Software Requirements Specification (SRS)

CommUnity: Seamless Community Interaction and Management

1. Project Overview

CommUnity is a modular, multi-tenant Community Management System designed to simplify housing society management by connecting members and streamlining daily activities. The platform provides an efficient interface for both administrators and residents, facilitating interaction, management, and engagement within the community. The CommUnity platform is designed as a comprehensive digital solution for housing society management that bridges the gap between residents and administrators. It aims to modernize and streamline community interactions, daily operations, and administrative tasks through a unified digital platform.

2. Business Objectives

- Simplify housing society management
- Enhance communication between residents and administrators
- Increase resident participation in community activities
- Facilitate transparent financial transactions
- Improve community safety and security

3. Stakeholders

- Residents
- Society Administrators
- Security Personnel
- Maintenance Staff
- Service Vendors
- System Administrators

4. Functional Requirements

4.1 User Authentication and Registration

- Implement user registration for residents and admins
- Develop JWT-based authentication mechanism
- Create profile management functionality
- Support role-based access control

4.2 Resident and Society Management

- Design comprehensive database schema for society, flat, and resident information
- Enable society admins to add and update resident details
- Manage flat and resident associations
- Maintain detailed user profiles

4.3 Event and Notice Management

- Provide interface for admins to post events, notices, and circulars
- Allow residents to view and provide feedback on events
- Support event creation, update, and deletion

4.4 Complaint and Service Request Management

- Design complaint and service request interface
- Enable residents to log issues and service requests
- Develop backend functionality to track complaint status
- Allow admin monitoring and ticket closure
- Provide transparent communication on request progress

4.5 Maintenance Payment System

- Integrate Razorpay API for online maintenance bill payments
- Generate and manage monthly bills
- Provide payment history tracking
- Enable admin payment status monitoring

5. Non Functional Requirements:

1. **Security Requirements**

- Implement strong password policies with minimum length and complexity requirements
- Encrypt sensitive user data and payment information during storage and transmission
- Ensure secure session management with proper timeout mechanisms
- Implement role-based access control (RBAC) for different user types (admin, residents)
- Regular security audits and vulnerability assessments
- Secure storage of payment-related information complying with payment industry standards

2. **Performance Requirements**

- System should support concurrent access by multiple users with minimal latency
- Page load time should not exceed 3 seconds under normal conditions
- Database queries should be optimized to handle large volumes of resident data
- Support for multiple simultaneous payment transactions
- System should handle at least 1000 concurrent users without performance degradation

3. **Scalability and Availability**

- Application should be available 24/7 with 99.9% uptime
- System should be horizontally scalable to accommodate growing number of societies
- Support for multiple housing societies (multi-tenant architecture)
- Implement load balancing for better resource utilization
- Regular backup mechanism with quick disaster recovery capabilities

4. **Usability Requirements**

- Intuitive and user-friendly interface for both admin and residents
- Mobile-responsive design for access across different devices
- Consistent UI/UX across all modules
- Clear error messages and user notifications
- Support for multiple languages to accommodate diverse user base
- Accessibility compliance for users with disabilities

5. **Maintainability and Support**

- Well-documented code following industry standard practices
- Modular architecture for easy maintenance and updates
- Regular system maintenance windows for updates
- Logging mechanism for system activities and errors
- Support for different browsers (Chrome, Firefox, Safari, Edge)

6. **Compliance and Data Privacy**

- GDPR compliance for handling personal data
- Compliance with local data protection laws
- Regular data backup and retention policies
- Clear privacy policy and terms of service
- Audit trails for all critical operations

