Software Requirements Specification

For

Society Management System

Version 1.1
Prepared By:

Abdul Ahad	**22L-7852**
Faraz Iftikhar	**22L-7974**
Hamza Naveed	**22L-7871**
Ahmad Tahir	**22L-7869**

Table of Contents

1. Introduction

1.1 Purpose

The Purpose is to create a seamlessly integrated and user-friendly Society Management System that fosters transparent communication, efficient governance, and a sense of community. Our vision is to empower residents, committee members, and management personnel with a digital platform that not only simplifies daily tasks but also nurtures a vibrant and engaged living environment.

The concept is groundbreaking, especially considering the prevalent practices in housing societies where residents and management rely on fragmented chat applications like WhatsApp and Telegram for communication. Additionally, the persistence of paper-based challans for society maintenance bills and the lack of access for security guards to efficient communication platforms highlight significant inefficiencies in existing systems.

Unlike traditional housing societies websites that primarily focus on catering to potential real estate buyers and handle amenities reservations through physical visits, our innovative approach addresses these shortcomings comprehensively. By introducing a centralized web-based platform, we aim to bridge communication gaps, streamline maintenance bill processes, and empower security personnel with efficient tools. Residents will no longer have to visit the society premises for amenity reservations, enhancing convenience and accessibility. This pioneering initiative is set to revolutionize the way housing societies operate, fostering a sense of community, transparency, and security among residents.

1.2 Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents. This document's content is written in Arial font with a size of 11. The body's major titles are written in Times New Roman, font size 18, while the subtitles are written in Times New Roman, font size 14. Requirements have been broken down into main functional categories, with each functional area subdivided into features. Within the laid-out document format, there are functional and nonfunctional needs. This document consists of text and different diagrams to elaborate the requirements. Every functional requirement is in hierarchical order with further division to the minimum level.

1.3 Intended Audience and Reading Suggestions

The intended audience for this Society Management System (SMS) document encompasses a diverse group of stakeholders, each playing a crucial role in the project's success. This includes:

•**Project Manager**: Utilizes the document to make informed decisions regarding the development process.

•**Test Case Engineers**: Refer to the document for crafting test cases aligned with specified requirements.

•**Software Engineers**: Rely on the document to guide the development process, ensuring the software meets outlined requirements.

•**Residents**: End-users who directly engage with the SMS, relying on the document to understand and navigate the system.

•**Stakeholders and Managers**: Gain valuable insights into the forthcoming SMS, aiding their understanding and expectations.

The document is structured to provide a seamless reading experience, beginning with an overview section offering essential background knowledge. Tailored reading sequences are recommended for different user types, ensuring developers, project managers, users, and testers can efficiently access details relevant to their roles. Comprehensive explanations, supported by diagrams, detail the SMS's goals, characteristics, interfaces, functionalities, and operational constraints, fostering a holistic understanding for all involved parties.

1.4 Project Scope

To develop a cutting-edge web-based Society Management System aimed at enhancing operational efficiency and effectiveness within the community. The system's primary objectives include fostering seamless communication and collaboration among society members and management, ultimately elevating transparency and accountability in the management processes.

1.5 References

[1] IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

Revision History

Name	Date	Reason For Changes	Version
Requirements Document	25-09-2023	New Document	1.0
Requirements Document	30-11-2023	Revised Document	1.1

2. SRE Product Description

2.1 Product Perspective

The Society Management System (Resimate) is an innovative, yet-to-be-introduced product that stands as a pioneering solution in the realm of community management. It's not a mere replacement for existing systems but a new, self-contained product designed to redefine the landscape of housing society operations. Unlike conventional approaches relying on fragmented communication tools, Resimate introduces a revolutionary platform that seamlessly integrates features like House Owner Contacts, Upcoming Events, Maintenance Bills, and more. The system also interfaces with third-party service providers, ensuring smooth integration of functionalities like payment processing and vehicle tracking. Resimate not only streamlines administrative tasks but also enhances community engagement through features like Resident Voting, Campaign Contributions, and Visitor Management. This unique product, as outlined in the Software Requirements Specification, is poised to address the shortcomings of current practices by providing a holistic and user-friendly solution for transparent communication, efficient governance, and vibrant community engagement.

2.2 Product Functions

2.2.1. User Information Management: Enables residents to view and update contacts.

2.2.2. Event Management: Displays upcoming events and facilitates organization.

2.2.3. Billing and Payments: Generates electronic maintenance bills and allows secure online payments.

2.2.4. Complaints and Feedback: Provides a platform for complaint submission and feedback.

2.2.5. Amenity Reservations: Allows residents to book community facilities with a payment option.

2.2.6. Security and Monitoring: Implements a Vehicle Monitoring System and displays emergency helpline numbers.

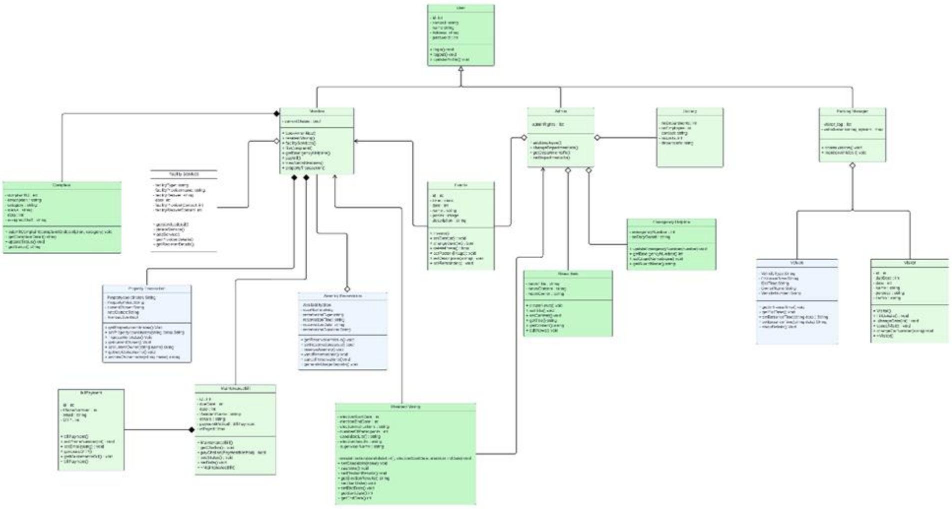
2.2.7. Community Engagement: Supports digital voting, campaign contributions, and streamlined visitor registration.

2.2.8. Property Transactions: Facilitates property listing, buying, selling, and renting with clearance checks.

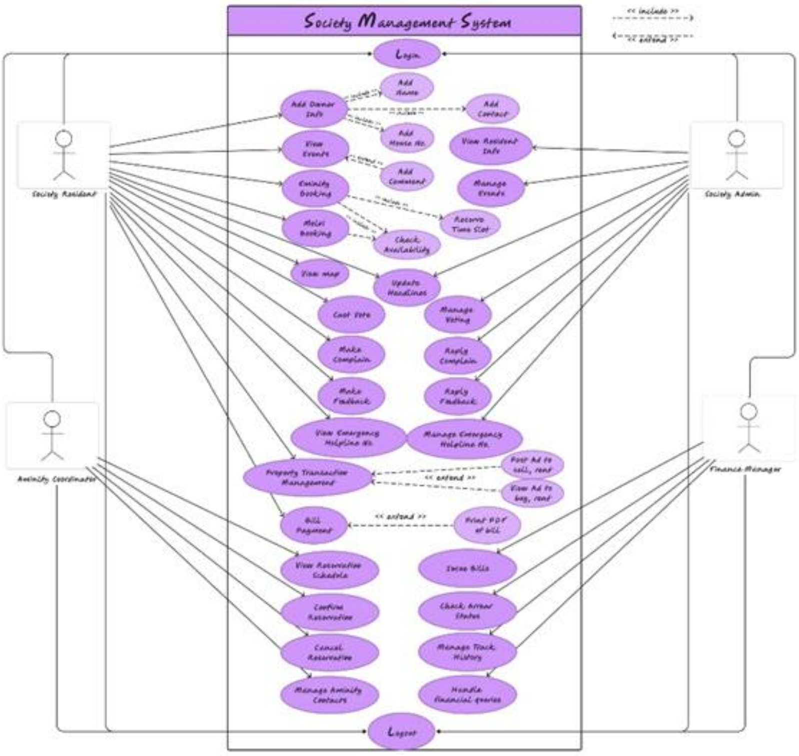
2.2.9. Media and Information: Displays news headlines, community updates, and an interactive map.

2.2.10. Integration with Third-party Services: Interfaces with external providers for payment processing, vehicle tracking, and media services.

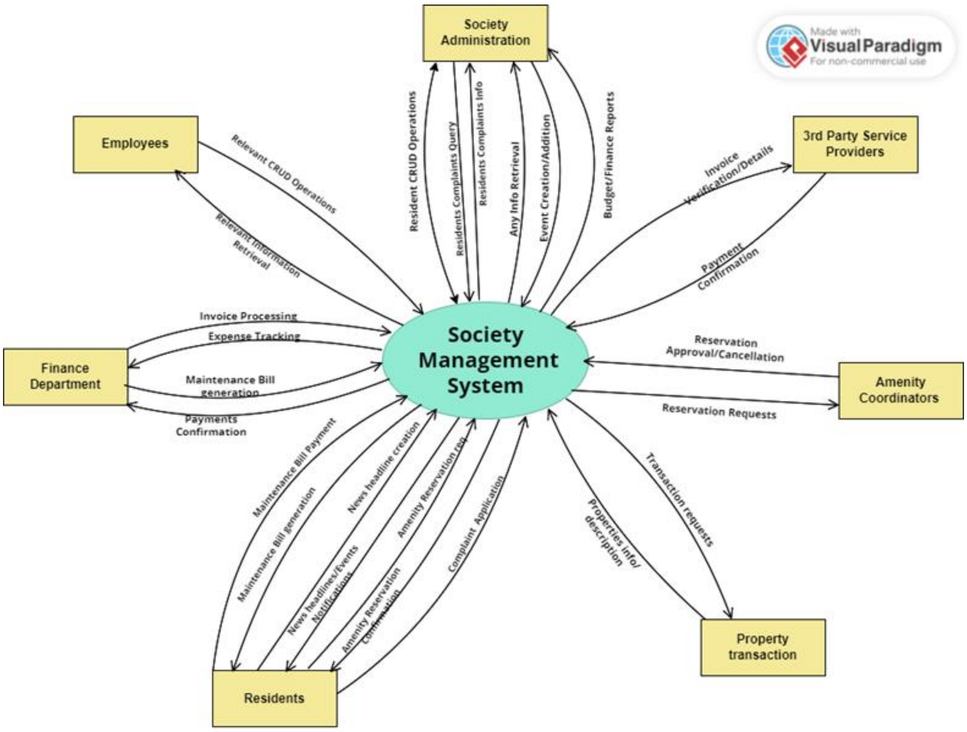
CLASS DIAGRAM



USE CASE DIAGRAM



CONTEXT LEVEL DATA FLOW DIAGRAM



2.3 User Classes and Characteristics

In a Society Management System, various user classes or roles may exist, each with distinct responsibilities and permissions. Here are some typical user classes for such a system:

