

Project Title:
Residential Township Management System

Document Type:
Functional Requirement Document (FRD)

Prepared By:
Suraj Patil
Business Analyst
surajpatil@outlook.com

Date:
19 July 2025

User Requirements Document
for
SaaS Based Residential Township Management System

Table Of Contents:

TITLE	SUB POINTS (if any)	PAGE NO
Introduction	Project Overview	3
	Project Goal	3
	Target Audience	3
Key Features		4,5
Problems		5
Opportunities		6
User Persona	-	7,8
User List From User Persona	-	9
User Stories	-	10,11
Use Case	-	12
Functional Requirements	-	13,14
Non-Functional Requirements	-	15
Technical Requirements	-	16
Infrastructure	Components	16
Conclusion	-	17

Introduction

Project Overview:

The SaaS-Based Residential Township Management System is a cloud-based platform designed to manage and automate day-to-day operations in residential communities. It provides modules for resident management, billing, complaints, visitor tracking, and amenity bookings. The system offers role-based access for residents, admins, and staff, ensuring security and ease of use. It aims to enhance efficiency, transparency, and communication across the township.

Project Goals:

- **Digitize Township Operations:** Eliminate manual processes by automating key functions such as billing, maintenance requests, visitor tracking, and facility booking.
- **Enhance Resident Experience:** Provide a user-friendly interface for residents to access services, raise complaints, and receive updates in real time.
- **Improve Transparency & Communication:** Enable clear and timely communication between residents, management, and staff through notices, alerts, and notifications.
- **Centralized Multi-Township Management:** Support multiple townships under one platform with scalable architecture and modular features.
- **Ensure Data Security & Role-Based Access:** Implement robust authentication and authorization mechanisms to protect sensitive user and financial data.

Target Audience:

- **Township Management Committees (TMCs)** – Responsible for the day-to-day administration of residential communities.
- **Facility Management Companies** – Organizations managing maintenance, security, and housekeeping services in residential complexes.
- **Residents & Homeowners** – Individuals and families living within the township who need access to community services and information.
- **Security & Support Staff** – On-ground personnel using the system for visitor management, complaint resolution, and routine updates.
- **Real Estate Developers** – Builders or developers managing multiple townships who require centralized control and insights.
- **Service Vendors** – External vendors such as electricians, plumbers, or delivery services interacting with township operations via the system.

Key Features:

- **Resident Management:**
 - Add, edit, and manage resident profiles
 - Flat/unit allotment and ownership records
- **Billing & Invoicing:**
 - Automated generation of maintenance bills
 - Online payment gateway integration
 - Payment history and reminders
- **Visitor Management:**
 - Real-time visitor entry/exit tracking
 - Pre-approval of guests by residents
 - Digital gate pass system
- **Complaint & Maintenance Request Handling:**
 - Lodge, track, and resolve complaints
 - Assign issues to staff with status updates
- **Facility Booking System:**
 - Book common areas (clubhouse, gym, etc.)
 - Manage slot availability and usage rules
- **Communication & Notifications:**
 - Notices, circulars, and announcements
 - Push notifications, SMS, or email alerts
- **Asset & Inventory Management:**
 - Track common assets and maintenance schedules
 - Monitor inventory usage and replenishment
- **Role-Based Access Control:**
 - Different user access levels (admin, resident, staff)
 - Secure login with permissions per role
- **Analytics & Reporting:**
 - Dashboard with KPIs and performance insights
 - Financial, maintenance, and user activity reports

- **Multi-Township Support:**

Manage multiple communities under one platform

Centralized administration for developers or management firms

- **Mobile & Web Access:**

Fully responsive system accessible via app or browser

Cross-platform compatibility

Problems:

- **Manual and Paper-Based Processes:**

Most townships rely on manual records for billing, visitor logs, complaints, and communication, which are time-consuming and error-prone.

- **Lack of Transparency:**

Residents often have limited visibility into billing calculations, maintenance schedules, and complaint status, leading to mistrust and dissatisfaction.

- **Inefficient Communication:**

Important notices and updates are often missed due to outdated communication methods (e.g., notice boards or WhatsApp groups).

- **Fragmented Systems:**

Townships use multiple disconnected tools for different operations (billing, complaints, visitor logs), causing data inconsistency and management issues.

- **Limited Access to Information:**

Residents and staff cannot access necessary information remotely or on demand, affecting convenience and responsiveness.

