SRS report

1. Introduction:

In Pakistan's busy cities, almost 37.73% of people live, which is about 78 million people. This makes Pakistan one of the countries in South Asia with lots of city dwellers. Among them, about 20% live in housing societies across the country, and in Lahore, it's even higher at 21%.

Given these numbers, it's clear that many people live in housing societies, where they share common facilities like parks, gyms, and swimming pools. However, managing these places can be a big challenge because there are so many people with different needs.

That's where our project comes in.

1.1. Purpose of the Project:

We're creating the Housing Society Resident Management System (HSRMS) to make life easier for everyone in these housing societies. Our goal is to use technology to help manage things smoothly.

With HSRMS, residents, staff, and administrators can all work together better. Residents can book amenities, report problems, and keep in touch with each other. Staff can keep track of maintenance requests and make sure everything runs smoothly. Administrators can share important announcements and keep everyone informed.

By making things simpler and more organized, we hope to make living in housing societies a lot more enjoyable for everyone involved.

1.2. Intended Audience and Reading Suggestions:

The Software Requirement Specification (SRS) Report is intended for a diverse audience involved in the development, testers, software architects, designers and management of the Housing Society Resident Management System (HSRMS).

1.3. Project Scope:

The project scope entails developing the Housing Society Resident Management System (HSRMS) to streamline housing society activities. HSRMS will feature user management, amenity booking, announcements, maintenance requests, visitor management, complaint

resolution, and billing/payment management. It prioritizes user-friendly interfaces, secure authentication, and efficient data management. Integration with existing systems and scalability are key considerations. HSRMS aims to optimize operations, enhance resident satisfaction, and foster community cohesion.

1.4. Definitions, acronyms, and abbreviations (Glossary / Terminology):

Terminology	Definition
HSRMS	Housing Society Resident Management System
User	Individual interacting with HSRMS
Resident	Person living in the housing society
Administrator	Overseer of HSRMS operations
Amenity	Society facility (e.g., clubhouse)
Maintenance Request	Resident-reported issue
Visitor	Temporary society guest
Complaint	Resident dissatisfaction report
Billing	Invoice generation for services
Payment	Settling fees for society services
Authentication	User identity verification
Authorization	User access permission
Profile Management	User information update
Notification	Important message to users
Emergency Vehicle	Priority-response vehicle

This glossary aids in understanding terms pertinent to the HSRMS project, ensuring clear communication.

2. Overall Description

This section will explain the aspects of the HSRMS and requirements.

2.1. Project Function:

2.1.1. User Management:

User Management ensures seamless access and security for residents, staff, and administrators within the system.

- Facilitates effortless user registration and login processes.
- Manages password functionalities, including reset and change options for enhanced security.
- Assigns user roles and permissions effectively to control access levels.
- Enables profile management, allowing users to update personal information conveniently.

2.1.2. Resident Information:

Resident Information functionality maintains a comprehensive database of resident details, promoting efficient communication and emergency response.

- Stores resident personal information securely within the system.
- Records essential unit details such as apartment/house numbers, blocks, and floors.
- Manages family member and occupant information for accurate demographic insights.
- Preserves emergency contact details for swift reference during critical situations.

2.1.3. Amenity Booking:

Amenity Booking empowers residents to reserve community amenities hassle-free, enhancing their living experience.

- Displays a curated list of available amenities for residents' convenience.
- Streamlines the reservation process with intuitive date and time selection options.
- Tracks booking history and status updates to manage amenity usage effectively.
- Establishes clear rules and policies to ensure fair and equitable access to amenities.

2.1.4. Announcements:

Announcements functionality keeps residents informed about critical updates and community events, fostering a sense of belonging.

- Empowers administrators to create and edit announcements seamlessly.
- Ensures timely notification delivery to residents through a robust mechanism.
- Maintains an archive of past announcements for reference and historical context.

2.1.5. Maintenance Requests:

Maintenance Requests streamline the reporting and resolution of maintenance issues within the housing society.

