**Software Engineering**

**Housing Society Management System**

# 

# **Group Members:** Alian Anwar 21i-0730 Ammar Arshad 21i-0456 Muhammad Raza 21i-0511

# 

# Delieverable #1

# Submission Date: April 7,2024

**Software Requirements**

**Specification**

for

**Housing Society management System**

Version 1.0 approved   
  
April 7th, 2024

# 

# **Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction**

1.1 Purpose

1.2 Document Conventions

1.3 Intended Audience and Reading Suggestions

1.4 Product Scope

1.5 References

**2. Overall Description**

2.1 Product Perspective

2.2 Product Functions

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 User Documentation

2.7 Assumptions and Dependencies

**3. External Interface Requirements**

3.1 User Interfaces

3.2 Hardware Interfaces

3.3 Software Interfaces

3.4 Communications Interfaces

**4. System Features**

4.1 Resident Management

4.2 Financial Management

4.2 Facility Booking

4.2 Complaint Handling

4.2 Communication

**5. Other Nonfunctional Requirements**

5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software Quality Attributes

5.5 Business Rules

**6. Diagrams**

6.1 Use Case Diagram

6.2 Activity Diagram

6.3 Sequence Diagram

**Appendix A**: Glossary

**Appendix B**: Analysis Models

**Appendix C**: To Be Determined List

# **Revision History**

| Name | Date | Reason For Changes | Version |

**1. Introduction**

**1.1 Purpose**

The purpose of this Software Requirements Specification (SRS) document is to outline the functional and non-functional requirements of the Housing Society Management System. This document serves as a guideline for the development team to understand the scope of the project, its objectives, and the features to be implemented. The current version of this document is Version 1.0.

**1.2 Document Conventions**

This SRS follows standard formatting conventions to ensure clarity and consistency throughout the document. Requirements are prioritized based on their importance and relevance to the overall system. Each requirement statement is accompanied by its priority level to guide the development process.

**1.3 Intended Audience and Reading Suggestions**

This document is intended for various stakeholders involved in the development and management of the Housing Society Management System. This includes developers, project managers, marketing staff, users, testers, and documentation writers.

The document is organized into sections, starting with an overview of the system and proceeding to detailed descriptions of requirements and features. It is recommended to begin with the overview sections and then delve into the sections most pertinent to the reader's role.

**1.4 Product Scope**

The Housing Society Management System is designed to streamline the management of housing societies by automating various tasks such as maintenance fee management, visitor tracking, facility booking, complaint handling, and communication among residents and management.

The system aims to improve efficiency, transparency, and communication within housing societies, ultimately enhancing the living experience for residents. It aligns with the corporate goals of promoting community engagement and improving operational efficiency.

**1.5 References**

- "Community Management Software Specifications," by HousingTech Solutions, Version 2.5, December 2023.

- "Best Practices for Housing Society Management Systems," by Housing Management Association, 2022.

**2. Overall Description**

**2.1 Product Perspective**

The Housing Society Management System (HSMS) is a standalone product designed to streamline the management of housing societies. It serves as a centralized platform for managing various aspects of a housing society, including resident information, financial transactions, facility bookings, and communication. While the HSMS is a standalone product, it may interface with external systems such as accounting software for financial data integration.

**2.2 Product Functions**

* **Resident management:** Add, update, and delete resident profiles, including contact information and unit details.
* **Financial management:** Track maintenance fees, generate invoices, and manage payments.
* **Facility booking**: Allow residents to book community facilities such as clubhouses or swimming pools.
* **Complaint handling:** Enable residents to submit and track maintenance or facility-related complaints.
* **Communication:** Facilitate communication between residents and management through announcements, notices, and messaging features.

**2.3 User Classes and Characteristic**s

* **Residents:** Regular users who interact with the system to access information, make bookings, and submit complaints.
* **Management Committee:** Administrators responsible for overseeing the management of the housing society, including managing resident data, financial transactions, and facility bookings.
* **Maintenance Staff:** Users responsible for addressing maintenance requests and handling facility bookings on-site.

Each user class may have varying levels of technical expertise and access privileges within the system.