- **No Centralized Control for Developers/Managers:**

Real estate developers or companies managing multiple townships lack a centralized dashboard to monitor operations across locations.

- **Security Concerns:**

Unauthorized access to township premises due to inefficient visitor tracking can pose security threats.

- **Delayed Issue Resolution:**

Without a structured complaint management system, service requests may be delayed or unresolved.

Opportunities:

- **Digital Transformation of Residential Communities:**
The project presents an opportunity to modernize township operations by replacing outdated, manual systems with automated digital processes.
- **Growing Demand for Smart Living Solutions:**
With the rise of smart cities and tech-savvy residents, there's increasing demand for efficient, tech-enabled community management platforms.
- **Recurring Revenue Through SaaS Model:**
The subscription-based model provides sustainable, recurring revenue from townships while offering flexible pricing based on scale.
- **Scalability Across Regions and Townships:**
The platform can be easily scaled to serve multiple townships, residential societies, or real estate developers across different geographies.
- **Data-Driven Decision Making:**
The system's built-in analytics can help township managers make informed decisions using real-time insights on finances, complaints, and usage patterns.
- **Third-Party Integration Potential:**
Future integration with services like payment gateways, smart devices (IoT), e-Governance APIs, or delivery apps can increase value and functionality.
- **Brand Positioning in a Niche Market:**
By targeting gated communities, real estate firms, and facility managers, the system can carve out a strong presence in a specialized B2B market.

User Personas:

1. Shweta – Resident/Homeowner

Age: 34

Role: Apartment Owner in a gated township

Goals:

- View and pay maintenance bills online
- Raise and track service complaints
- Book facilities like clubhouse or gym
- Receive real-time alerts and updates

Pain Points:

- Delayed communication from management
- Lack of visibility into complaint status
- Inconvenient payment process

2. Suraj – Facility Manager

Age: 30

Role: Works for a property management company

Goals:

- Assign and track maintenance tasks
- Manage facility bookings and staff schedules
- Monitor complaints and close them efficiently

Pain Points:

- Difficulty in tracking daily operational tasks
- No unified system for managing staff, assets, and services
- Communication gaps with residents

3. Krishna – Security Guard

Age: 28

Role: Entry gate security staff

Goals:

- Register visitor entries and exits quickly
- Approve or deny visitor access based on resident input
- Generate daily visitor logs

Pain Points:

- Manual registers are slow and prone to errors
- Difficulty verifying guests without resident input
- No real-time visitor approval mechanism

4. Shubham – Township Admin/Super Admin

Age: 45

Role: Management committee member or admin

Goals:

- Monitor all operations across the township
- Generate reports on billing, complaints, facility usage
- Manage users and set permissions

Pain Points:

- Lack of central visibility into all activities
- Difficulty tracking financial and operational KPIs
- Too much dependency on manual processes

5. Vikram – Real Estate Developer

Age: 50

Role: Developer managing multiple townships

Goals:

- Get performance reports from all townships
- Monitor tenant/resident satisfaction
- Centralize data and scale operations

Pain Points:

- No centralized system for cross-township visibility
- Delayed performance insights from management teams
- Operational inconsistency between different locations

User List from User Persona:

User Type	Primary/Secondary Reason	
Resident/Homeowner	Primary User	Core users of the system for bills, complaints, facility booking, and notifications.
Facility Manager	Primary User	Actively uses the system daily to manage operations, tasks, and services.
Security Guard	Primary User	Uses the system regularly for visitor entry, gate passes, and logs.
Township Admin	Primary User	Oversees all township-level functions and manages data and users.
Super Admin	Secondary User	Uses the system to monitor multiple townships and configure high-level settings.
Real Estate Developer	Secondary User	Occasionally accesses dashboards and reports for performance monitoring.

User Stories:

Here are user stories according to user role

- **Resident/Homeowner:**

As a resident, I want to view and download my monthly maintenance bills so that I can track and pay my dues on time.

As a resident, I want to raise complaints about facility or maintenance issues so that they are addressed quickly.

As a resident, I want to book common amenities like the gym or clubhouse so that I can use them without conflicts.

As a resident, I want to receive notices and emergency alerts so that I stay informed about community matters.

As a resident, I want to approve or reject visitors digitally so that security at the gate is managed efficiently.