- Categorizes maintenance issues for efficient handling, including plumbing, electrical, etc.
- Facilitates resident-initiated maintenance request submissions for prompt resolution.
- Assigns tasks and tracks progress for maintenance staff to ensure timely resolution.
- Provides status updates and resolution notifications to keep residents informed.

2.1.6. Water Delivery Bookings:

Water Delivery Bookings simplify the process of requesting and receiving water deliveries, ensuring residents' convenience.

- Guides residents through a straightforward request process for water deliveries.
- Tracks delivery schedules efficiently to accommodate residents' preferences.
- Manages billing and payment processes seamlessly for transparent transactions.

2.1.7. Visitor Management:

Visitor Management enhances security and organization by managing the registration and approval of visitors to the housing society.

- Implements a structured visitor registration and approval workflow for enhanced security.
- Grants temporary access permissions to approved visitors, ensuring smooth entry and exit.

2.1.8. Complaint Board:

The Complaint Board provides residents with a platform to voice concerns and ensures systematic resolution of issues.

- Offers a user-friendly platform for residents to submit complaints and feedback.

- Categorizes complaints to streamline resolution processes, such as noise or cleanliness issues.
- Tracks complaint status and provides updates on resolution progress to residents.

2.1.9. Billing and Payments:

Billing and Payments functionality simplifies financial transactions and record-keeping for maintenance fees and amenities.

- Generates bills accurately for various services and fees within the housing society.
- Offers convenient online and offline payment options for resident’s flexibility.
- Maintains billing history and tracks payment status for easy reference.
- Sends timely notifications for late payments, ensuring residents stay informed.

2.2. User Classes and Characteristics

This section outlines the user classes and their respective characteristics within the Housing Society Resident Management System (HSRMS).

2.2.1. User Classes:

1. **Residents**: Individuals residing within the housing society.
2. **Staff**: Personnel responsible for managing and maintaining the housing society.
3. **Administrators**: Designated individuals overseeing the operation and administration of the HSRMS.

2.2.2. Characteristics and Expectations:

- **Interaction with the System**: Users interact with the HSRMS through various interfaces, including web and mobile application.
- **Minimal User Interaction**: As the system primarily operates within housing societies, user interaction is expected to be minimal, especially for routine tasks.
- **Limited System Knowledge**: Users may have little to no prior knowledge of the system's functionalities and may require intuitive interfaces and guidance.

- **Assumptions on User Interaction**: Users can interact with the system through basic actions such as clicking, typing, and navigating through menus.

2.3. Operating Environment for HSRMS

The Housing Society Resident Management System (HSRMS) is designed to be flexible and adaptable, operating as a web-based application accessible from various devices and platforms. The system operates smoothly in diverse operating environments and does not require extensive hardware resources. Key considerations for its operating environment include:

2.3.1. Hardware Requirements:

- Standard Computing Devices: HSRMS functions efficiently on desktop computers, laptops, tablets, and smartphones commonly used by residents, staff, and administrators.
- Basic Processor: The system operates effectively on devices with standard processors commonly found in modern computing devices.
- Adequate Memory: While HSRMS does not require extensive memory, sufficient RAM is recommended to ensure smooth performance during user interactions.

2.3.2. Software Specifications:

- Cross-Platform Compatibility: HSRMS is compatible with popular web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge, ensuring accessibility across different operating systems including Windows, macOS, iOS, and Android.
- Responsive Design: The user interface of HSRMS is designed to be responsive, adapting seamlessly to various screen sizes and resolutions, providing an optimal viewing experience for users on different devices.

2.3.3. Internet Connectivity:

- Stable Internet Connection: HSRMS relies on a stable internet connection to ensure real-time access to features and functionalities, such as amenity booking, announcement viewing, and maintenance request submission.

3. Requirement Specification

This section includes requirements to build our software system and description of its behaviour.

3.1. User Interface

The Housing Society Resident Management System (HSRMS) offers a streamlined user interface (UI) designed for accessibility and ease of use. Users navigate through a centralized dashboard and sleek menu to access features like amenity booking, maintenance requests, announcements, and billing/payment. Personalized profiles allow users to manage settings and track activity, while administrators oversee complaints and visitor registrations. With a focus on simplicity and functionality, the HSRMS UI enhances the management of housing society tasks for all users.