**2.4 Operating Environment**

The HSMS will operate in a standard computing environment, including:

* **Hardware Platform:** Compatible with common hardware configurations, including desktop computers, laptops, and mobile devices.
* **Operating System:** Compatible with Windows, macOS, and Linux operating systems.
* **Software Components:** Requires a web browser for accessing the system's web-based interface.

**2.5 Design and Implementation Constraint**s

* **Compliance with Regulatory Policies:** The system must adhere to relevant regulations governing data privacy and financial transactions.
* **Integration with External Systems:** The HSMS may need to integrate with accounting software for financial data synchronization.
* **Security Considerations:** Implementation must adhere to industry-standard security practices to safeguard resident data and financial transactions.

**2.6 User Documentation**

User documentation components will include user manuals, on-line help resources, and tutorials. Documentation will be provided in digital formats accessible through the system interface.

**2.7 Assumptions and Dependencies**

**Assumptions:**

* Assumed availability of internet connectivity for accessing the system.
* Third-party libraries and frameworks may be used for certain functionalities.
* Availability of necessary hardware and software resources for system operation.

**Dependencies:**

* Integration with accounting software for financial data synchronization.
* Availability of third-party services for features such as messaging and notifications.

**3. External Interface Requirements**

**3.1 User Interfaces**

The Housing Society Management System (HSMS) will feature an intuitive user interface accessible through web browsers on desktop and mobile devices. The interface will adhere to modern design principles and include the following characteristics:

* Clean and user-friendly layout for easy navigation.
* Responsive design to ensure compatibility across various screen sizes and resolutions.
* Standard buttons for common functions such as submit, cancel, save, and back.
* Consistent error message display standards for effective user feedback.
* Help documentation accessible within the system for user assistance.
* Sample screen images and layout constraints will be documented in the User Interface Specification.

**3.2 Hardware Interfaces**

The HSMS will interact with standard hardware components, including:

* Desktop computers
* Laptops
* Mobile devices

The system will not have direct physical interactions with hardware components. It will operate in a hardware-agnostic manner, allowing users to access the system from their preferred devices.

**3.3 Software Interfaces**

The HSMS will integrate with the following software components:

* Database Management System (DBMS): The system will interact with a relational database for storing and retrieving data related to residents, financial transactions, and facility bookings. The specific DBMS and version will be determined during the implementation phase.
* Operating System: The system will be compatible with various operating systems, including Windows, macOS, and Linux.
* Web Browsers: The user interface will be accessible through standard web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge.
* Third-Party Services: Integration with third-party services may be required for features such as messaging and notifications. The specific services and protocols will be determined based on project requirements.
* Data sharing between software components will be facilitated through APIs (Application Programming Interfaces) to ensure seamless communication and interoperability. Detailed API protocols will be documented separately.

**3.4 Communications Interfaces**

Communication functions required by the HSMS include:

* Web server communications protocols: The system will communicate with web servers using HTTP (Hypertext Transfer Protocol) for data exchange.
* Email notifications: The system may send email notifications to residents for important updates such as payment reminders or facility booking confirmations.
* Messaging services: Integration with messaging services such as SMS (Short Message Service) or push notifications may be implemented for instant communication with residents.
* Communication security: Secure communication protocols such as HTTPS (Hypertext Transfer Protocol Secure) will be used to encrypt data transmission and ensure communication security.
* Data transfer rates and synchronization mechanisms will be optimized for efficient communication between the system components.

**4 System Features**

**4.1 Resident Management**

**4.1.1 Description and Priority**

This feature allows administrators to manage resident profiles within the system. It is of high priority as it forms the foundation for other functionalities such as financial management and communication.

**Priority:** High

**4.1.2 Stimulus/Response Sequences**

* User navigates to the resident management section.
* System displays a list of existing resident profiles.
* User selects an option to add, update, or delete a resident profile.
* System prompts user for necessary information and updates the database accordingly.