- **Facility Manager:**

As a facility manager, I want to assign incoming complaints to the appropriate staff so that issues are resolved efficiently.

As a facility manager, I want to monitor facility bookings and availability so that double bookings are avoided.

As a facility manager, I want to manage inventory and track usage so that supplies are replenished on time.

As a facility manager, I want to update complaint status (e.g., in-progress, resolved) so that residents are informed.

- **Security Guard:**

As a security guard, I want to log visitor entries and exits digitally so that I can maintain accurate visitor records.

As a security guard, I want to verify visitor approval status so that only authorized guests are allowed in.

As a security guard, I want to generate end-of-day visitor reports so that the management has clear visibility.

- **Township Admin:**

As a township admin, I want to manage user accounts and roles so that only authorized personnel can access specific features.

As a township admin, I want to view dashboards and reports on billing, complaints, and usage so that I can monitor performance.

As a township admin, I want to post announcements and notices so that all residents are informed in a timely manner.

- **Super Admin / Developer:**

As a super admin, I want to oversee multiple townships from a centralized dashboard so that I can compare and monitor performance.

As a super admin, I want to configure global settings and permissions so that operations across townships stay consistent.

As a developer, I want to generate summary reports of occupancy, billing, and resident satisfaction so that I can assess ROI.

User Case:

Use Case ID	Use Case Name	Actor(s)	Description
UC-01	View & Pay Maintenance Bills	Resident	Allows residents to view, download, and pay their maintenance bills online.
UC-02	Lodge a Complaint	Resident	Enables residents to raise service/maintenance-related complaints.
UC-03	Track Complaint Status	Resident	Lets residents track the progress and resolution status of their complaints.
UC-04	Book a Facility	Resident	Allows booking of shared amenities (e.g., gym, hall) with time-slot selection.
UC-05	Receive Notifications & Alerts	Resident	Residents receive announcements, alerts, and updates from management.
UC-06	Approve Visitor Entry	Resident	Enables residents to pre-approve or reject visitor access from their device.
UC-07	Manage Visitor Entry	Security Guard	Security staff records visitor entry/exit and checks for resident approvals.
UC-08	Generate Visitor Reports	Security Guard, Admin	Allows generation of daily/weekly visitor logs for review and security checks.
UC-09	Assign Complaints to Staff	Facility Manager	Managers assign incoming issues to the appropriate personnel or department.
UC-10	Update Complaint Status	Facility Manager	Staff/managers update the complaint resolution progress (e.g., pending, closed).
UC-11	Manage Facility Booking	Facility Manager	Administers availability and slot control for shared amenities.
UC-12	Manage Inventory & Assets	Facility Manager	Enables tracking and updating of stock levels, equipment, and maintenance needs.
UC-13	Send Notices & Announcements	Township Admin	Sends important communication to all or specific groups of users.
UC-14	Generate Reports & Dashboards	Township Admin, Super Admin	Generates insights and KPIs on billing, complaints, usage, etc.
UC-15	Manage Users and Permissions	Township Admin	Adds, edits, or deletes users and assigns them roles with appropriate access.
UC-16	Monitor Multiple Townships	Super Admin, Developer	Provides centralized monitoring of multiple townships on one platform.
UC-17	Configure System Settings	Super Admin	Configures settings like township name, modules, limits, and default roles.

Functional Requirements:

- **Resident Functions:**

FR-01: The system shall allow residents to **view, download, and pay maintenance bills** online.

FR-02: The system shall enable residents to **raise complaints** and view their current status.

FR-03: The system shall allow residents to **book facilities** (e.g., gym, clubhouse) and prevent double bookings.

FR-04: The system shall send **real-time notifications and announcements** to residents.

FR-05: The system shall allow residents to **approve or reject visitor entries** via mobile or web app.

- **Security Functions:**

FR-06: The system shall allow security guards to **register visitor entries and exits**.

FR-07: The system shall verify visitor access against **resident approvals** before allowing entry.

FR-08: The system shall generate **daily visitor logs and reports**.

- **Facility Manager Functions:**

FR-09: The system shall allow facility managers to **assign complaints** to staff or departments.

FR-10: The system shall allow managers to **update complaint status** (e.g., Open, In-Progress, Resolved).

FR-11: The system shall allow facility managers to **monitor and manage facility bookings**.

FR-12: The system shall allow facility managers to **track inventory and assets**, and send alerts for low stock.

