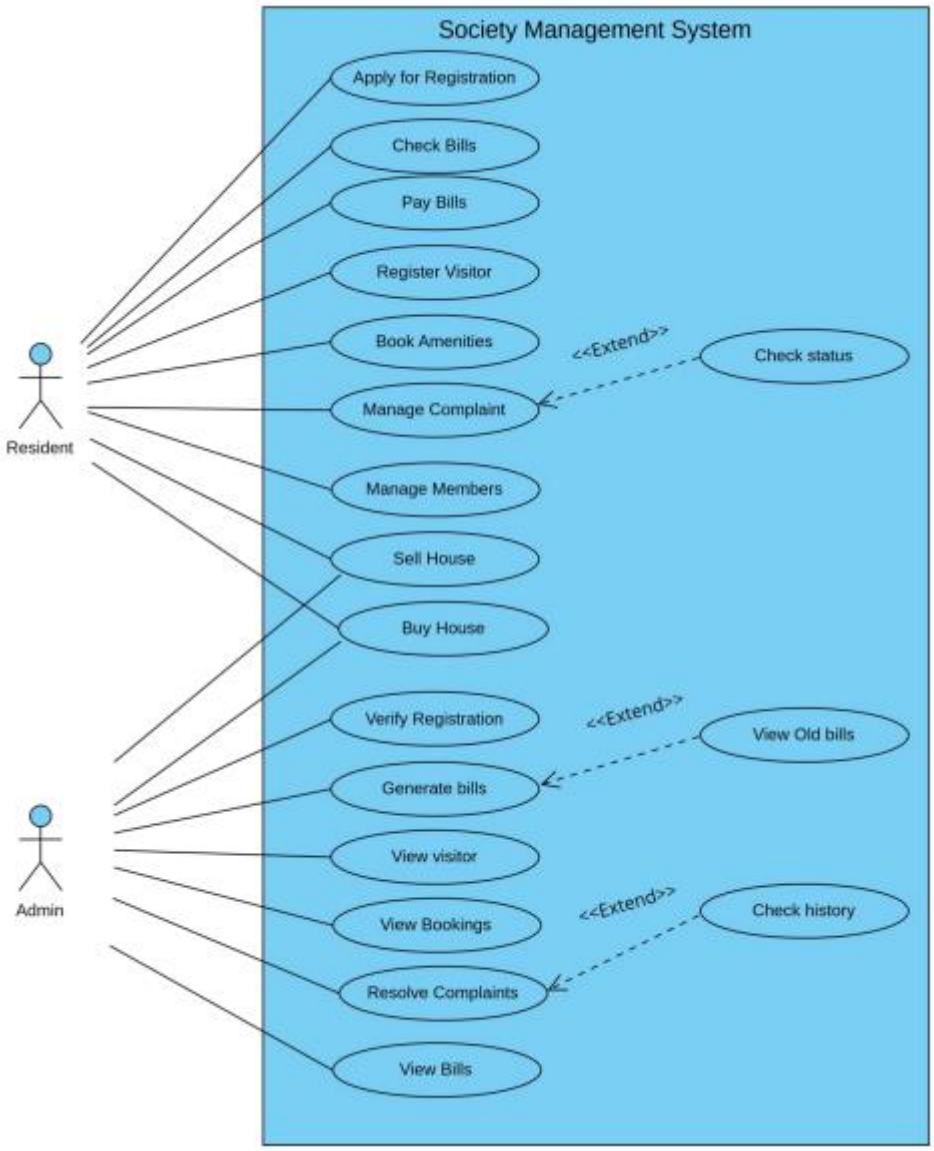
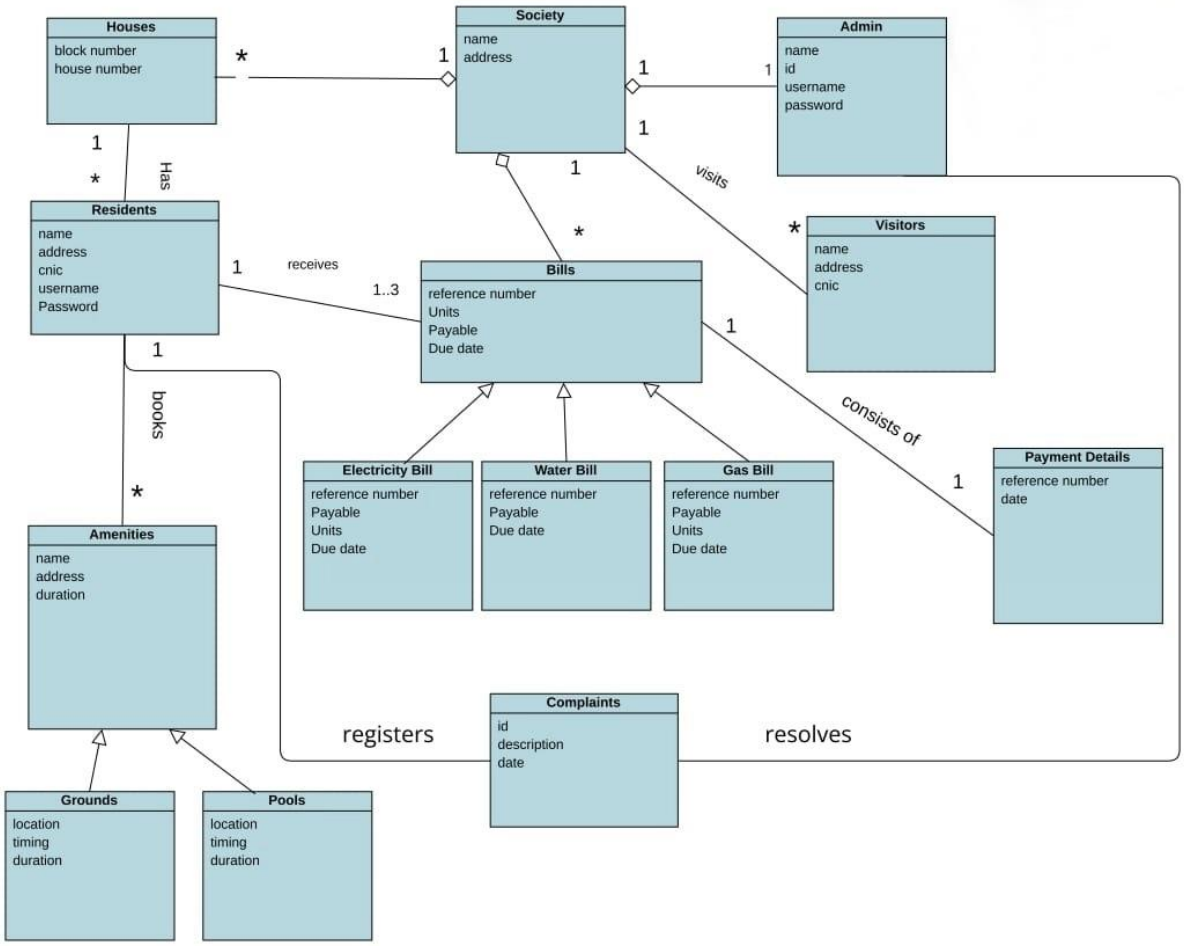