**4.1.3 Functional Requirements**

* REQ-1: The system shall allow administrators to add new resident profiles, capturing details such as name, contact information, and unit number.
* REQ-2: The system shall allow administrators to update existing resident profiles, enabling modifications to contact information or unit details.
* REQ-3: The system shall allow administrators to delete resident profiles, ensuring appropriate data integrity measures are in place.
* REQ-4: The system shall validate user inputs to ensure accuracy and completeness of resident information.
* REQ-5: In case of invalid inputs, the system shall provide appropriate error messages and guide the user to correct them.

**4.2 Financial Management**

**4.2.1 Description and Priority**

This feature enables the management of financial transactions within the housing society, including maintenance fee management and invoice generation. It is of medium priority as it is crucial for financial transparency and accountability.

**Priority:** Medium

**4.2.2 Stimulus/Response Sequences**

* User accesses the financial management module.
* System displays options for maintenance fee management and invoice generation.
* User selects an option and provides necessary inputs.
* System processes the request and updates financial records accordingly.

**4.2.3 Functional Requirements**

* REQ-6: The system shall allow administrators to record maintenance fees for each resident, specifying the amount and due date.
* REQ-7: The system shall generate invoices for residents based on maintenance fee records, providing itemized details and due dates.
* REQ-8: The system shall track payment statuses for invoices, marking them as paid upon receipt of payment.
* REQ-9: The system shall generate reports summarizing financial transactions, including overdue payments and outstanding balances.
* REQ-10: The system shall ensure secure storage and retrieval of financial data, implementing encryption and access control mechanisms as necessary.

**4.3 Facility Booking**

**4.3.1 Description and Priority**

This feature allows residents to book community facilities such as clubhouses, swimming pools, or event spaces. It is of high priority as it enhances resident engagement and facilitates efficient facility management.

**Priority:** High

**4.3.2 Stimulus/Response Sequences**

* User navigates to the facility booking section.
* System displays available facilities along with their schedules.
* User selects a facility and specifies the desired date and time for booking.
* System checks availability and confirms or denies the booking request.
* If confirmed, system updates the booking schedule and notifies the user.

**4.3.3 Functional Requirements**

* REQ-11: The system shall display a list of available facilities along with their booking schedules.
* REQ-12: The system shall allow residents to select a facility and specify booking details such as date and time.
* REQ-13: The system shall check the availability of the selected facility for the specified date and time.
* REQ-14: If the facility is available, the system shall confirm the booking and update the booking schedule.
* REQ-15: If the facility is not available, the system shall notify the user and provide alternative booking options if available.

**4.4 Complaint Handling**

**4.4.1 Description and Priority**

This feature enables residents to submit maintenance or facility-related complaints to the management committee. It is of medium priority as it contributes to resident satisfaction and community well-being.

Priority: Medium

**4.4.2 Stimulus/Response Sequences**

* User accesses the complaint submission section.
* System presents a form for entering complaint details.
* User provides information about the nature of the complaint and submits it.
* System acknowledges receipt of the complaint and assigns it a tracking number.
* Management committee reviews and addresses the complaint as per established procedures.

**4.4.3 Functional Requirements**

* REQ-16: The system shall provide a user-friendly interface for residents to submit complaints, including options to categorize the nature of the complaint.
* REQ-17: Upon submission, the system shall assign a unique tracking number to each complaint for reference and tracking purposes.
* REQ-18: The system shall notify the management committee of new complaints and provide tools for managing and tracking complaint resolution progress.
* REQ-19: The system shall maintain a history of all complaints, including status updates and resolution details, for auditing and reporting purposes.

**4.5 Communication**

**4.5.1 Description and Priority**

This feature facilitates communication between residents and the management committee through announcements, notices, and messaging features. It is of high priority as effective communication is essential for community engagement and coordination.

Priority: High

**4.5.2 Stimulus/Response Sequences**

* User accesses the communication module.
* System displays options for sending announcements, posting notices, or initiating messages.
* User selects the desired communication type and provides necessary content.
* System delivers the communication to the intended recipients and logs the activity.