- **Admin/Super Admin Functions:**

FR-13: The system shall allow township admins to **create, edit, deactivate users** and assign roles.

FR-14: The system shall allow admins to **send notices and announcements** to individuals or groups.

FR-15: The system shall provide dashboards and **generate reports** on complaints, billing, occupancy, etc.

FR-16: The system shall allow super admins to **monitor multiple townships** from a centralized interface.

FR-17: The system shall allow super admins to **configure settings** such as modules, role templates, and township details.

- **General System Functions:**

FR-18: The system shall provide **role-based access control** to ensure users only access relevant modules.

FR-19: The system shall be accessible via **web and mobile platforms**.

FR-20: The system shall maintain **audit logs** for all critical operations (e.g., user creation, billing updates).

FR-21: The system shall provide **multi-language support** (if applicable).

FR-22: The system shall support **secure online payments** through integrated payment gateways.

FR-23: The system shall send **email/SMS/push notifications** for important alerts.

Non-Functional Requirements:

- **NFR-01 – Performance:**
The system shall respond to user actions within 2 seconds under normal load and support at least 500 concurrent users efficiently.
- **NFR-02 – Scalability:**
The system shall be designed to scale horizontally, allowing it to support multiple residential townships and thousands of users without performance degradation.
- **NFR-03 – Availability:**
The system shall ensure 99.9% uptime to maintain uninterrupted service access for all users.
- **NFR-04 – Security:**
The platform shall enforce secure login mechanisms, role-based access control, and use SSL/TLS encryption for all data transmissions.
- **NFR-05 – Reliability:**
The system shall handle exceptions gracefully and retry critical operations like payment processing in the event of temporary failures.
- **NFR-06 – Usability:**
The application interface shall be intuitive and easy to use across devices, requiring minimal training for end-users.
- **NFR-07 – Portability:**
The system shall be compatible with all modern web browsers and available on Android and iOS platforms.
- **NFR-08 – Maintainability:**
The codebase shall follow modular design principles and be well-documented to support easy maintenance, debugging, and updates.
- **NFR-09 – Data Backup:**
The system shall automatically perform daily backups and store them securely, ensuring data recovery is possible for at least the last 30 days.
- **NFR-10 – Compliance:**
The system shall comply with relevant data privacy regulations such as GDPR or national IT laws, ensuring secure and lawful data handling.
- **NFR-11 – Localization:**
The platform shall support multiple languages and local date/time formats to cater to users in different regions if needed.
- **NFR-12 – Auditability:**
The system shall log all key activities, including logins, transactions, and data changes, to support traceability and accountability.

Technical Requirements:

Technology Stack:

- **Frontend:**
React.js / Angular / Vue.js, HTML5, CSS3, JavaScript, TailwindCSS or Bootstrap
- **Backend:**
Node.js (with Express) / Django (Python) / Spring Boot (Java), RESTful APIs, JWT Authentication
- **Database:**
PostgreSQL / MySQL (primary), MongoDB (for logs, messages, analytics)
- **Cloud Platform:**
AWS / Microsoft Azure / Google Cloud Platform (GCP)
- **Security:**
SSL/TLS encryption, Role-Based Access Control (RBAC), bcrypt for password hashing, OWASP compliance
- **CI/CD:**
Github or Jenkins for CI/CD

Infrastructure: Components

Cloud Hosting Platform:

AWS / Azure / GCP — for hosting backend services and databases.

Web & App Servers:

Hosts backend APIs and serves frontend apps (e.g., EC2, App Services).

Database Servers:

PostgreSQL / MySQL for transactional data

MongoDB (optional) for unstructured/log data

Object Storage:

AWS S3 / Azure Blob Storage — for storing files, documents, and images

Content Delivery Network (CDN):

Speeds up static content delivery (e.g., CloudFront, Azure CDN)

Load Balancer:

Distributes traffic across multiple servers for high availability

CI/CD Tools:

Jenkins / GitHub Actions — for automated testing and deployments

Security Infrastructure:

SSL/TLS certificates, firewalls, IAM roles, and access control mechanisms

Conclusion:

The **User Requirements Document (URD)** for the **SaaS-Based Residential Township Management System** presents a comprehensive overview of the system's purpose, goals, users, and functionality. It outlines key features, use cases, technical and infrastructure requirements, and the security and performance standards necessary for a reliable and scalable solution.