Society Management System
Apply for Registration
Check Bills
Pay Bills
Register Visitor
Book Amenities
Manage Complaint
Manage Members
Sell House
Buy House
Verify Registration
Generate bills
View visitor
View Bookings
Resolve Complaints
View Bills
Check status
View Old bills
Check history
<<Extend>>
<<Extend>>
<<Extend>>
Resident
Admin



1.1: Apply for registration

Identifier	UC-001
Name	Apply for Registration
Summary	This use case describes the process of applying for registration on a website.
Priority	High
Actors	Resident
Pre-condition(s)	The resident is not registered on the platform. The resident has access to the registration form.
Post-condition(s)	The resident's registration application is submitted. The admin receives a notification for verification.

Typical Course of Action

S#	Actor Action	System Response
1.	The resident accesses the registration form.	
2.	The resident fills out the registration form, providing personal information such as name, email, and address.	
3.	The resident chooses a password and confirms it by entering it again.	
4.	The resident submits the registration application.	
5.		The system sends a notification to the admin with the resident's registration details for verification.

Alternate Course of Action (5. Details not correct/invalid)

S#	Actor Action	System Response
6.	If the resident provides incorrect or invalid information	
7.		System identifies the errors and provides feedback to the resident

1.2: Check bills

Identifier	UC-002
Name	Check bills
Summary	This use case allows a user to view their old bills and generate a PDF of the current month's bill.
Priority	High
Actors	Resident
Pre-condition(s)	The Resident must be logged into their account. The system must have access to the user's billing history. There should be a new bill available for the current month.
Post-condition(s)	The Resident can view their old bills. The Resident can generate a PDF of the current month's bill.