**4.5.3 Functional Requirements**

* REQ-20: The system shall provide tools for administrators to create and send announcements to all residents or specific groups.
* REQ-21: The system shall allow administrators to post notices regarding upcoming events, maintenance schedules, or other important information.
* REQ-22: The system shall support direct messaging functionality, enabling residents to communicate with each other or the management committee privately.
* REQ-23: The system shall maintain a communication log for tracking sent and received messages, including timestamps and sender/receiver information.
* REQ-24: The system shall provide notification mechanisms to alert users of new announcements, notices, or messages.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

* The system should respond to user interactions within 2 seconds under normal load conditions to ensure a smooth user experience.
* The system should be able to handle concurrent requests from multiple users without significant degradation in performance, with a minimum capacity of 100 concurrent users.
* The database queries should execute within 1 second for common operations such as resident profile lookup or financial transaction retrieval.

**5.2 Safety Requirements**

* The system should implement access controls to ensure that only authorized users can perform sensitive actions such as financial transactions or resident data modifications.
* The system should regularly back up critical data to prevent data loss in case of system failures or disasters.
* In case of critical errors, the system should provide graceful error handling to prevent data corruption or system instability.

**5.3 Security Requirements**

* User authentication: The system should implement secure user authentication mechanisms such as password-based authentication or two-factor authentication to prevent unauthorized access.
* Data encryption: Sensitive data such as resident information or financial transactions should be encrypted during transmission and storage to protect against unauthorized access.
* Role-based access control: The system should enforce role-based access control to restrict users' access to specific functionalities based on their roles and privileges.

**5.4 Software Quality Attributes**

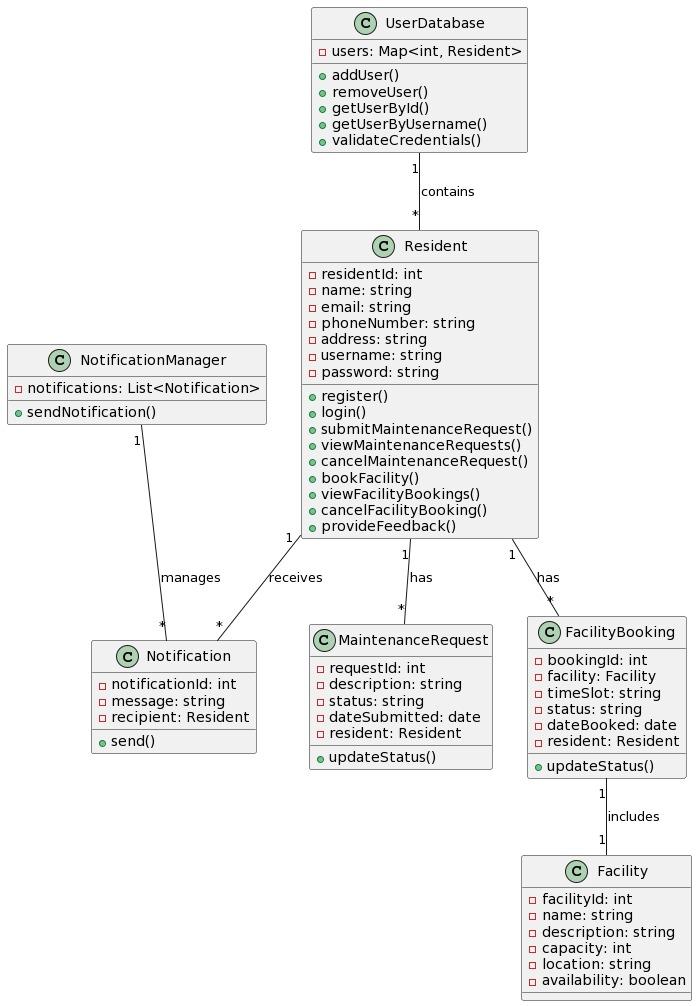
* Maintainability: The system should be designed with modular and well-documented code to facilitate ease of maintenance and future enhancements.
* Reliability: The system should be highly reliable, with a target uptime of at least 99.9%, to ensure uninterrupted access for users.
* Usability: The system should have an intuitive user interface and provide clear instructions to users to minimize learning curves and enhance usability.