1. **Administrator/Management Committee**
   - Responsible for overall system administration.
   - Manages user roles and permissions.
   - Access to all features and functionalities of the system.
   - Can generate reports and monitor system health.
   - Residents/Homeowners
   - Access to personal profile information.
   - View and pay bills.
   - Participate in community votes and surveys.
   - Access to event information.
   - Submit complaints and feedback.
2. **Event Coordinator**
   - Manages and updates community event information.
   - Sends event notifications to residents.
   - Tracks attendance and feedback for events.
   - Billing and Finance Personnel:
   - Generates and manages bills for residents.
   - Tracks payment statuses.
   - Generates financial reports for the community.
3. **Maintenance Staff**
   - Receives and resolves maintenance requests submitted by residents.
   - Updates the status of maintenance tasks.
   - Manages and schedules routine maintenance activities.
   - Security Personnel:
   - Access to emergency contact information.
   - Monitors and logs security-related incidents.
   - Communicates with residents regarding security matters.
   - Complaints Resolution Team:
   - Reviews and addresses resident complaints.
   - Updates residents on the status and resolution of their complaints.
4. **Guests/Visitors:**
   - Limited access to community event information.
   - Access to emergency contact information.
   - Restricted ability to participate in votes or submit complaints.
   - Auditors/External Inspectors:
   - Limited access for auditing and compliance purposes.
   - Access to relevant financial and operational reports.

2.4 Operating Environment

The Society Management System (RESIMATE) is meticulously designed for seamless operation in the digital realm, catering to diverse hardware and software environments. It is optimized for standard computing devices, including desktops, laptops, and mobile devices, ensuring efficient performance on modern processors and memory capacities. Resimate exhibits compatibility with major operating systems such as Windows and macOS for desktops/laptops and Android/iOS for mobile devices, staying attuned to the latest versions for heightened security and features. While its compatibility extends to mainstream web browsers like Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge, it currently does not offer native support for Linux systems. The RESIMATE operates independently of specific software applications, relying on web-based functionalities for accessibility. Its adept integration with third-party services remains a hallmark, facilitating collaboration with external systems for payment processing, vehicle tracking, and more. Users can expect a versatile and user-friendly experience, though Linux users may need to access the system through supported web browsers on their preferred platform.

2.5 Design and Implementation Constraints

The development of the Society Management System (Resimate) encounters specific guidelines and limitations that shape its design and implementation. Ensuring the utmost security and privacy of homeowner information is a priority, influencing how we handle and store data. The system's integration with external services, such as payment processing, imposes constraints that we navigate with care. We operate within defined technological and tool boundaries to ensure seamless compatibility. Adhering to corporate and regulatory policies guides our system architecture and coding practices. Given the reliance on internet connectivity, we're implementing robust backup plans to address potential disruptions. Our language choices prioritize web-based functionalities for simplicity and user-friendliness. Additionally, as the client's team will be responsible for long-term maintenance, our design considers ease of future management and updates.

2.6 User Documentation

A detailed video tutorial to be provided upon completion of the project along with a detailed user guide available on the website.

2.7 Assumptions and Dependencies

The development of the Society Management System (Resimate) relies on a set of assumptions and dependencies that shape its trajectory. It is assumed that users will have reliable internet access and personal devices for optimal interaction with Resimate. Successful project outcomes are contingent on continued collaboration from crucial stakeholders, particularly the Finance Department and Security Personnel. Dependencies include the seamless integration of third-party services such as payment processing and vehicle tracking, emphasizing the importance of reliable external components. User proficiency with digital platforms is also presumed for effective system utilization. Any deviations from these assumptions or disruptions in dependencies could impact the Resimate's successful implementation and its alignment with envisioned goals.

3. External Interface Requirements

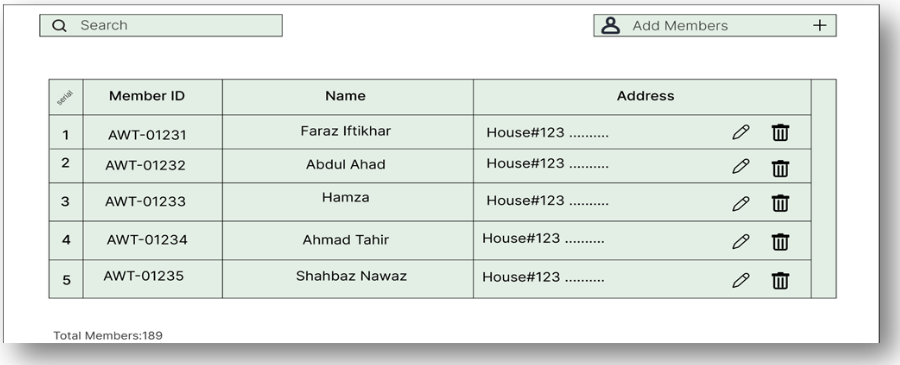
3.1 User Interfaces

The user interface (GUI) will make or break the success of your application. The user interface must be well-planned so that it is consistent and easy for users to comprehend and interact with; hence, it is critical to define GUI standards at the outset of your development project but we have tried our best as in the interface for homeowner contacts are displayed in a directory-style layout with clear fields for names, addresses, phone numbers, and email addresses. The upcoming events interface features event dates and descriptions. Maintenance bill management includes a secure and organized interface for bill details, payment history, and online payment options. The news updates section presents a news feed-style layout with headline previews, article summaries, and sharing options. Complaint submission offers a form-style layout with clear categories and priority levels for submitting concerns. The emergency helplines interface displays easily accessible contact numbers with clear categories such as medical, fire, and police contacts. Each interface follows cohesive screen layout constraints, standard buttons and functions for consistent user interactions, and error message display standards to effectively communicate issues when they arise. This ensures that the user interfaces collectively provide a seamless and efficient experience for the residents interacting with the society management system website.

Here we have shared a few interfaces of the screens.

Members Information Screen:

Here the admin will act as the user.



serial	Member ID	Name	Address	
1	AWT-01231	Faraz Iftikhar	House#123	✎ 🗑
2	AWT-01232	Abdul Ahad	House#123	✎ 🗑
3	AWT-01233	Hamza	House#123	✎ 🗑
4	AWT-01234	Ahmad Tahir	House#123	✎ 🗑
5	AWT-01235	Shahbaz Nawaz	House#123	✎ 🗑

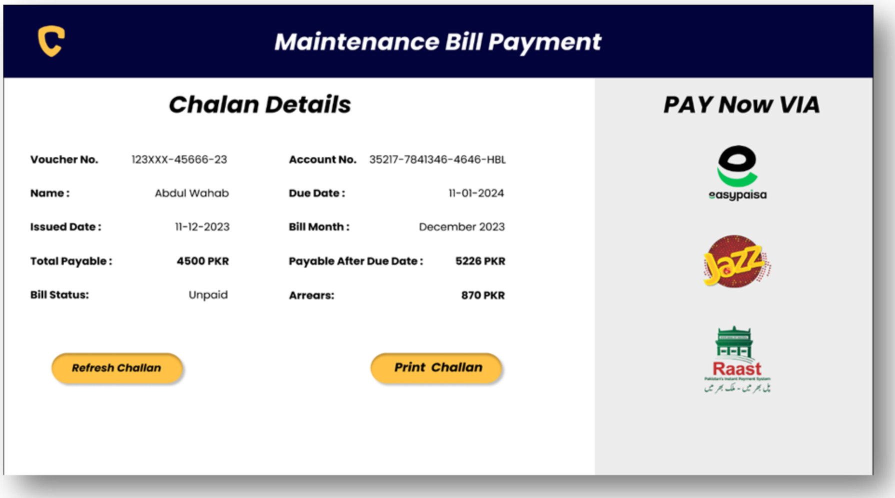
Total Members:189

- **User Interface:** Directory-style display of homeowner's info (names, addresses, Ids).
- **Screen Layout:** Clear and organized for easy navigation.
- **Buttons/Functions:** Add, edit and delete members.

Maintenance Bills Screen

Here the resident will act as the user.

- **User Interface:** Secure and user-friendly for bill viewing and management.
- **Screen Layout:** Organized with details, payment history, and online payment options.
- **Buttons/Functions:** View bill, make a payment, and download challan.
- **Error Messages:** Clear for payment or account access issues.



Maintenance Bill Payment

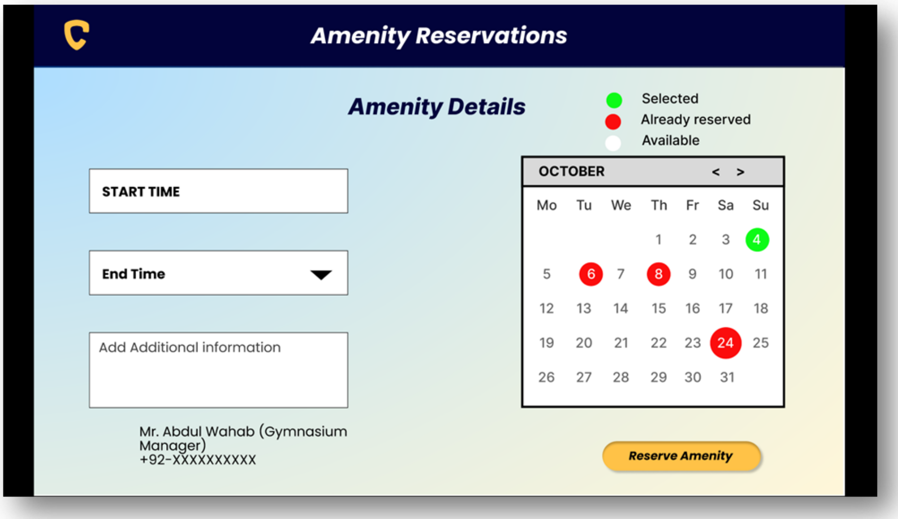
Chalan Details

Voucher No.	123XXX-45666-23	Account No.	35217-7841346-4646-HBL
Name :	Abdul Wahab	Due Date :	11-01-2024
Issued Date :	11-12-2023	Bill Month :	December 2023
Total Payable :	4500 PKR	Payable After Due Date :	5226 PKR
Bill Status:	Unpaid	Arrears:	870 PKR

Amenity Reservations Screen:

- **User Interface:** Efficient for booking community facilities with availability and confirmation.
- **Screen Layout:** Overview, booking calendar, and reservation details (start and end time).
- **Buttons/Functions:** Reserve, select date, and select time.