Typical Course of Action

S#	Actor Action	System Response
1.	Resident logs into their account.	
2.		The system authenticates the Resident.
3.	Resident navigates to the billing section.	
4.		The system displays the billing section.
5.	Resident selects the option to view bills.	
6.		The system retrieves and displays the Resident billing history.
	Resident selects a specific bill to view.	
		The system displays the selected bill.
	Resident selects the option to generate a PDF for the current month's bill.	The system generates a PDF of the current month's bill and provides a download link.

1.3 Pay Bills

Identifier	UC-003
Name	Pay Bills
Summary	This use case outlines the steps involved in a resident verifying a bill payment, which includes confirming bank details, providing a screenshot of the payment, and verifying the time of the payment.
Priority	High
Actors	Resident
Pre-condition(s)	The resident must have initiated a bill payment. The resident must have relevant payment details, including bank information and a screenshot of the payment
Post-condition(s)	The payment verification is completed successfully. The resident is assured that the payment has been made correctly.

Typical Course of Action

S#	Actor Action	System Response
1.	The resident logs into their account on the payment platform.	
2.	The resident selects the specific bill payment they want to verify.	
3.		The system displays the selected bill payment details
4.		The system presents the bank details used for the payment.
5.	The resident reviews the bank details to confirm that the payment was made to the correct bank or account.	

6.		The system provides an option to upload the screenshot
	The resident uploads the screenshot of the payment as supporting evidence.	
	The resident verifies the time details of the payment, such as the date and time of the transaction.	
	The resident submits the verification request.	
		The system records the verification request and notifies the resident of successful submission.
Alternate Course of Action (5. Details not correct/invalid)		
S#	**Actor Action**	**System Response**
	If the resident notices any incorrect or invalid information during the verification process, they can choose to dispute the payment.	
		The system acknowledges the dispute request.
		The system initiates an investigation into the disputed payment.
		The system may request additional information or documentation from the resident to support the dispute.
		The resident receives updates on the status of the dispute resolution process.
		Once the dispute is resolved, the system notifies the resident of the outcome and any necessary actions to be taken.

1.4: Register a visitor:

Identifier	UC-004
Name	Register a visitor
Summary	This use case outlines the process for residents of a secure society to register visitors who wish to enter the premises. This use case takes into consideration the constraints of restricted access for security reasons.
Priority	Medium
Actors	Resident
Pre-condition(s)	The resident must be a member of the secure society. The resident must have a valid reason for the visitor's entry. The date and time of the visit must be predetermined. The visitor's identity and purpose of the visit must be known.
Post-condition(s)	The visitor is registered and granted access at the specified date and time. Security personnel are informed of the registered visitor.

Typical Course of Action

S#	**Actor Action**	**System Response**
1.	Initiates the visitor registration process.	
2.		System prompts the actor to provide visitor details including visitor's name, purpose of the visit, and date of the visit and time of the visit.
3.	Actor enters the required information	
4.		System verifies the information.
5.	If the information is valid.	
6.		System notifies security personnel grant access to the visitor at the specified date and time.
		System records the visitor's registration.

Alternate Course of Action (5. Details not correct/invalid)

S#	**Actor Action**	**System Response**
6.	If the actor (Resident) enters incorrect or invalid information	

		System displays an error message.
	Actor corrects the information.	
		System re-verifies the corrected information
	If the information is now valid, the system proceeds with the typical course of action.	
	If the information remains invalid after correction, the registration is denied, and the visitor is not granted access.	

1.5: Book amenities:

Identifier	UC-005
Name	Book Amenities
Summary	This use case describes the process of booking amenities and it ensures that only one amenity can be booked by one person at a time, and allows for bookings on different days or times.
Priority	High
Actors	Resident System
Pre-condition(s)	The visitor must have access to the booking system. The visitor must be logged in.
Post-condition(s)	The amenity is successfully booked for the specified date and time. The visitor's booking is recorded in the system.

Typical Course of Action

S#	**Actor Action**	**System Response**
1.	The resident selects the type of amenity they want to book	
2.	The resident specifies the date and time they want to book the amenity for.	
3.	The resident confirms the booking.	

4.		The system checks for availability of the selected amenity at the specified date and time.
5.		If the amenity is available, the system books it for the visitor.
6.		The system confirms the booking to the resident.
		The system records the booking in the database.
Alternate Course of Action (5. Details not correct/invalid)		
S#	**Actor Action**	**System Response**
6.	The resident provides incorrect or invalid details during the booking process.	.
		The system detects the incorrect or invalid details.
		The system informs the visitor of the error and asks them to correct the details.
	The visitor corrects the details.	