**5.5 Business Rules**

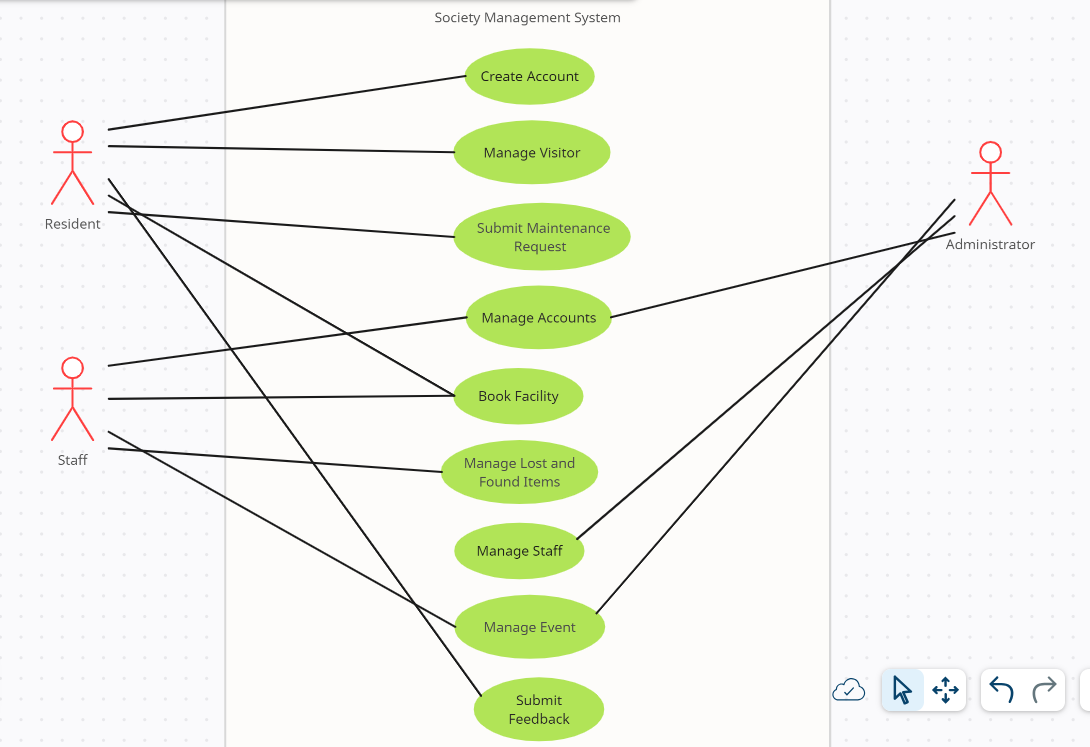
* Only administrators or authorized personnel can add or delete resident profiles.
* Residents can only book facilities for a maximum duration of 3 hours per booking session.
* Financial transactions above a certain threshold require approval from the management committee.
* Complaints should be addressed within 48 hours of submission, with regular updates provided to residents on the resolution status.