- **Error Messages:** Dates in the calendar marked with red circles indicate that those dates are not selectable.



Amenity Reservations

Amenity Details

● Selected
● Already reserved
○ Available

START TIME

End Time

Add Additional information

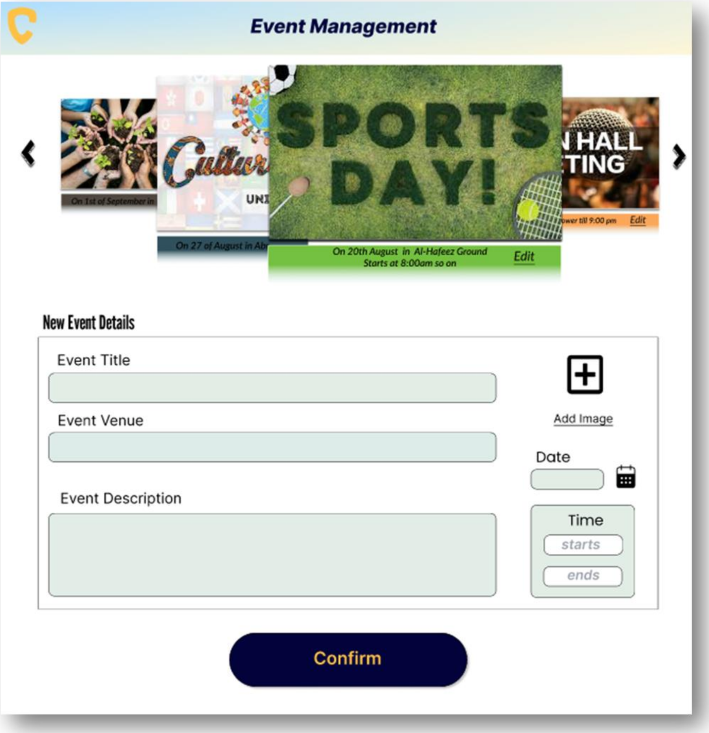
Mr. Abdul Wahab (Gymnasium Manager)
+92-XXXXXXXXXX

OCTOBER < >

Mo	Tu	We	Th	Fr	Sa	Su
			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	

Events Management Screen:

- **User Interface:** Visually appealing display of society events, and details options.
- **Screen Layout:** Flex layout with event dates and descriptions.
- **Buttons/Functions:** Add, Edit and view details of the events.



3.2 Hardware Interfaces

The hardware interfaces for the Society Management System website are meticulously designed for seamless interaction with a diverse range of devices, including desktop computers, laptops, tablets, and smartphones. Ensuring accessibility for a broad user base, these interfaces enable efficient data retrieval and rendering for functionalities such as House Owner Contacts, supporting actions like searching, adding, editing, and exporting contact information.

The Upcoming Events interface ensures effective display and navigation of society events across various devices, supporting control interactions like calendar integration, and event sharing. The Maintenance Bills interface offers secure and user-friendly online viewing and management, presenting organized bill details, payment history, and options for online payments.

For News Headlines, the hardware interface delivers the latest society news updates through visually engaging feeds on diverse devices, allowing users to read full articles, share news, comment, and navigate between news categories. The Complaint Box interface facilitates efficient submission of resident concerns, supporting control interactions for submitting, viewing, and tracking complaints.

Emergency Helplines provide easily accessible contact numbers for immediate assistance, displayed through a straightforward hardware interface with categorized emergency numbers.

Amenity Reservations are facilitated through an efficient hardware interface, offering an overview of facilities, a booking calendar, and confirmation features. The Resident Voting interface supports digital community voting with a clear and accessible ballot-style layout, enabling control interactions for casting votes, viewing history, and accessing election information.

The Vehicle Monitoring System enhances security with a secure and efficient hardware interface for tracking and managing vehicles. It includes a tracking dashboard with real-time locations, history logs, and security alerts, supporting control interactions for tracking, viewing history, and managing access.

The Feedback System utilizes a hardware interface for efficient data submission and presentation of resident suggestions and feedback. It supports control interactions for submitting feedback, viewing history, and managing preferences. Overall, these hardware interfaces prioritize accessibility, efficiency, and user-friendly interactions for an enhanced Society Management System experience.

3.3 Software Interfaces

The Society Management System is intricately connected to various software components to ensure seamless functionality and efficient data management. The system relies on a relational database, specifically using MongoDB to store and retrieve resident information, event details, maintenance bills, and other relevant data. The connection with the database involves the exchange of structured queries and responses, facilitating the storage and retrieval of information critical to the system's operations.

The system leverages standard web development tools like React and libraries such as HTML5, CSS3, and JavaScript, ensuring a responsive and dynamic user interface.

3.4 Communications Interfaces

The Society Management System relies on various communication interfaces to facilitate seamless interactions between users and the system. User notifications, such as event updates and important announcements, ensure timely communication with residents. The system is accessible through standard web browsers, providing a user-friendly interface for residents to access features and information. Network server communications protocols, specifically HTTPS, are employed to secure data transmission between the user's device and the system, safeguarding sensitive information during online transactions and interactions. Electronic forms are utilized for functionalities like the Complaint Box and Feedback System, allowing residents to submit concerns and suggestions efficiently. The system adheres to communication standards such as HTTP for web-based interactions, ensuring compatibility and interoperability. Security measures include encryption protocols for sensitive data, maintaining the confidentiality and integrity of resident information. Data transfer rates are optimized to provide a responsive user experience, and synchronization mechanisms are in place to ensure consistency and real-time updates across different modules of the Society Management System.

4. System Feature

4.1 House Owner Contacts Access (Admin)

4.1.1 Description and Priority

This feature provides access to homeowner contact information. It is of High priority as it facilitates communication and engagement with property owners.

Rating: - Benefit: 9 Penalty: 3 Cost: 5 Risk: 4

4.1.2 Stimulus/Response Sequences

Stimulus 1: Admin selects the "House Owner Contacts" option.

Response 1: The system displays a list of available homeowner contacts.

Stimulus 2: Admin initiates a search for a specific homeowner.

Response 2: The system retrieves and displays the contact information for the specified homeowner.

Stimulus 3: Admin attempts to fun homeowner contact details.

Response 3: The system allows the user to modify and save changes to the homeowner's contact information.

Stimulus 4: Admin adds a new homeowner contact.

Response 4: The system prompts the admin to enter the necessary details Like Owner’s Name, Address, Contact and saves the new information.

Stimulus 5: Admin tries to delete a homeowner contact.

Response 5: The system confirms the deletion and removes the homeowner contact from the list.

4.1.3 Functional Requirements

4.1.3.1 The system shall provide a user interface with an option to manage homeowner contacts.

4.1.3.2 The system shall display a searchable list of homeowner contacts.

4.1.3.3 The system shall allow user to search for homeowners based on specific criteria.

4.1.3.4 The system shall enable users to view and edit existing homeowner contact details.

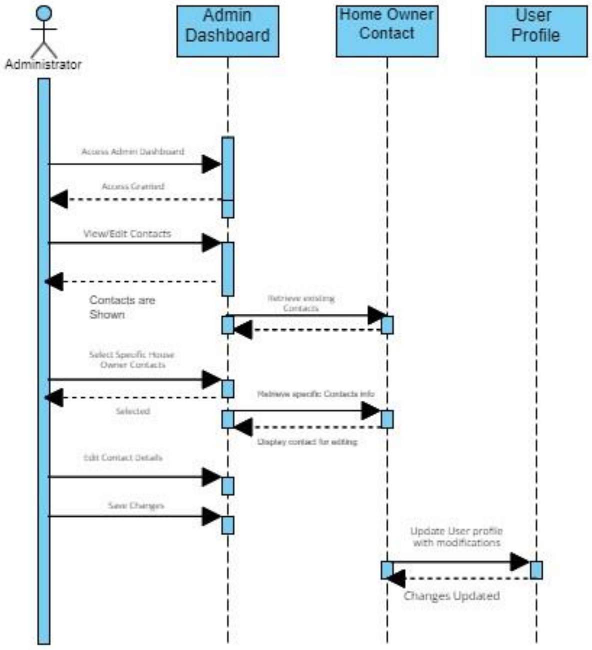
4.1.3.5 The system shall support the addition of new homeowner contacts, requiring necessary information such as name, address, and contact number.

4.1.3.6 The system shall include a confirmation prompt before deleting any homeowner contact.

4.1.3.7 The system shall handle errors gracefully, providing informative messages for invalid inputs or system failures.

ID & Name:	**UC-01 View and edit contact details**	
Created By:	Faraz Iftikhar	**Date created:** 30/11/2023
Primary Actors:	Residents, Administrator	**Secondary Actor:** N/A
Description:	This use case outlines the process of administrator viewing and editing existing homeowner contact details within the Society Management System.	