1.6: Manage Complaint

Identifier	UC-006
Name	Manage Complaint
Summary	This use case describes how a resident can submit a complaint, have it resolved, and verify its resolution status.
Priority	Medium
Actors	Resident
Pre-condition(s)	The resident must be registered in the system. The resident must be logged into the system.
Post-condition(s)	The complaint is recorded in the database. The complaint data is stored in the database and not deleted.

Typical Course of Action

S#	Actor Action	System Response
1.	Resident logs into the system.	
2.	Resident navigates to the "Complaints" section	
3.		The system displays a list of existing complaints and also can write description and a "Submit Complaint" button.
4.	Resident clicks on the "Submit Complaint" button.	
5.		The system records the complaint in the database with a status of "Unresolved.
		The system displays a confirmation message to the resident.

	Resident can choose to check the status of the complaint or log out.	
		If the resident chooses to check the status of the complaint, they can see it marked as "Unresolved" in the complaints list.
Alternate Course of Action (5. Details not correct/invalid)		
S#	**Actor Action**	**System Response**

1.7: Manage Members:

Identifier	UC-007	
Name	Manage Members	
Summary	This use case describes the process of managing member accounts within a system. It includes actions such as verifying login credentials, changing passwords, and handling scenarios where login details are not correct or invalid	
Priority	High	
Actors	Resident	
Pre-condition(s)	The resident is registered as a member in the system. The resident has access to their login credentials (username and password).	
Post-condition(s)	The member's account information is updated if necessary. The member's password is changed if requested.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1.	The Resident initiates the member management process.	
2.		The system presents options for member management, including changing the password and verifying login credentials.

3.	The Resident selects the "Verify Login" option.	
4.		The system prompts the Resident to enter their current username and password.
5.	The Resident enters their current username and password.	The system verifies the provided login credentials.
	If the login credentials are correct, the Resident proceeds to the next step. If the credentials are incorrect, the Resident is prompted to re-enter them	
	The Resident selects the "Change Password" option.	
	The system prompts the Resident to enter a new password.	
		The system validates the new password according to password policy rules, and if the password meets the requirements, it is update
	The Resident confirms the new password.	
		The system updates the member's password.
	The Resident confirms the member management process is complete.	
		The system displays a confirmation message and returns to the main menu.
Alternate Course of Action (5. Details not correct/invalid)		
S#	**Actor Action**	**System Response**
6.	The Resident enters incorrect login credentials.	.
		The system informs the Resident that the login credentials are incorrect and provides an option to retry or return to the main menu.
	The Resident can choose to retry or return to the main menu.	

		If the Resident chooses to retry, they are prompted to re-enter their login credentials. If they choose to return to the main menu, the process ends.

1.8: Verify registration:

Identifier	UC-008
Name	Verify registration
Summary	This use case outlines the steps for an admin to verify the registration of a new user in the system.
Priority	High
Actors	Admin
Pre-condition(s)	Admin has logged into the system. A new user has completed the registration process but requires verification.
Post-condition(s)	The user's registration status is updated to "verified." The user gains access to their account and system features.

Typical Course of Action

S#	**Actor Action**	**System Response**
1.	Admin selects the "Pending Registrations" or similar option from the admin dashboard.	
2.		The system displays a list of users with pending registrations.
3.	Admin selects the user to be verified.	

4.		The system provides details of the selected user's registration information.
5.	Admin reviews the registration information for accuracy and completeness.	
		The system provides options to mark the user as "Verified" or "Not Verified."
	Admin selects "Verified" if the registration information is correct.	
		The user's registration status is updated to "verified."
		The user is notified of their verified status and provided access to their account.
	Admin selects "Not Verified" if the registration information is incorrect or incomplete.	
		The user's registration status remains as "pending."
		The admin may choose to send a notification or request additional information from the user to complete the registration.
Alternate Course of Action (5. Details not correct/invalid)		
S#	**Actor Action**	**System Response**
6.	Admin selects "Not Verified" if the registration information is incorrect or invalid.	.
		The user's registration status remains as "pending."
		The admin may choose to send a notification or request additional information from the user to complete the registration.