**6 Diagrams**

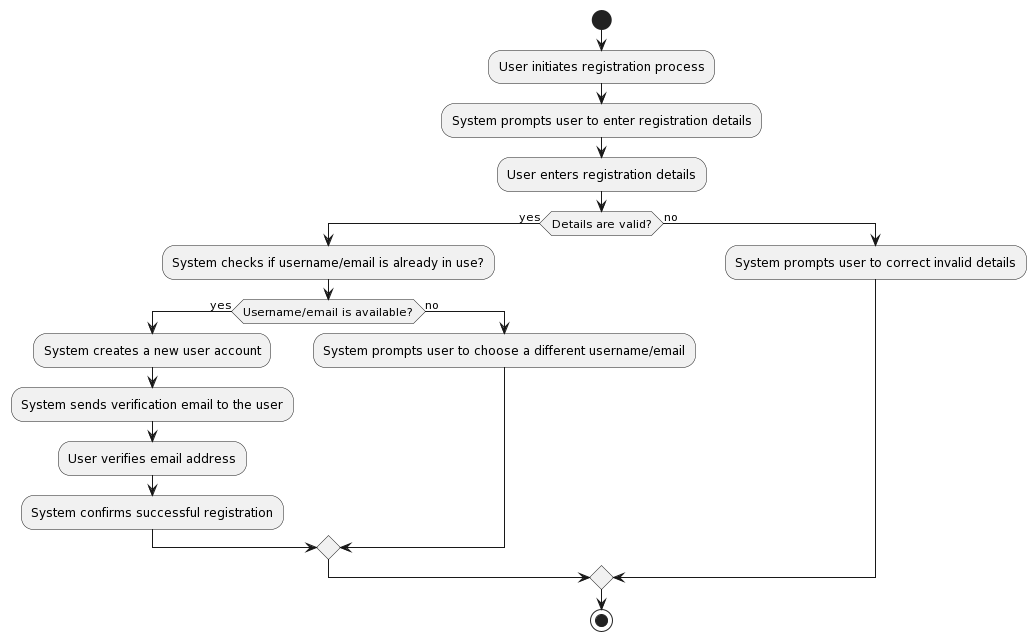
**6.1 Class Diagram**

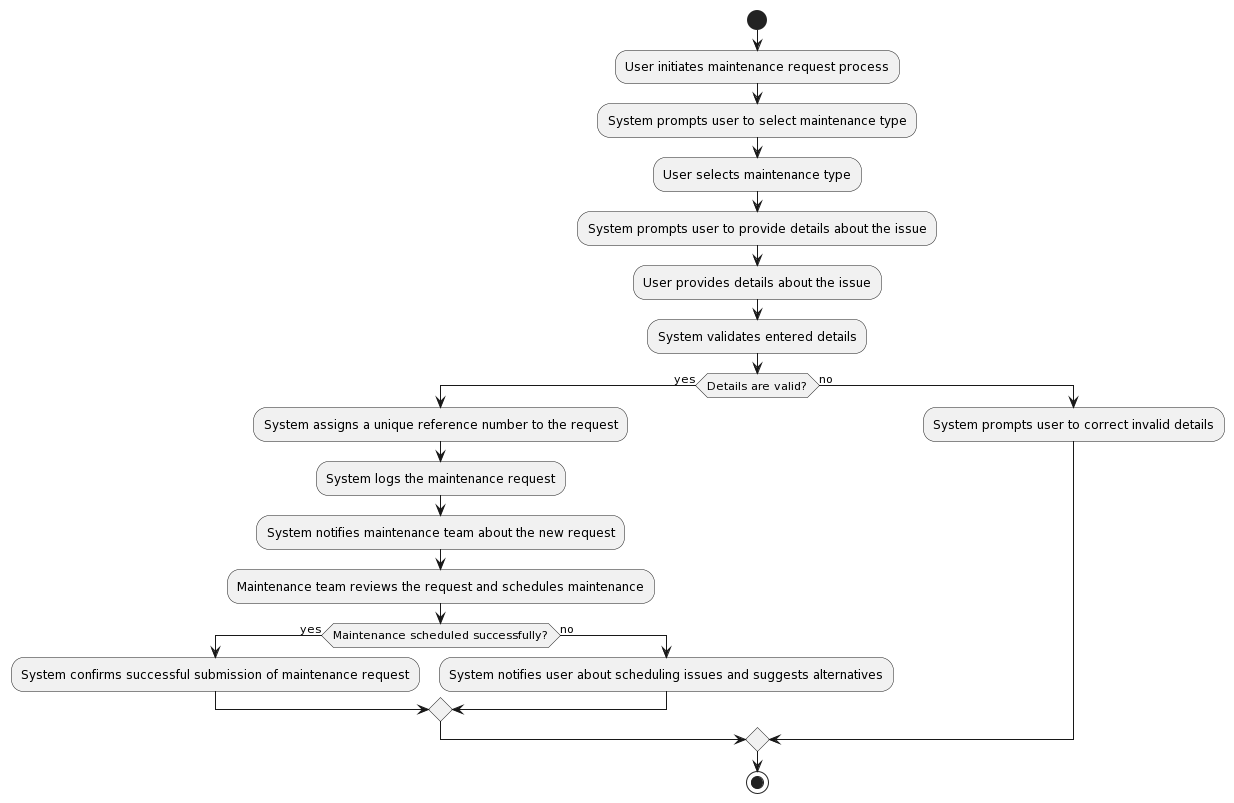
****

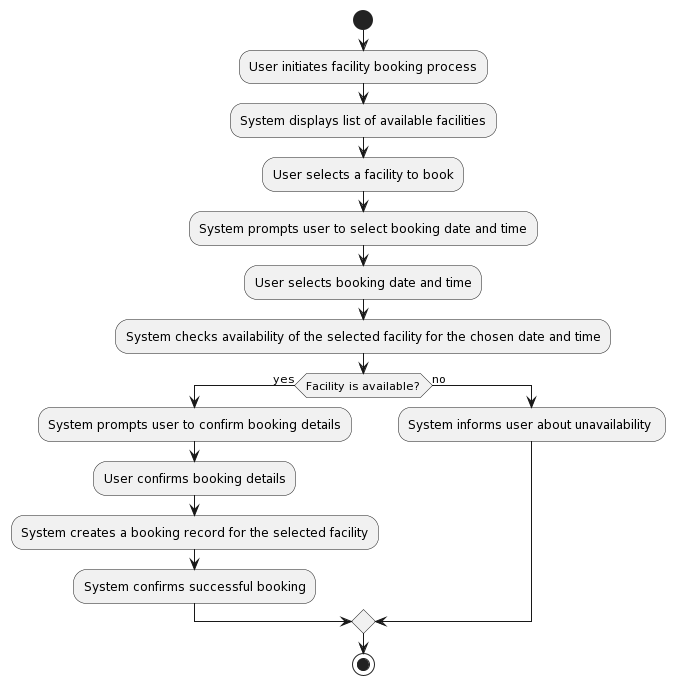
**6.2 Use Case Diagram**

****

**6.3 Activity Diagram**

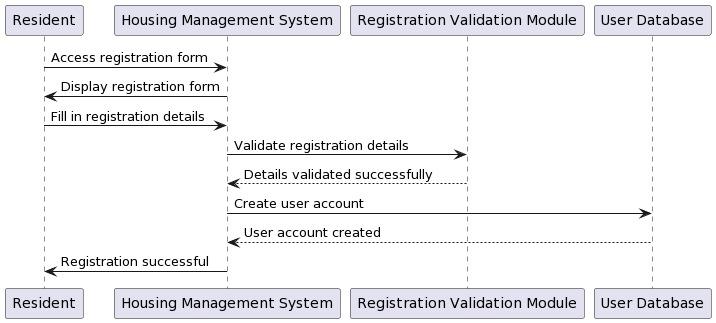
* **User Registration:  
  **
* **Maintenance Request Submission:**