3.2. Software Interface

The HSRMS software interface facilitates secure user authentication, API integration for data exchange, and efficient database management. It enables communication channels for announcements and for tasks like amenity booking. An admin dashboard offers insights into housing society operations. Overall, the interface streamlines user interaction, enhancing housing society management.

3.3. Functional Requirement:

Name	Details
User Management	Registration, login, and profile management for residents, staff, and administrators.
Resident Information	Storage of resident personal details, unit information, and emergency contacts.
Amenity Booking	Reservation of community amenities like the clubhouse, gym, and swimming pool.
Announcements	Creation and dissemination of important information and notices by administrators.
Maintenance Requests	Logging and tracking of maintenance issues reported by residents.
Water Delivery Bookings	Requesting and scheduling water deliveries.
Visitor Management	Registering and approving visitors to the housing society.
Complaint Board	Logging and tracking resident complaints for resolution.
Billing and Payments	Generating bills for maintenance fees, and amenities, and facilitating various payment options.

3.3.1. User Management:

3.3.1.1 User Registration and Login:

- The system shall provide a user registration interface allowing administrators to create and add an account for a resident or staff member or a new administrator.
- The system shall have only one resident account linked to each unit.
- Upon successful registration, the system shall authenticate users through a secure login process.

3.3.1.2 Password Management:

- Users shall have the ability to change their passwords through the user profile settings.
- The system shall provide an option for users to reset their passwords through a secure phone number verification process.

3.3.1.3 User Roles and Permissions

- The system shall define roles such as resident, staff, and administrator, each with specific permissions.
- Administrators shall have the capability to assign and modify roles and permissions for users.
- The system shall incorporate a structured staff hierarchy comprising five distinct categories, each assigned specific permissions and responsibilities.
- The system shall define staff categories tailored to distinct functional areas, encompassing amenity booking-related requests, facilitation of maintenance requests, coordination of water delivery bookings, administration of the complaint board, and oversight of billing and payments management.

3.3.1.4 Profile Management:

- Residents shall be able to update their personal information, including contact details and emergency contacts, through the user profile.
- The system shall maintain an audit trail of profile updates, recording the date and time of modifications.

3.3.2. Resident Information:

3.3.2.1 Storage of Personal Details:

- The system shall provide a secure database for storing resident personal details, including names, contact information, and identification documents.

3.3.2.2 Unit Details:

- Administration shall input and update information about resident's units, including apartment/house number, block, and floor.
- The system shall enforce data validation to ensure accurate unit details are recorded.

3.3.2.3 Family Members and Occupants:

- The system shall allow residents to add and manage information about family members and other occupants residing in the unit.

3.3.2.4 Emergency Contact Details:

- Residents shall provide and update emergency contact details, which will be accessible for quick reference during emergencies.

3.3.3. Amenity Booking:

3.3.3.1 List of Available Amenities:

- The system shall display a list of community amenities available for booking with the time slots they are available.

3.3.3.2 Reservation Process:

- Residents shall be able to select a specific amenity, date, and time slot for reservation.
- The system shall prevent double bookings for the same amenity and time slot.
- The system shall add the new bookings to an approval list that shall be approved by the relevant staff member.
- Staff members shall be able to view bookings and approve or discuss about new bookings.

3.3.3.3 Booking History and Status:

- The system shall maintain a history of amenity bookings, including details of reservations, cancellations, status tracking and charges.
- Residents shall receive bills for amenity bookings, adding them to the payment centre.

3.3.3.4 Rules and Policies:

- The system shall provide a section displaying rules and policies for amenity usage, ensuring residents are informed and compliant.

3.3.4 Announcements:

3.3.4.1 Announcement Creation and Editing:

- Administrators shall have the ability to create and edit announcements, specifying the target audience and visibility settings.
- The system shall be able to distinguish and display staff announcements on their dashboard and resident announcements on their dashboard.

3.3.4.2 Notification Mechanism:

- The system shall notify residents or staff members of new announcements through push notifications, or an in-app notification center.

3.3.4.3 Archive of Past Announcements:

- The system shall maintain an archive of past announcements for reference, categorizing them based on date and relevance.