1.9 Generate bills:

Identifier	UC-009
Name	Generate bills

Summary	This use case outlines the process by which an admin can generate bills for customers.
Priority	High
Actors	Admin
Pre-condition(s)	The admin must be logged into the billing system.
Post-condition(s)	Bills for the selected customers are generated and saved in the system. Customers receive their bills via email or other designated communication channels.

Typical Course of Action

S#	Actor Action	System Response
1.	The admin logs into the billing system	
2.		The system verifies the admin's credentials and grants access to the Admin Control Center.
3.	The admin navigates to the "Generate Bills" section	
4.		The system displays options for selecting the billing period and criteria for generating bills.
5.	The admin selects the billing period and criteria and initiates the bill generation process.	
		The system generates bills based on the selected criteria and displays a summary of the generated bills.
	The admin reviews the summary of generated bills for accuracy.	
		The system presents the admin with options to make corrections or adjustments if necessary.
	The admin confirms the accuracy of the generated bills.	
		The system finalizes the bills and saves them in the system.
	The admin selects the option to notify bills to customers.	

		The system notify to customers via email .
Alternate Course of Action (5. Details not correct/invalid)		
S#	**Actor Action**	**System Response**
6.	If the admin initiates the bill generation process with incorrect or invalid criteria.	
		The system displays an error message indicating that the criteria are not valid or the details provided are incorrect.
	The admin reviews the error message and either corrects the criteria or provides valid details.	
		The system allows the admin to make corrections and reinitiate the bill generation process.

1.10 View visitors:

Identifier	UC-010	
Name	View visitors	
Summary	This use case allows the admin to view a list of all visitors in the system.	
Priority	High	
Actors	Admin	
Pre-condition(s)	The admin is authenticated and logged into the system.	
Post-condition(s)	The admin is presented with a list of all visitors.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1.	The admin selects the "Access Visitor Records" option from the main menu.	
2.		The system retrieves a list of all visitors stored in the database.

3.	The system displays a paginated list of visitors, including their names, contact information, and visit details.	
4.		The list of visitors is displayed to the admin, allowing them to scroll through the pages to view all the visitors.
5.	The admin can click on a visitor's name to view more details or perform actions	
		If the admin clicks on a visitor's name, the system displays a detailed view of the visitor's information
Alternate Course of Action (5. Details not correct/invalid)		
S#	**Actor Action**	**System Response**
6.	The admin enters incorrect login credentials or encounters an authentication issue.	
		The system displays an error message, prompting the admin to enter the correct credentials or contact support for assistance.
	The admin selects the "Access Visitor Records" option, but there is an issue with the database connection or data retrieval.	
		The system displays an error message, informing the admin that there is a technical issue and advises them to try again later or contact technical support.
	The admin attempts to view a visitor's details that do not exist in the system.	
		The system displays a message indicating that the requested visitor's details are not found.

1.11 View bookings:

Identifier	UC-011
Name	View bookings
Summary	This use case describes how an admin can view bookings in the system.
Priority	High
Actors	Admin
Pre-condition(s)	The admin is logged into the system. There are existing bookings in the system.
Post-condition(s)	The admin successfully views the bookings.

Typical Course of Action

S#	Actor Action	System Response
1.	the admin selects the "View Bookings" option from the admin dashboard	
2.		The system displays a list of existing bookings, including details such as booking ID, customer name, date, time, and location.

Alternate Course of Action (5. Details not correct/invalid)

S#	Actor Action	System Response
6.	The admin notices incorrect or invalid booking details while viewing bookings.	
		The system provides the admin with different options
		Cancel the booking if it's deemed necessary due to inaccuracies or invalid information.
		Update the booking status as "Pending" while awaiting clarification or correction from the resident.

1.12: Resolve complaints:

Identifier	UC-012

Name	Resolve complaints
Summary	This use case describes how an admin resolves a customer complaint.
Priority	High
Actors	Admin
Pre-condition(s)	Admin is logged into the complaint resolution system.
Post-condition(s)	The customer complaint is marked as resolved. If necessary, appropriate actions are taken to address the complaint.

Typical Course of Action

S#	**Actor Action**	**System Response**
1.	The admin logs into the complaint resolution system.	
2.		The admin is successfully logged in and presented with a list of unresolved complaints.
	The admin selects an unresolved complaint from the list.	
		The details of the selected complaint are displayed on the screen.
	The admin reviews the complaint details and gathers any additional information if required.	
	The admin takes appropriate actions to resolve the complaint	
	The admin Implementing corrective measures to prevent similar complaints in the future.	
	Once the complaint is resolved, the admin updates the status of the complaint in the system to "Resolved."	
		The complaint status is changed to "Resolved," and the system records the date and time of resolution.