****

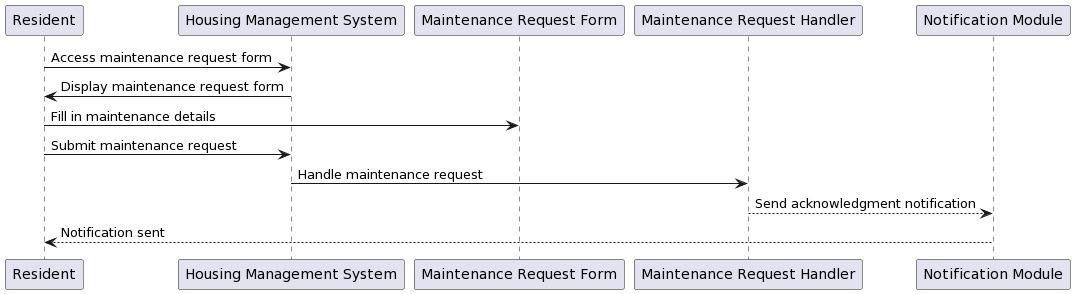
* **Facility Booking:  
  **

**6.4 Sequence Diagram**

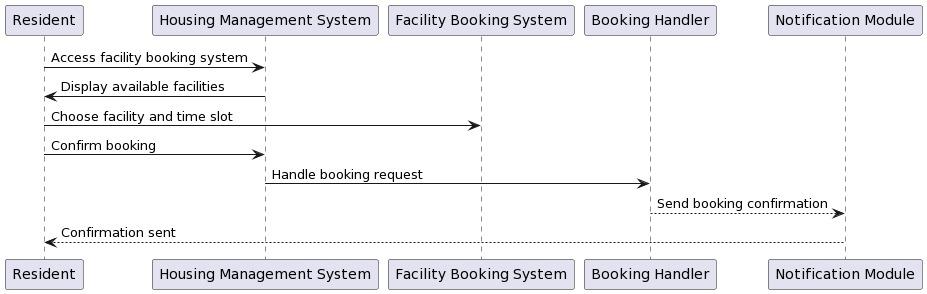
* **User Registration:**

****

* **Maintenance Request Submission:**

****

* **Facility Booking:**

****

**Appendix A: Glossary**

* HSMS: Housing Society Management System
* DBMS: Database Management System
* HTTP: Hypertext Transfer Protocol
* HTTPS: Hypertext Transfer Protocol Secure
* API: Application Programming Interface
* SMS: Short Message Service
* FTP: File Transfer Protocol
* GUI: Graphical User Interface

**Appendix B: Analysis Models**

* Use Case Diagram: A diagram illustrating the interactions between users (actors) and the system, depicting the various use cases and their relationships.
* Entity-Relationship Diagram: A diagram representing the relationships between different entities in the system and their attributes.
* Data Flow Diagram: A diagram illustrating the flow of data within the system, showing how data moves from one process to another and the external entities involved.

**Appendix C: To Be Determined List**

1. TBD: Performance requirements for specific functionalities.

2. TBD: Detailed API protocols for integration with third-party services.

3. TBD: Specific hardware and software requirements for optimal system operation.

4. TBD: Communication standards and protocols for email notifications and messaging features.

5. TBD: Safety certifications and regulations applicable to the housing society management system.

**Sprint Backlog - Sprint 1:**

**1. User Story: User Registration**

* Task 1: Design and create registration form UI.
* Task 2: Implement backend logic for handling registration requests.
* Task 3: Integrate form submission with database to store housing society details.
* Task 4: Implement validation for registration inputs.
* Task 5: Write unit tests for registration functionality.

**2. User Story: Maintenance Request Submission**

* Task 1: Develop UI components for submitting maintenance requests.
* Task 2: Implement backend API endpoints for handling maintenance request submissions.
* Task 3: Integrate form data with database to store maintenance requests.
* Task 4: Ensure validation of maintenance request inputs.
* Task 5: Write unit tests for maintenance request submission.

**3. User Story: Facility Booking**

* Task 1: Design and implement UI for facility booking feature.
* Task 2: Develop backend functionality for processing facility booking requests.
* Task 3: Integrate booking information with database for storage and retrieval.
* Task 4: Validate booking requests and handle conflicts in booking schedules.
* Task 5: Write unit tests for facility booking.

**4. User Story: Real-time Notifications and Alerts**

* Task 1: Design notification system architecture.
* Task 2: Implement backend services for generating real-time notifications.
* Task 3: Integrate notification triggers with relevant system events.
* Task 4: Implement frontend components for displaying notifications to users.
* Task 5: Write unit tests for notification system.

**5. Documentation and Testing**

* Task 1: Document user stories and acceptance criteria for implemented features.
* Task 2: Write unit tests for backend API endpoints and frontend components.
* Task 3: Conduct manual testing to ensure functionality and usability of implemented features.
* Task 4: Prepare documentation for sprint review and retrospective.
* Task 5: Review and refine documentation based on stakeholder feedback.

**6. Sprint Review and Retrospective**

* Task 1: Demo implemented features to stakeholders and gather feedback.
* Task 2: Reflect on sprint progress, identify areas for improvement, and plan adjustments for future sprints.
* Task 3: Document sprint review and retrospective findings for continuous improvement.

This sprint backlog includes tasks for implementing the selected user stories, ensuring that the core functionalities of the Housing Society Management System are developed and tested within the allocated time frame.