Trigger:	Admin selects the 'View/Edit Contacts' option in the user profile settings
Pre-Conditions:	Pre 1: Admin must be logged in. Pre 2: Homeowner contact details must be available in the system.
Post-Conditions:	Post 1: Changes to contact details are saved Post 2: The user's profile is updated.
Normal Flow:	1. Administrator navigates to the admin dashboard. 2. Administrator selects 'View/Edit Contacts.' 3. System displays a list of existing homeowners' contact details. 4. Administrator selects a specific homeowner to view or edit their contact information. 5. Administrator edits the contact information as needed. 6. Administrator saves the changes. 7. System updates the user's profile with the modified contact details.
Alternative Flow:	If there are no existing homeowner contact details, the administrator is prompted to add them.
Exceptions:	Technical issues prevent the saving of changes.
Priority:	Medium
Frequency of Use:	Occasionally when administrators need to update homeowner contact information or resolve issues. Multiple times a week.
Business Rules:	BR- **5.5.1**
Other Information:	Ensure data privacy and secure transmission of updated contact details.
Assumptions:	Administrators have the necessary permissions to edit homeowner contact details, and the system stores contact details accurately.



4.2 Upcoming Events (User)

4.2.1 Description and Priority

This feature keeps residents informed about society events. It is of High priority as it enhances community engagement and cohesion.

Rating: - Benefit: 9 Penalty: 2 Cost: 6 Risk: 3

4.2.2 Stimulus/Response Sequences

Stimulus 1: User selects "Upcoming Events."

Response 1: The system displays a list of upcoming society events with details such as date, time, and venue.

Stimulus 2: User clicks on a specific event.

Response 2: The system provides detailed information about the selected event.

Stimulus 3: User submits a comment on an event.

Response 3: The system records the resident's response.

4.2.3 Functional Requirements

4.2.3.1 The system shall present an intuitive interface for residents to access information on upcoming society events.

4.2.3.2 The system shall display a comprehensive list of upcoming events with relevant details.

4.2.3.3 Residents shall be able to view additional details about each event by clicking on the event listing.

4.2.3.4 Residents shall be able to add comments on the society events.

4.2.3.5 The system shall send notifications to residents about upcoming events.

4.3 Upcoming Events (Admin)

4.3.1 Description and Priority

This feature empowers administrators to manage society events by adding event details. It is of High priority as it enables effective event planning and community engagement.

Rating: - Benefit: 9 Penalty: 2 Cost: 6 Risk: 3

4.3.2 Stimulus/Response Sequences

Stimulus 1: Admin selects " Events Management ".

Response 1: The system displays an interface for managing upcoming society events.

Stimulus 2: Admin adds a new event with title, description, image, date, time, and venue.

Response 2: The system records the new event in the database with the provided details.

Stimulus 3: Admin clicks on an existing event for modification.

Response 3: The system allows the admin to edit and save changes to the event details.

4.3.3 Functional Requirements

4.3.3.1 The system shall provide an administrative interface for managing upcoming society events.

4.3.3.2 Admins should be able to add new events with details such as title, description, image, date, time, and venue.

4.3.3.3 The system shall store event details in a database for future reference.

4.3.3.4 Admins shall have the capability to edit and update existing event information.

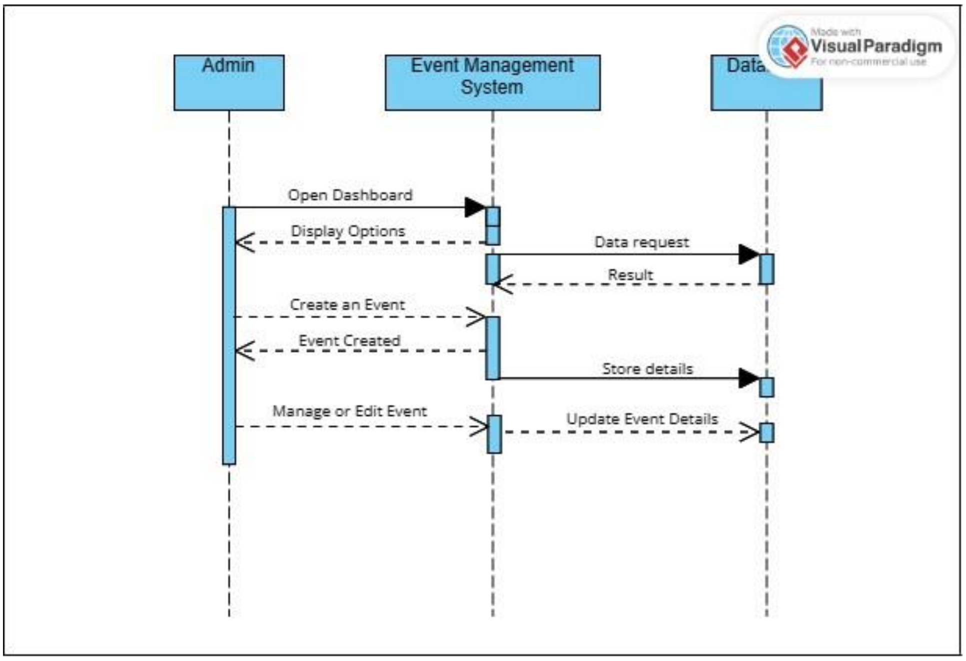
4.3.3.5 The system shall ensure that each event has a unique identifier.

4.3.3.6 The system shall support the addition of comments by residents on society events.

4.3.3.7 The system shall send notifications to residents about newly added events.

ID & Name	**US-02 Event Management by Admin**	
Created By	Muhammad Ahmad	**Date Created:**22/10/23
Primary Actors	Admin	**Secondary Actor:** N/A
Description	This use case outlines the process by which the admin manages events within the society management system.	

Trigger	The admin accesses the society management system to manage events.
Pre-Conditions	**Pre-1:** The admin must be logged into the society management system with appropriate privileges. **Pre-2:** The system events information should be up to date.
Post-Conditions	The event management system is updated with new events, and relevant updates should be added to the upcoming event section.
Normal Flow	1. Admin selects "Event Management" from the dashboard. 2. System displays options to create a new event, edit existing events, or view event details. 3. Admin chooses the option to create a new event. 4. System prompts the admin to provide event details including name, date, time, location, and description. 5. Admin submits the event details. 6. System validates the event details and prompts the admin to approve or neglect the event. 7. Admin approves the event. 8. The system confirms the event approval and adds the event to the upcoming events list.
Alternative Flow	**Flow a.** If the admin chooses to edit an existing event, the system presents the details of the selected event for modification. **Flow b.** If the admin chooses to neglect the event, the system does not add the event to the upcoming events list and notifies the event organizer accordingly.
Exceptions	• If the event details are invalid or incomplete, the system prompts the admin to correct the details before proceeding.
Priority	High
Frequency of Use	Approximately 1-2 times per day.
Business Rules	• Only the admin has the authority to approve or neglect events. • Events must be approved by the admin to be added to the upcoming events list.
Other Information	• The system sends feedback to residents regarding upcoming events upon event approval.
Assumptions	• The admin has the necessary authorization to manage events within the society management system. • The event details provided by the admin are accurate and complete. • The society management system is operational and accessible to the admin.



4.4 Maintenance Bills Management (User)

4.4.1 Description and Priority

This feature enables residents to manage and pay their maintenance bills online. It is of High priority for streamlined community management.

Rating: - Benefit: 8 Penalty: 3 Cost: 7 Risk: 4

4.4.2 Stimulus/Response Sequences

Stimulus 1: User selects "Pay Maintenance Bills."

Response 1: The system presents a breakdown of charges and due dates.

Stimulus 2: User initiates an online payment.

Response 2: The system processes the payment securely and updates the payment status.

Stimulus 3: User clicks on download PDF.

Response 3: The system downloads the maintenance bill copy and saves it into downloads folder.

4.4.3 Functional Requirements

4.4.3.1 The system shall provide a secure platform for residents to access and manage their maintenance bills.

4.4.3.2 Residents shall be able to view an overview of current maintenance bills.

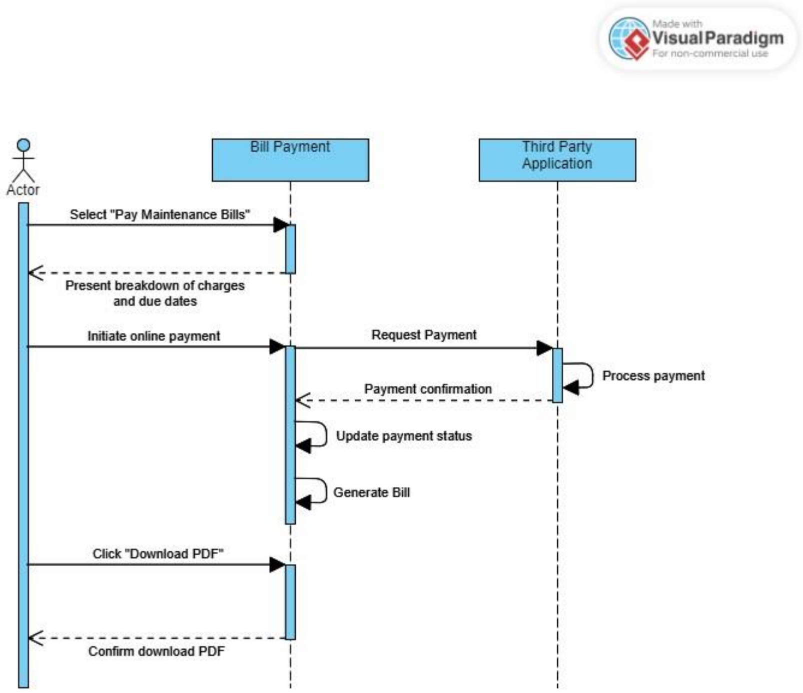
4.4.3.3 The system shall display detailed information for maintenance bill including total payable amount, due date, and previous arrears.

4.4.3.4 Residents shall have the option to make online payments through the system.

4.4.3.5 The system shall generate and send electronic receipts for payments.

4.4.3.6 The system shall be able to download the pdf copy of maintenance bill.

ID & Name:	**UC-03 Paying maintenance bills**	
Created By:	Abdul Ahad	**Date Created:** 30/11/23
Primary Actors:	Residents	**Secondary Actors:** Community Administrator
Description:	This use case outlines the process for residents to manage and pay their maintenance bills online through the society management system (Resimate).	
Trigger:	User selects "Maintenance Bills" option.	
Pre-Conditions:	The user is logged into the society management system.	
Post-Conditions:	The maintenance bill is paid, and the payment status is updated.	
Normal Flow:	1. User selects "Maintenance Bills" option. 2. System presents a breakdown of charges and due dates. 3. User initiates an online payment. 4. System processes the payment securely and updates the payment status. 5. User clicks on download PDF. 6. System downloads the maintenance bill copy and saves it into the downloads folder.	
Alternative Flow:	If the user chooses not to make a payment, they can exit the process without affecting the payment status.	
Exceptions:	If there is an issue processing the payment, the system alerts the user and provides guidance on resolving the issue.	
Priority:	High	
Frequency of Use:	By the end of every month.	
Business Rules:	BR-**5.5.3**	
Other Information:	This feature is essential for streamlined society management, providing residents with a convenient and efficient way to handle their maintenance bills.	
Assumptions:	Users have a reliable internet connection to access and use the online payment system.	