Alternate Course of Action (5. Details not correct/invalid)

S#	**Actor Action**	**System Response**

6.	The admin reviews the complaint details but realizes that the information provided is incorrect or invalid.	
		The admin may choose to contact the customer to verify the details or request more accurate information.

1.13: View bills

Identifier	UC-013	
Name	View bills	
Summary	This use case describes how an admin can view all bills in the system.	
Priority	High	
Actors	Admin	
Pre-condition(s)	Admin is logged into the system. Bills exist in the system.	
Post-condition(s)	Admin successfully views all bills. The system remains in the same state.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1.	Admin navigates to the "Billing Overview" section of the admin interface.	
		The system displays a list of all bills, including details such as bill ID, date, amount, and status.
Alternate Course of Action (5. Details not correct/invalid)		
S#	**Actor Action**	**System Response**

6.	Admin enters incorrect login credentials.	
		The system displays an error message indicating that the login credentials are incorrect.
	Admin faces a network connectivity issue while trying to access billing data.	
		The system displays an error message indicating a network problem and advises the admin to check their internet connection.

1.14: Sell house

Identifier	UC-014	
Name	Sell a house	
Summary	This use case describes the process of a resident selling their house through an admin-managed real estate system.	
Priority	High	
Actors	Admin Resident	
Pre-condition(s)	Resident has an active account in the real estate system. Admin has access to the real estate system.	
Post-condition(s)	The resident successfully lists their house for sale. The admin reviews and approves the house listing.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1.	Resident logs into their account.	
	Resident navigates to the "Sell a House" section.	
	Resident enters the details of the house they want to sell, including address, price, and description.	

	Resident uploads photos of the house.	
	Resident submits the listing for review.	
		The system confirms the submission and informs the resident that the listing is pending admin approval.
	Admin logs into their admin account and navigates to the "Pending Listings" section.	
	Admin reviews the details and photos of the house listing.	
	Admin either approves or rejects the listing.	
		If approved, the system publishes the house listing on the platform.
		If rejected, the system sends a notification to the resident with the reason for rejection.
Alternate Course of Action (5. Details not correct/invalid)		
S#	**Actor Action**	**System Response**
6.	Resident enters incomplete or incorrect details of the house.	
	Resident uploads inappropriate or irrelevant photos and Resident submits the listing.	
		The system provides error messages indicating the issues with the submitted information.
		Resident corrects the information and resubmits the listing.
	Admin reviews the listing with incorrect or invalid details.	
		Admin rejects the listing and sends a notification to the resident explaining the reasons for rejection.

1.15: Buy house

Identifier	UC-015
Name	Buy house
Summary	This use case describes the process of a resident purchasing a house through an admin-managed real estate system.
Priority	High
Actors	Admin Resident
Pre-condition(s)	Resident has an active account in the real estate system. Admin has access to the real estate system. Houses are listed for sale on the real estate platform.
Post-condition(s)	The resident successfully purchases a house. The admin processes the purchase request and updates the house's status.

Typical Course of Action

S#	Actor Action	System Response
1.	Resident logs into their account.	
	Resident searches for houses based on their preferences (e.g., location, price, size).	
	Resident selects a house listing they are interested in.	
	Resident reviews the details and photos of the selected house.	
	Resident clicks on the "Buy Now" or "Contact Seller" button.	
		If "Buy Now" is selected, the system initiates the purchase process.

		If "Contact Seller" is selected, the system sends a message to the seller indicating the resident's interest.
	Admin receives a notification of the purchase request.	
	Admin reviews the request and verifies.	
		If the resident is verified, the system marks the house as "Pending Sale" and notifies the resident and seller.
	If the purchase is approved, the resident proceeds with the payment and provides necessary details.	
		The system processes the payment, updates the house's status to "Sold," and sends confirmation to both the resident and the seller.
Alternate Course of Action (5. Details not correct/invalid)		
S#	**Actor Action**	**System Response**
6.	Resident tries to initiate the purchase with incorrect or incomplete details.	
		The system provides error messages indicating the issues with the submitted information..
		Resident corrects the information and resubmits the purchase request.
	Admin receives the corrected purchase request.	
		Admin reviews the corrected request, and if the details are now correct, proceeds with the purchase process as outlined in the typical course of action.