3.3.5 Maintenance Requests:

3.3.5.1 Maintenance Issue Categorization:

- Residents shall categorize maintenance issues (e.g. plumbing, electrical) when submitting maintenance requests.

3.3.5.2 Resident-initiated Maintenance Requests:

- The system shall provide residents with a platform to submit maintenance requests, attaching relevant details and images.

3.3.5.3 Request Assignment and Tracking:

- Maintenance staff shall receive requests with detailed information and update the status of the request.

3.3.5.4 Status Updates and Resolution Notifications:

- The system shall notify residents of maintenance request status updates and resolution notifications through push notifications, or an in-app notification center.

3.3.6 Water Delivery Bookings:

3.3.6.1 Request Process for Water Deliveries:

- Residents shall request water deliveries through a user-friendly interface, specifying quantity and preferred delivery frequency and time.
- Residents shall be able to cancel any booked order one day before the next delivery.

3.3.6.2 Delivery Schedule and Tracking:

- The system shall provide residents with a delivery schedule, tracking the status of their water delivery requests.

3.3.6.3 Billing and Payment:

- Residents shall receive bills for water deliveries, adding them to the payment centre.

3.3.7. Visitor Management:

3.3.7.1 Visitor Registration and Approval Workflow:

- Residents shall register visitors through the system, providing necessary details for approval by the administration.

3.3.7.2 Temporary Access Permissions:

- Approved visitors shall be granted temporary access permissions, automatically revoked after each month.

3.3.8. Complaint Board:

3.3.8.1 Platform for Complaint Submission:

- The system shall provide a user-friendly platform for residents to submit complaints, selecting the appropriate category.

3.3.8.2 Categorization of Complaints:

- The system shall categorize complaints for efficient tracking and resolution, allowing administrators to prioritize urgent issues.

3.3.8.3 Status Tracking and Resolution Updates:

- Residents shall receive real-time updates on the status of their complaints and notifications upon issue resolution.

- Related staff members shall be able to update the status of each complaint.

3.3.9. Billing and Payments:

3.3.9.1 Bill Generation:

- The system shall automatically add up bills for maintenance fees, amenities, water deliveries, and other dues throughout the month.
- The system shall generate an overall bill at the last date of each month.
- The system shall send a notification to the resident upon the generation of their bill.
- The system shall provide users with an electronic bill receipt that can be accessed in the Payments Center.
- The electronic bill receipt shall be printable, ensuring users have the option to maintain physical copies for their records.

3.3.9.2 Online and Offline Payment Options:

- Residents shall have the option to make payments online through secure payment gateways or offline methods.

3.3.9.3 Billing History and Payment Tracking:

- The system shall maintain a detailed billing history for each resident, including payment receipts and transaction tracking.

3.3.9.4 Late Payment Notifications:

- The system shall send automated notifications to residents for overdue payments, specifying late fees and payment deadlines.

3.4. Non-Functional Requirement:

Name	Details
Reusability	Promotes modular code for easy reuse and future enhancements.
Maintainability	Ensures well-documented, modular code for easy updates.
Reliability	Aims for 99% system availability with redundant systems.
Security	Implements multi-factor authentication and data encryption.
Scalability	Accommodates increasing users and data volumes.

Performance	Ensures timely response to user requests and optimizes code.
Flexibility	Adapts to changing requirements without extensive modifications.
Portability	Easily deployable across different platforms.
Interoperability	Enables seamless data exchange with external systems.

3.4.1 Reusability (Object-oriented approach):

Requirement:

- The system shall be designed using object-oriented principles to promote reusability of components.
- Code modules should be organized in a modular and cohesive manner to facilitate easy reuse in future enhancements or similar projects.

Justification:

- This approach promotes code modularity and reusability, allowing for easier maintenance and for future development of new features.

3.4.2 Maintainability:

Requirement:

- The system shall be designed with well-documented code throughout the development process to enhance readability and ease of maintenance.
- Modular design with clear separation of components should be implemented to allow for easy updates and modifications i-e without impacting other parts of the system.
- Documentation, including user manuals and technical guides should be provided to assist the cutomer in understanding and maintaining the system.