4.5 Maintenance Bills Management (Admin)

4.5.1 Description and Priority

This feature provides administrators with the ability to manage and oversee residents' maintenance bills. It is of High priority for efficient community financial management.

Rating:	Benefit: 8	Penalty: 3	Cost: 7	Risk: 4

4.5.2 Stimulus/Response Sequences

Stimulus 1: Admin selects "Finance Management."

Response 1: The system presents options for Bill generation and view payment history of all residents.

Stimulus 2: Admin clicks on "Bill Generation."

Response 2: The system displays a breakdown of charges, due dates, and any previous arrears.

Stimulus 3: Admin clicks on a payment History for specific resident's detailed information.

Response 3: The system presents an overview of all resident’s current maintenance bills status.

Stimulus 4: Admin updates a resident's bill status.

Response 4: The system processes the status change and updates the payment record.

4.5.3 Functional Requirements

4.5.3.1 The system shall provide an administrative interface for managing residents' maintenance bills.

4.5.3.2 Admins should be able to view an overview of current maintenance bills for all residents.

4.5.3.3 The system shall display detailed information for each maintenance bill, including the total payable amount, due date, and any previous arrears.

4.5.3.4 Admins should have the capability to update the status of a resident's maintenance bill (e.g., paid, unpaid, in arrears).

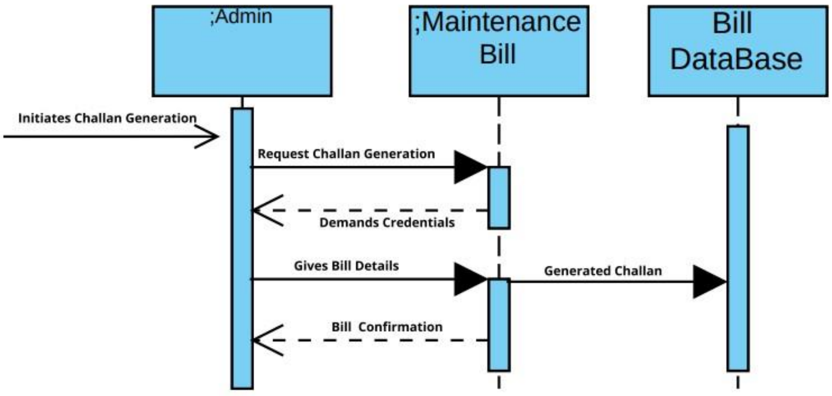
4.5.3.5 The system shall generate and send electronic receipts for payments made by residents.

4.5.3.6 Admins should be able to download a PDF copy of a resident's maintenance bill for record-keeping and auditing purposes.

4.5.3.7 The system shall ensure secure access to maintenance bill information, limiting access only to authorized administrators.

ID & Name	**UC 04 – Generate Maintenance Bill**
Created By	Muhammad Ahmad **Date Created:**22/10/23
Primary Actors	Admin **Secondary Actor:** N/A
Description	This use case describes the process of generating maintenance bills by the admin in the society management system.
Trigger	Admin initiates the process of generating maintenance bills.
Pre-Conditions	**Pre-1:** The admin is logged into the society management system. **Pre-2:** Society parameters for maintenance billing are configured and up to date. **Pre-3:** Resident information, including unit details and billing preferences, is accurate.
Post-Conditions	1. Maintenance bills are successfully generated for selected residents. 2. Residents receive notifications about their maintenance bills.
Normal Flow	1. Admin navigates to the "Billing" section of the admin dashboard. 2. System presents options related to maintenance billing. 3. Admin selects "Generate Bills." 4. System presents parameters for billing cycle, due dates, and unit rates. 5. Admin confirms parameters or adjusts if necessary. 6. System presents a list of eligible residents for the current billing cycle. 7. Admin selects residents for whom bills need to be generated. 8. System generates a preview of maintenance bills. 9. Admin reviews the preview for accuracy. 10. Admin confirms the bill generation. 11. The system generates maintenance bills and sends notifications to residents.
Alternative Flow	**1. Adjust Billing Parameters:** 2. If needed, admin adjusts billing parameters.

	3. Admin confirms changes. **4. Select Residents:** 5. Admin filters or selects specific residents. 6. Admin selects residents for billing. **7. Preview Bills:** 8. System generates a preview of maintenance bills. 9. Admin reviews and makes last-minute changes if necessary. **10. Confirm and Generate:** 11. Admin confirms bill generation. 12. The system generates maintenance bills and sends notifications.
Exceptions	**Invalid Parameters:** ○ If admin attempts to generate bills with invalid parameters, the system provides an error message and prompts for correction. **Billing Preview Issues:** ○ If there are issues with the billing preview, such as discrepancies or errors, the system alerts the admin and provides details for correction.
Priority	High
Frequency of Use	Approximately once a month.
Business Rules	1. Only authorized admins can access and generate maintenance bills. 2. Billing parameters should adhere to society policies and regulations.
Other Information	• The bills generated should include a breakdown of charges, due dates, and any additional information relevant to residents. • The system maintains a record of generated bills for auditing purposes.
Assumptions	1. Residents have provided accurate billing information. 2. The society management system is connected to a secure payment gateway for online bill payments.



4.6 Emergency Helplines (User)

4.6.1 Description and Priority

This feature provides important contact numbers for immediate assistance. It is of High priority for ensuring resident safety.

Rating: - Benefit: 9 Penalty: 1 Cost: 5 Risk: 4

4.6.2 Stimulus/Response Sequences

Stimulus 1: User selects "Emergency Helplines."

Response 1: The system displays a list of important contact numbers for emergencies.

Stimulus 2: User dials an emergency number.

Response 2: The system initiates a call to the selected emergency helpline.

4.6.3 Functional Requirements

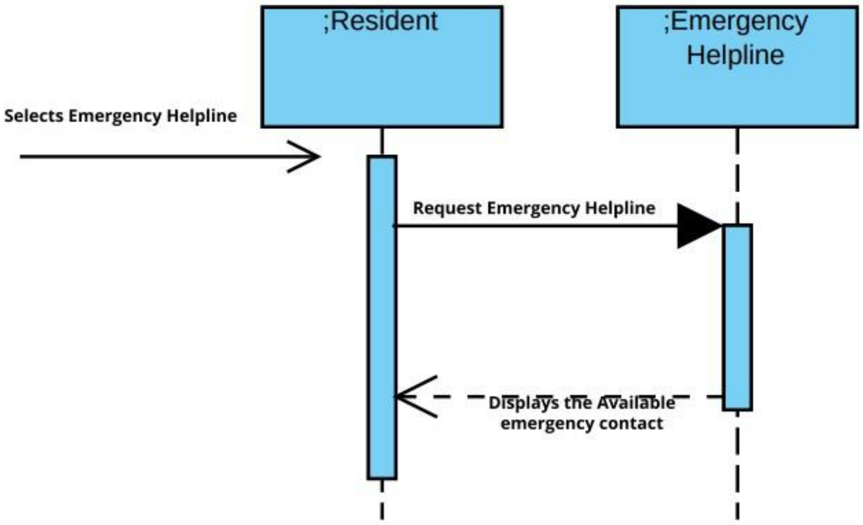
4.6.3.1 The system shall provide a readily accessible list of emergency contact numbers.

4.6.3.2 Users shall be able to initiate calls to emergency helplines directly from the system.

4.6.3.3 The system shall display On Duty Guard name and contact.

ID & Name	**UC-05 Request Emergency Helplines**
Created By	Hamza Naveed **Date Created:**22/10/23
Primary Actors	Resident **Secondary Actor:** Society Administration
Description	This use case enables residents to access and initiate calls to emergency helplines for immediate assistance. The system also provides information about the On Duty Guard.
Trigger	The resident selects "Emergency Helplines."
Pre-Conditions	**Pre-1:** The resident is logged into the system.
Post-Conditions	**Post-1:** The resident has access to a list of emergency contact numbers. **Post-2:** The On Duty Guard's name and contact information are displayed.
Normal Flow	1. User selects "Emergency Helplines." 2. System displays a list of important contact numbers for emergencies. 3. User views the On Duty Guard name and contact information. 4. User dials an emergency number. 5. System initiates a call to the selected emergency helpline.
Alternative Flow	N/A
Exceptions	**System Unavailability:** 1. The system is temporarily unavailable. 2. The user is unable to access the emergency helplines. 3. An error message is displayed, prompting the user to try again later. **No On Duty Guard Information:** 1. Information about the On Duty Guard is not available. 2. The system indicates that the On Duty Guard information is currently inaccessible.

Priority	High
Frequency of Use	Approximately 2-3 times by a resident per month
Business Rules	N/A
Other Information	In case of a scheduling conflict, the system will provide clear and timely notifications to the resident to ensure they are aware of the issue.
Assumptions	N/A


;Resident
;Emergency Helpline
Selects Emergency Helpline
Request Emergency Helpline
Displays the Available emergency contact

4.7 Complaint Box (User)

4.7.1 Description and Priority

This feature allows residents to submit their facing issues. It is of High priority for addressing resident issues effectively.

Rating: - Benefit: 8 Penalty: 3 Cost: 6 Risk: 4

4.7.2 Stimulus/Response Sequences

Stimulus 1: User selects "Complaint Box."

Response 1: The system presents a form for residents to submit their complaints.

Stimulus 2: User submits a complaint.

Response 2: The system acknowledges

Stimulus 3: Administrator reviews and resolves the complaint.

Response 3: The system updates the status.

4.7.3 Functional Requirements

4.7.3.1 The system shall offer an accessible interface for residents to submit complaints.

4.7.3.2 Residents shall be able to add the complaint title and brief description.

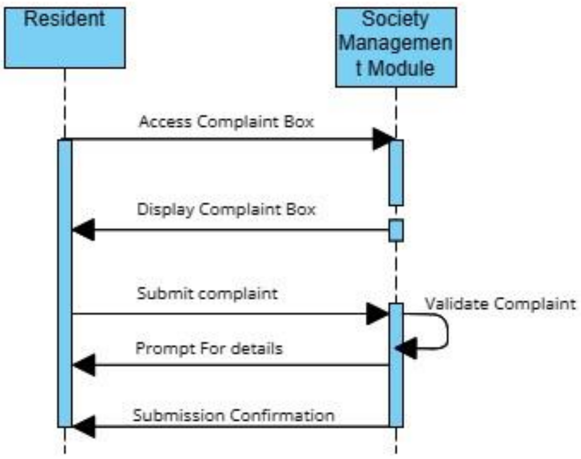
4.7.3.3 Residents shall also be able to add experience through emojis.

4.7.3.4 Administrators shall have access to a dashboard for reviewing and resolving complaints.

4.7.3.5 The system shall update the status of complaint after resolution.

ID & Name	**US-06 Submit Complaint**
Created By	Muhammad Ahmad **Date Created:**22/10/23
Primary Actors	Resident **Secondary Actor:** N/A
Description	This use case describes the process of a resident submitting a complaint through the Complaint Box feature on the society management website.
Trigger	The resident encounters an issue or concern that requires attention from the society management.
Pre-Conditions	**Pre-1: The** resident is registered and logged into the society management system. **Pre-2:** The Complaint Box feature is accessible and functional.
Post-Conditions	1. The resident's complaint is successfully submitted. 2. The complaint is recorded in the system for further resolution.
Normal Flow	1. Resident navigates to the Complaint Box section. 2. Resident selects the option to submit a new complaint. 3. System presents a form prompting the resident to provide details of the complaint, including description and relevant information. 4. Resident fills in the complaint details. 5. Resident submits the complaint. 6. System validates the complaint and records it in the system. 1. The system displays a confirmation message to the resident.
Alternative Flow	If the resident decides not to submit the complaint, they can cancel the process at any point.
Exceptions	• If there is a technical issue preventing complaint submission, an error message is displayed, and the resident is prompted to try again. • If the complaint form is incomplete or invalid, the system prompts the resident to provide the required information.
Priority	High
Frequency of Use	Approximately once a month.
Business Rules	N/A.

Other Information	The Complaint Box feature may include options for residents to track the status of their submitted complaints.
Assumptions	1. Residents have reliable internet access to use the online Complaint Box. 2. The society management team is responsible for addressing and resolving the submitted complaints in a timely manner.



4.8 Amenity Reservations (User)

4.8.1 Description and Priority

Amenity Reservations allow residents to book community facilities. This feature is of High priority as it enhances user convenience and community engagement.

Rating: - Benefit: 8 Penalty: 2 Cost: 5 Risk: 3

4.8.2 Stimulus/Response Sequences

Stimulus 1: User selects the "Amenity Reservations" option.

Response 1: System displays a list of available community facilities.

Stimulus 2: User selects a specific facility for reservation.

Response 2: System prompts the user to choose a date and time for the reservation.

Stimulus 3: User confirms the reservation details.

Response 3: System confirms the reservation and updates the facility schedule.

4.8.3 Functional Requirements

4.8.3.1 The system shall provide a list of available community facilities for reservation

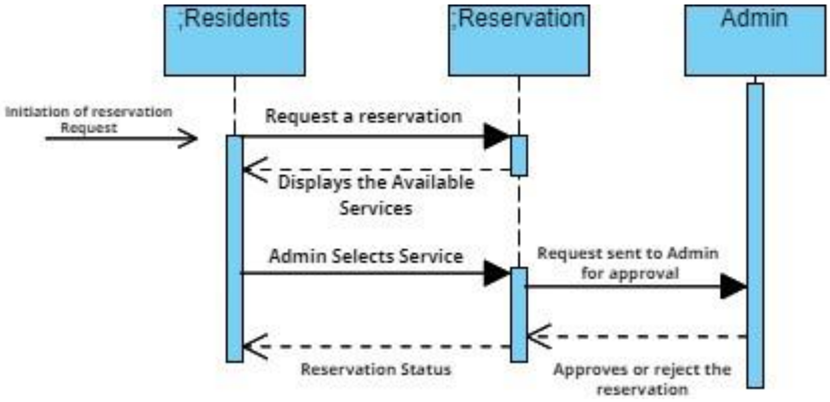
4.8.3.2 The system shall allow users to select a specific facility for reservation.

4.8.3.3 The system shall prompt users to choose a date and time for the reservation.

4.8.3.4 The system shall confirm and update the facility schedule upon user confirmation.

4.8.3.5 The system shall handle and notify users of any scheduling conflicts.

ID & Name	**UC-07 Reserve an Amenity Service**
Created By	Hamza Naveed **Date Created:**22/10/23
Primary Actors	Resident **Secondary Actor:** Society Administration
Description	This use case enables residents to book community facilities. The system provides a user-friendly interface for residents to browse available facilities, select a specific one, choose a date and time for reservation, confirm the booking, and receive notifications.
Trigger	The resident selects the "Amenity Reservations" option in the user interface.
Pre-Conditions	**Pre-1:** The resident is logged into the system. **Pre-2:** Community facility information is up to date in the system.
Post-Conditions	**Post-1:** The facility schedule is updated with the reservation. **Post-2:** The resident receives confirmation of the reservation.
Normal Flow	1. Residents select the "Amenity Reservations" option. 2. The system displays a list of available community facilities. 3. Residents select a specific facility for reservation. 4. System prompts the user to choose a date and time for the reservation. 5. Resident confirms the reservation details. 6. System confirms the reservation and updates the facility schedule.
Alternative Flow	1. System detects a scheduling conflict. 2. System notifies the user of the conflict. 3. User is prompted to choose an alternative date or facility.
Exceptions	**System Unavailability:** 1. The system is temporarily unavailable. 2. System displays an error message. 3. The user is prompted to try again later.
Priority	High
Frequency of Use	Approximately 2-3 times by a resident per month
Business Rules	BR-**5.5.2**
Other Information	In case of a scheduling conflict, the system will provide clear and timely notifications to the resident to ensure they are aware of the issue.
Assumptions	Users are familiar with community facilities. Residents are aware of reservation limitations and agree to terms.


:Residents
:Reservation
Admin
Initiation of reservation Request
Request a reservation
Displays the Available Services
Admin Selects Service
Request sent to Admin for approval
Reservation Status
Approves or reject the reservation

4.9 Administration of Amenity Reservations

4.9.1 Description and Priority

Administration of Amenity Reservations allows administrators to manage bookings for community facilities. This feature is of High priority as it streamlines facility usage and ensures efficient administration.

Rating: - Benefit: 8 Penalty: 2 Cost: 5 Risk: 3

4.9.2 Stimulus/Response Sequences

Stimulus 1: Administrator selects the "Amenity Reservations" option.

Response 1: The system displays an administrative dashboard with an overview of facility bookings.

Stimulus 2: Administrator reviews a specific reservation.

Response 2: The system provides detailed information about the selected reservation.

Stimulus 3: Administrator modifies an existing reservation.

Response 3: The system allows the administrator to adjust reservation details and updates the schedule.

Stimulus 4: Administrator cancels a reservation.

Response 4: The system confirms the cancellation and adjusts the facility schedule accordingly.

4.9.3 Functional Requirements

4.9.3.1 The system shall provide an administrative interface with an option to access amenity reservations.

4.9.3.2 The system shall display an overview of facility bookings on the administrative dashboard.

4.9.3.3 The system shall allow administrators to view detailed information about specific reservations.

4.9.3.4 The system shall enable administrators to modify existing reservations, updating dates, times, and other relevant details.

4.9.3.5 The system shall allow administrators to cancel reservations, provide confirmation prompts and adjust the facility schedule.

ID & Name	**UC 08 - Manage Amenity Services**	
Created By	Muhammad Ahmad	**Date Created:**22/10/23
Primary Actors	Admin	**Secondary Actor:** N/A
Description	This use case outlines the process by which the admin manages amenity services in the society management system.	
Trigger	Admin initiates the process of managing amenity services.	
Pre-Conditions	**Pre-1:** The admin is logged into the society management system. **Pre-2:** Society amenity parameters, such as availability, booking rules, and rates, are configured and up to date.	
Post-Conditions	Amenity services are successfully managed by the admin.	
Normal Flow	1. Admin navigates to the "Amenities" section of the admin dashboard. 2. System presents options related to amenity services management. 3. Admin selects "Manage Amenities. **For Amenity Modification:** 7. Admin selects the option to "Modify Amenity." 4. The system presents a list of existing amenities. 5. Admin selects the amenity to be modified. 6. The system displays current details of the selected amenity. 7. Admin modifies the necessary details and confirms changes. **For Amenity Deletion:** 12. Admin selects the option to "Delete Amenity." 8. The system presents a list of existing amenities. 9. Admin selects the amenity to be deleted. 10. System prompts admin for confirmation. 11. Admin confirms the deletion.	
Alternative Flow	**Alternative Flow:** **12. Amenity Addition:** o If needed, the system may validate the input details to ensure they meet predefined criteria. o Admin corrects any validation errors. **13. Amenity Modification:** o If needed, the system may validate the modified details. o Admin corrects any validation errors.	
Exceptions	1. If there are issues with the input details, the system provides an error message and prompts the admin to correct the details. 2. Invalid Amenity Modification (Step 11b):	

	3. If there are issues with the modified details, the system provides an error message and prompts the admin to correct the details.
Priority	Medium
Frequency of Use	Approximately once a week.
Business Rules	1. Only authorized admins can manage amenity services. 2. Amenity details should adhere to society policies and regulations.
Other Information	• The system should maintain a record of changes made to amenity details for auditing purposes. • Availability schedules should be clearly communicated to residents.
Assumptions	1. Admins have a clear understanding of society's amenity policies. 2. The society management system is connected to a secure database to store and retrieve amenity details.

4.10 Visitor Management

4.10.1 Description and Priority

Visitor Management streamlines the registration of visitors by capturing their purpose, date, and time of visit. This feature is of medium priority as it enhances security and facilitates efficient visitor tracking within the community.

Rating: - Benefit: 6 Penalty: 3 Cost: 4 Risk: 5

4.10.2 Stimulus/Response Sequences

Stimulus 1: User selects the "Visitor Management" option.