Justification:

- This allows non-technical users with the help of readily available documentation and tutorials to perform basic maintenance task.

3.4.3 Reliability (operational availability):

Requirements:

- The system shall be available for use at least 99% of the time during business hours.

- Implement redundant systems and failover mechanisms to minimize downtime incase of unforeseen events and to ensure high availability.

Justification:

- Residents rely on the system for essential tasks like submitting maintenance requests and accessing important announcements. High availability ensures minimal disruption to their lives.

3.4.4 Security (User access and data encryption):

Requirements:

- The system shall implement multi-factor authentication for all user logins.
- The system shall encrypt all sensitive data, including resident personal information, financial data etc.
- Compliance with relevant security standards and regulations should be ensured to maintain the security and privacy of user data.

Justification:

- This adds an extra layer of security to prevent unauthorised access to resident information and system functionalities and furthermore data encryption protects sensitive information from unauthorised access even if the system is compromised.

3.4.5 Scalability:

Requirements:

- The system shall be able to accommodate an increasing number of users (residents, staff) and data (resident information, booking history, etc.) without significant performance degradation.
- The system should dynamically adjust its resources based on demand, automatically scaling up or down to maintain performance during peak usage periods.
- Implement load balancing mechanisms to evenly distribute incoming traffic across multiple servers, ensuring optimal resource utilization and preventing overload on any single server.

Justification:

- Housing societies can grow over time, and the system needs to adapt to this growth efficiently. Adding new residents, managing additional amenities, and handling increased

usage should not negatively impact system performance.

3.4.6 Performance:

Requirements:

- The system shall respond to user requests within an acceptable timeframe, ensuring a smooth user experience.
- Specify the maximum number of transactions or requests the system should be able to handle within a given time period, ensuring that it can support the expected workload.
- Regularly optimize code, database queries, and network configurations to minimize latency and maximize system efficiency, ensuring smooth performance under varying load conditions.

Justification:

- Residents rely on the system for various tasks, and slow response times can lead to frustration and hinder daily activities. Defining acceptable response times for different functionalities (e.g., login, amenity booking, bill payment) ensures efficient system operation.

3.4.7 Flexibility:

Requirements:

- The system shall be adaptable to changing requirements and new features without extensive modifications to the core architecture.
- Design the system with flexibility in mind, allowing for easy modification and extension of functionalities without requiring extensive changes to the existing codebase.
- Allow for dynamic configuration changes without the need for system downtime, enabling administrators to adjust settings on-the-fly to adapt to changing business needs.

Justification:

- As the society's needs evolve, new functionalities might be required. A flexible system allows for easier integration of these features without compromising existing functionalities. This can include features like adding new amenity types, integrating with smart home devices, or implementing online voting for society decisions.

3.4.8 Portability:

Requirements:

- The system should be designed to be portable across different hardware and software platforms with minimal effort.
- Package the application and its dependencies into containers using technologies like Docker to ensure consistent deployment and portability across different environments.
- Minimize dependencies on specific hardware or software configurations to facilitate deployment on diverse infrastructure environments.

Justification:

- Portability provides flexibility in choosing the underlying infrastructure (cloud-based, on-premise) and allows for easier migration to newer technologies in the future. This can be achieved through platform-independent programming languages and containerization technologies.

3.4.9 Interoperability:

Requirements:

- The system shall be able to exchange data and interact with other relevant systems seamlessly, such as accounting software for bill payments or external visitor management systems.
- Design well-documented and consistent APIs to facilitate integration with external systems, enabling seamless data exchange and interoperability.
- Support commonly used data formats such as JSON, XML, or CSV for data interchange, ensuring compatibility with a wide range of systems and applications.

Justification:

- Interoperability eliminates the need for manual data entry and ensures a smooth flow of information between the resident management system and other relevant applications used by the society. This can be achieved through the use of standard APIs and data formats.

4. References:

https://www.pbs.gov.pk/publication/report-recommendations-adoption-best-practices-7th-population-housing-census

https://www.worldometers.info/world-population/pakistan-population/