Response 1: The system prompts the user to enter visitor details, including purpose, date, and time of visit.

Stimulus 2: User submits visitor information.

Response 2: The system records and confirms the visitor registration.

Stimulus 3: User queries the system for a list of registered visitors on a specific date.

Response 3: The system displays a list of registered visitors for the specified date.

4.10.3 Functional Requirements

4.10.3.1 The system shall provide a user interface with an option for Visitor Management.

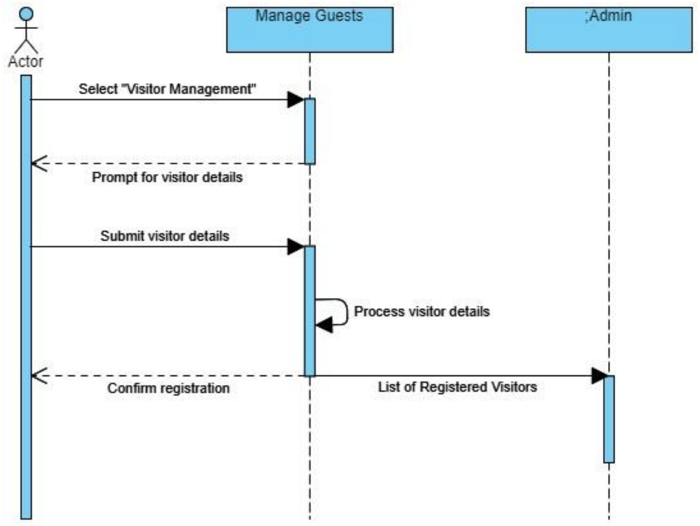
4.10.3.2 The system shall prompt users to enter visitor details, including purpose, date, and time of visit.

4.10.3.3 The system shall record and confirm visitor registration upon submission.

4.10.3.4 The system shall display a list of registered visitors based on specified dates.

4.10.3.5 The system shall handle errors gracefully, providing informative messages for invalid inputs or system failures.

ID & Name:	**UC-09 Manage visitors**
Created By:	Abdul Ahad **Date Created:** 30/11/23
Primary Actors:	Residents **Secondary Actors:** Community Administrator
Description:	This use case outlines the process of registering and tracking visitors within the community through the Visitor Management feature.
Trigger:	User selects the "Visitor Management" option.
Pre-Conditions:	The user is logged into the community management system.
Post-Conditions:	Visitor information is recorded and confirmed in the system.
Normal Flow:	1. User selects the "Visitor Management" option. 2. System prompts the user to enter visitor details, including purpose, date, and time of visit. 3. User submits visitor information. 4. System records and confirms the visitor registration.
Alternative Flow:	**2.1** User submits incomplete visitor information. 1. System prompts the user to provide the missing information. 2. User completes the required fields. 3. System records and confirms the visitor registration. **2.2** System encounters an error during visitor registration. 1. System displays an error message indicating the issue. 2. User follows provided instructions to resolve the issue. 3. System allows the user to reattempt the visitor registration.
Exceptions:	The system returns the user to the main menu or the previous screen, ending the visitor registration process without recording incomplete or inaccurate information.
Priority:	Medium
Frequency of Use:	Regularly, visitors need to be registered within the community.
Business Rules:	BR- **5.5.5**
Other Information:	This feature enhances security and facilitates efficient visitor tracking within the community.
Assumptions:	Users have the necessary permissions to access and utilize the Visitor Management feature.



4.11 Resident Voting (User)

4.11.1 Description and Priority

Resident Voting enables community members to participate in digital voting activities. This feature is of High priority as it promotes democratic participation within the community.

Rating: - Benefit: 9 Penalty: 2 Cost: 6
Risk: 4

4.11.2 Stimulus/Response Sequences

Stimulus 1: User selects the "Resident Voting" option.

Response 1: The system displays the list of ongoing and upcoming community votes.

Stimulus 2: User selects a specific voting event.

Response 2: The system presents detailed information about the selected voting event, including options and voting instructions.

Stimulus 3: User submits their vote.

Response 3: The system records the vote and updates the voting results in real-time.

4.11.3 Functional Requirements

4.11.3.1 The system shall provide a user interface with an option for Resident Voting.

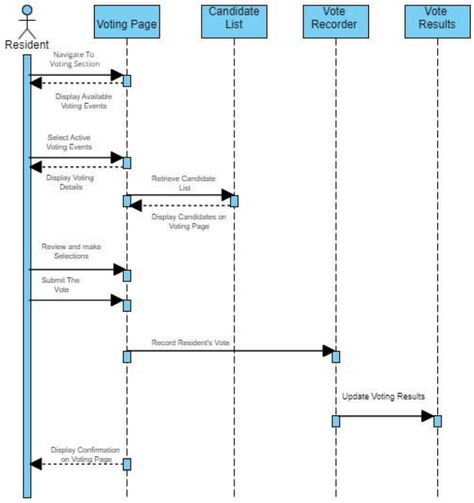
4.11.3.2 The system shall display a list of ongoing and upcoming community votes.

4.11.3.3 The system shall allow users to select a specific voting event to view details.

4.11.3.4 The system shall present detailed information about voting options and instructions for the selected event.

4.11.3.5 The system shall record and update votes in real-time.

ID & Name	**UC-10 Voting in Society**	
Created By	Faraz Iftikhar	**Date created:** 30/11/2023
Primary Actors	Residents, Administrator	**Secondary Actor:** N/A
Description	This use case describes the process of residents casting votes for fellow residents elected for different purposes in the community.	
Trigger	User selects the voting option in the community engagement section.	
Pre-Conditions	Pre 1: Users must be logged in Pre 2: An active voting event must be in progress.	
Post-Conditions	Post 1: The user's vote is recorded Post 2: The voting results are updated.	
Normal Flow	1. User navigates to the community engagement section. 2. User selects the active voting event. 3. User reviews the list of candidates and makes selections. 4. User submits the vote. 5. System records the vote and updates the voting results.	
Alternative Flow	- If there are no active voting events, the system notifies the user. - If the user has already voted, the system prevents multiple votes	
Exceptions	- Technical issues preventing vote submission.	
Priority	High	
Frequency of Use	Once a year.	
Business Rules	Only registered residents can participate in voting.	
Other Information	Ensure the voting process is secure and anonymous	
Assumptions	The system has accurate resident information, and there are no major disruptions during the voting process.	


Resident
Voting Page
Candidate List
Vote Recorder
Vote Results
Navigate To Voting Section
Display Available Voting Events
Select Active Voting Events
Display Voting Details
Retrieve Candidate List
Display Candidates on Voting Page
Review and make Selections
Submit The Vote
Record Resident's Vote
Update Voting Results
Display Confirmation on Voting Page

4.12 Administration of Resident Voting

4.12.1 Description and Priority

Administration of Resident Voting empowers administrators to create polls, view results, add candidates, and discard polls. This feature is of High priority as it enhances the control and management of digital voting activities within the community.

Rating: - Benefit: 8 Penalty: 2 Cost: 7 Risk: 3

4.12.2 Stimulus/Response Sequences

Stimulus 1: Administrator selects the "Resident Voting Administration" option.

Response 1: The system displays an administrative dashboard with options to create polls, view results, add candidates, and discard polls.

Stimulus 2: Administrator creates a new poll.

Response 2: The system prompts the administrator to input poll details, including options and voting period.

Stimulus 3: Administrator views results of a specific poll.

Response 3: The system presents detailed results, including the number of votes for each option.

Stimulus 4: Administrator adds a candidate to an ongoing poll.

Response 4: The system updates the poll with the new candidate information.

Stimulus 5: Administrator discards a poll.

Response 5: The system confirms the poll discard and removes it from the list.

4.12.3 Functional Requirements

4.12.3.1 The system shall provide an administrative interface with options for Resident Voting administration

4.12.3.2 The system shall allow administrators to create new polls, specifying details such as options and voting period.

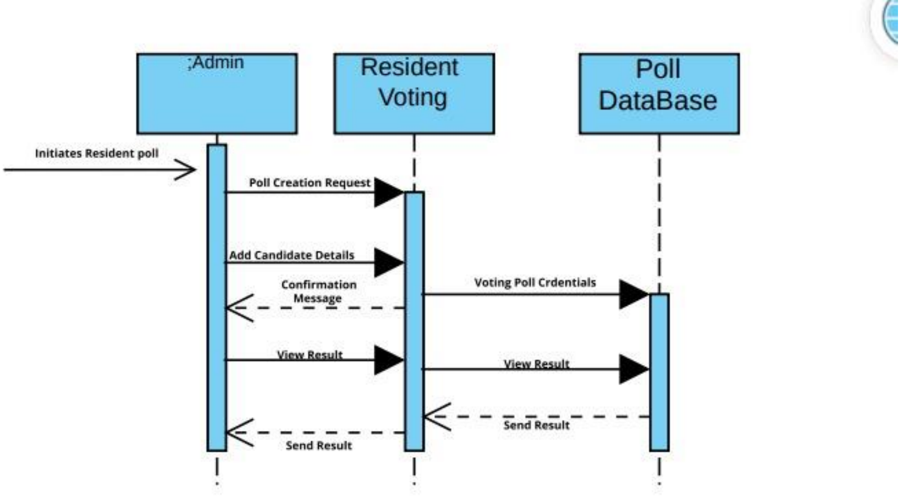
4.12.3.3 The system shall display detailed results for each poll upon administrator request.

4.12.3.4 The system shall enable administrators to add candidates to ongoing polls, updating the poll information.

4.12.3.5 The system shall allow administrators to discard polls, providing confirmation prompts.

ID & Name	**UC-11 Poll creation**
Created By	Hamza Naveed **Date Created:** 22/11/23
Primary Actors	Resident **Secondary Actor:** N/A
Description	This use case empowers the administrator to manage the voting process by adding candidates and creating polls for community elections and decisions
Trigger	The administrator selects the "Voting Management" option in the admin dashboard.
Pre-Conditions	**Pre-1** The Admin is logged into the system.
Post-Conditions	**Post-1** The candidate is added to the system **Post-2** a poll is created for community voting.
Normal Flow	1. Administrator selects the "Voting Management" option in the admin dashboard. 2. System displays an interface for the administrator to add candidates for community elections. 3. Administrator submits candidate details including name, position, and other relevant information. 4. System records the candidate in the system for upcoming community votes. 5. Administrator creates a new poll for community voting, specifying the details such as title, options, and duration. 6. System records the poll and makes it available for community members to participate.

Alternative Flow	4.1 Invalid Candidate Details: Administrator submits incomplete or invalid candidate information. System prompts the administrator to provide the necessary details for a valid candidate. 5.1 Invalid Poll Details: Administrator submits incomplete or invalid poll information. System prompts the administrator to provide the necessary details for a valid poll.
Exceptions	The system encounters a technical failure while managing candidates or creating polls. System displays an error message, indicating the issue.
Priority	Medium
Frequency of Use	Once in a year.
Business Rules	N/A
Other Information	The administrator must ensure that candidate information and poll details adhere to community guidelines and regulations.
Assumptions	The administrator is expected to provide accurate and truthful information when adding candidates and creating polls. The administrator is assumed to be familiar with the process of managing voting elements within the community.



4.13 Property Listings

4.13.1 Description and Priority

Property Listings allow users to post ads for property rentals and sales within the community. This feature is of medium priority as it enhances community members' ability to find suitable properties.

Rating: -	Benefit: 9	Penalty: 2	Cost: 6	Risk: 4

4.13.2 Stimulus/Response Sequences

Stimulus 1: User selects the "Property Listings" option.

Response 1: The system displays an interface for users to create new property listings.

Stimulus 2: User submits a property listing for rent.

Response 2: The system records the listing and makes it available for community members to view.

Stimulus 3: User submits a property listing for sale.

Response 3: The system records the listing and makes it available for community members to view.

Stimulus 4: User searches for properties based on specific criteria.

Response 4: The system displays a list of relevant property listings.

4.13.3 Functional Requirements

4.13.3.1 The system shall provide a user interface with an option for Property Listings.

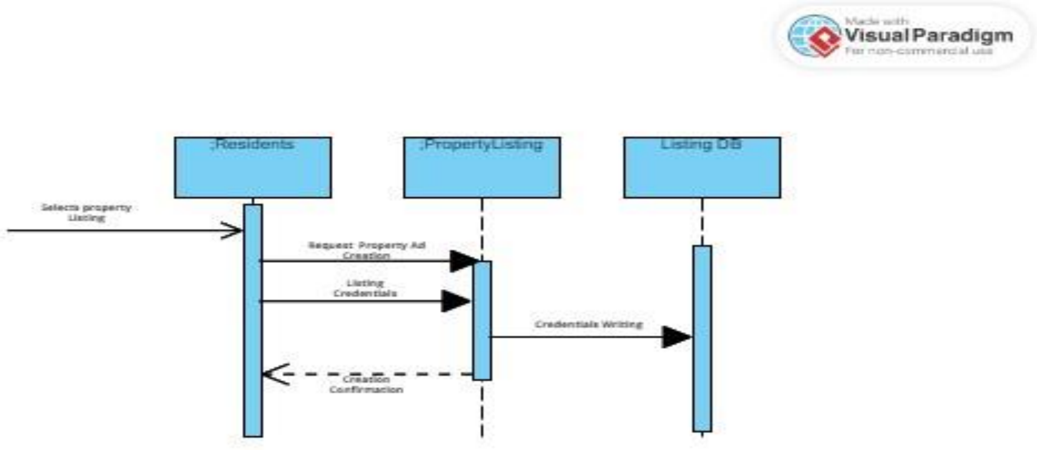
4.13.3.2 The system shall allow users to create new property listings, specifying details such as type (rent/sale), location, and price.

4.13.3.3 The system shall display property listings to community members upon submission.

4.13.3.4 The system shall enable users to search for properties based on criteria such as location, type, and price.

ID & Name	**UC-12 Listing of Property**	
Created By	Hamza Naveed	**Date Created:** 22/11/23
Primary Actors	Resident	**Secondary Actor:** N/A
Description	This use case enables community members to post property listings for rentals and sales, enhancing the community's ability to find suitable properties.	
Trigger	The community member selects the "Property Listings" option.	
Pre-Conditions	**Pre-1** The community member is logged into the system.	
Post-Conditions	**Post-1** The property listing is recorded in the system and made available for community members to view.	
Normal Flow	1. Community members select the "Property Listings" option. 2. System displays an interface for users to create new property listings. 3. Community members submit a property listing for rent or sale. 4. The system records the listing and makes it available for community members to view.	

	5. Community members search for properties based on specific criteria. 6. The system displays a list of relevant property listings.
Alternative Flow	**4.1 System Unavailability:** 1. The system is temporarily unavailable. 2. System displays an error message. 3. The user is prompted to try again later. **4.2 Invalid Listing Submission:** 1. User submits incomplete or invalid listing details. 2. System prompts the user to provide the necessary information for a valid listing.
Exceptions	1. The system encounters a technical failure while displaying property listings. 2. System displays an error message, indicating the issue. 3. Community members are advised to contact technical support or try again later. 4. If the issue persists, community members may miss out on viewing or posting property listings.
Priority	Medium
Frequency of Use	Rarely (once or twice) in a month
Business Rules	N/A
Other Information	**Community Guidelines:** Users are expected to adhere to community guidelines and regulations when creating property listings.
Assumptions	1. Users are expected to provide accurate and truthful information when creating property listings. 2. Users are assumed to be familiar with the process of creating and searching for property listings within the community.



5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 The system shall be able to accommodate a minimum of 5,000 homeowner profiles without significant performance degradation.

5.1.1.1 The performance degradation shall not exceed 1 second when loading 10,000 homeowner profiles simultaneously.

5.1.1.2 During high traffic conditions (usage of more than 10,000 users at once), the system shall manage to maintain optimal load times for user profile data.

5.1.2 The system shall load event listings within 5 seconds of user request.

5.1.2.1 Under normal conditions (users<5,000), the execution time for loading event listings should not exceed 0.5 seconds.

5.1.2.2 Under high traffic conditions (usage of more than 10,000 users at once), event listings should load without exceeding 1 second.

5.1.3 The system shall update the bill status within 1 hour of bill payment.

5.1.3.1 Under normal conditions (users<5,000), the execution time to update a bill status should not exceed 10 minutes.

5.1.3.2 Under high traffic conditions (usage of more than 10,000 users at once), the bill status should be updated within 1 hour.

5.1.4 The system shall accommodate concurrent resident voting by at least 500 residents without performance degradation.

5.1.4.1 Under normal conditions (residents <5,000), the execution time for processing simultaneous votes should not exceed 0.5 seconds.

5.1.4.2 During high traffic conditions (usage of more than 10,000 residents at once), the system shall process concurrent resident votes without performance degradation exceeding 1 second.

5.1.5 The system shall accommodate concurrent bill payments by at least 500 residents without performance degradation.

5.1.5.1 The execution time to process concurrent bill payments should not exceed 0.5 seconds under normal circumstances (users<5,000).

5.1.5.2 In case under heavy traffic (usage of more than 10,000 users at once), the execution time to process concurrent bill payments should not exceed 1 second.

5.2 Safety Requirements

5.2.1 The system shall implement robust data encryption protocols to safeguard sensitive resident information and financial transactions, ensuring protection against unauthorized access or data breaches.

5.2.2 The system shall notify if the user does not save the records.

5.2.2.1 The software shall save to previous checkpoint in case of power failure.

5.2.2.2 In case of no activity for more than 30 mins the system is to automatically log out to reduce load on server.

5.2.3 The system shall be updated once a month with a downtime of 2-3 hours at midnight.

5.2.3.1 The system shall be able to recover from a failure within an hour or less.

5.2.4 The system shall regularly undergo third-party security audits and penetration testing for validating encryption effectiveness and identifying potential vulnerabilities.

5.3 Security Requirements

5.3.1 The system shall implement Role-Based Access Control (RBAC) to restrict access to administrative features, allowing only authorized administrators to manage homeowner contacts, events, and maintenance bills.

5.3.2 The system shall enforce secure login mechanisms, including multi-factor authentication, to enhance the security of user accounts and prevent unauthorized access.

5.3.3 To prevent automated login, CAPTCHA will be used. If there are any issues while accessing the system, error messages will be presented.

5.3.4 In the case of downloadable files, such as PDF copies of maintenance bills, the system shall implement secure file handling to prevent unauthorized access or tampering.

5.3.5 The system shall implement secure session management practices, including automatic logout after a defined period of inactivity to reduce the risk of unauthorized access.

5.4 Software Quality Requirements

5.4.1 Robustness:

- The system shall handle unexpected inputs or scenarios without crashing, maintaining a failure rate of less than 1% in stress testing scenarios.

5.4.2 Usability:

- The system interface shall be intuitive, ensuring a task completion success rate of at least 90% for common user actions.

5.4.3 Adaptability:

- The system shall adapt to changes in society's policies or structure with minimal disruption, maintaining a documented change acceptance rate of 95%.

5.4.4 Availability:

- The system shall be available for use 99% of the time, excluding scheduled maintenance periods.

5.4.5 Correctness:

- The system shall ensure accurate representation and manipulation of data throughout its lifecycle.

5.4.6 Maintainability:

- The system codebase shall have a maintainability index of at least 80, as measured by static code analysis tools.

5.4.7 Interoperability:

- The system shall seamlessly integrate with commonly used third-party tools and platforms, achieving successful integration for at least 95% of tested cases.

5.4.8 Reliability:

- The system shall operate without critical failures for a minimum of 30 consecutive days under normal operating conditions.

5.4.9 Portability:

- The system shall be designed to run on multiple operating systems, ensuring compatibility with commonly used platforms (e.g., Windows, Linux, macOS).

5.4.10 Efficiency:

- The system shall optimize processes and workflows within each feature for efficiency, ensuring that user wait time and resource usage are minimized to less than 1 second.

5.5 BUSINESS RULES

5.5.1 Only administrators are authorized to add, edit, or delete homeowner contacts.

5.5.2 Amenity Reservations will only be approved and processed if all outstanding dues for the respective property are cleared.

5.5.3 An incremental penalty of 3% will be applied to outstanding dues every 10 days, starting from the due date.

5.5.4 Complaint submissions are reviewed by the relevant department or responsible party within 3 business days.

5.5.5 Visitor information must include purpose, date, and time of visit for accurate tracking.