7. **Integration Requirements**

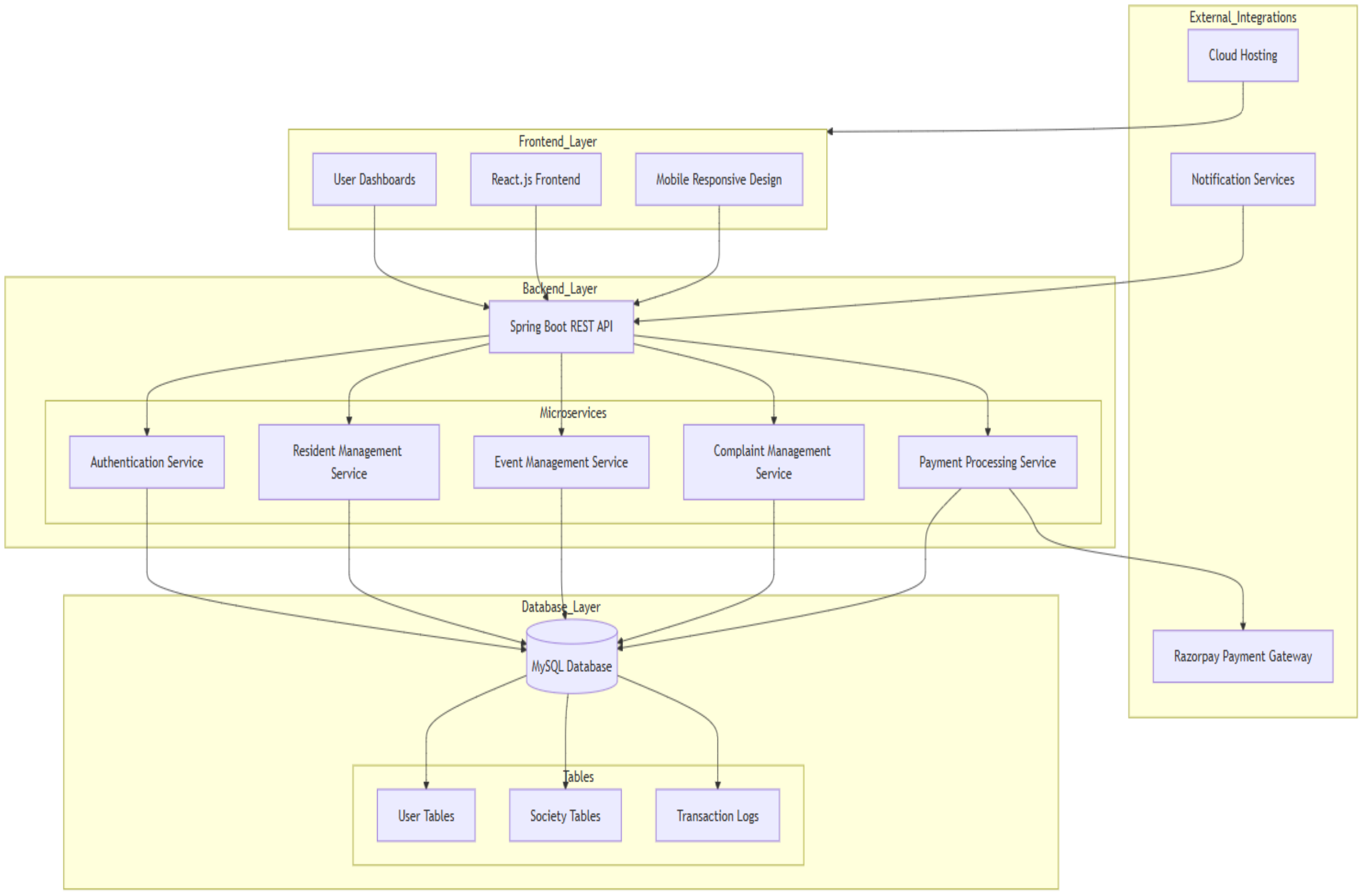
- Standard APIs for third-party integrations
- Seamless integration with payment gateway (Razorpay)
- Support for future integrations with other community services
- Standard data exchange formats (JSON/XML)

8. **Recovery and Business Continuity**
- Data backup every 24 hours
- Maximum recovery time objective (RTO) of 4 hours
- Maximum allowable downtime of 1 hour per month
- Automated system health monitoring and alerts

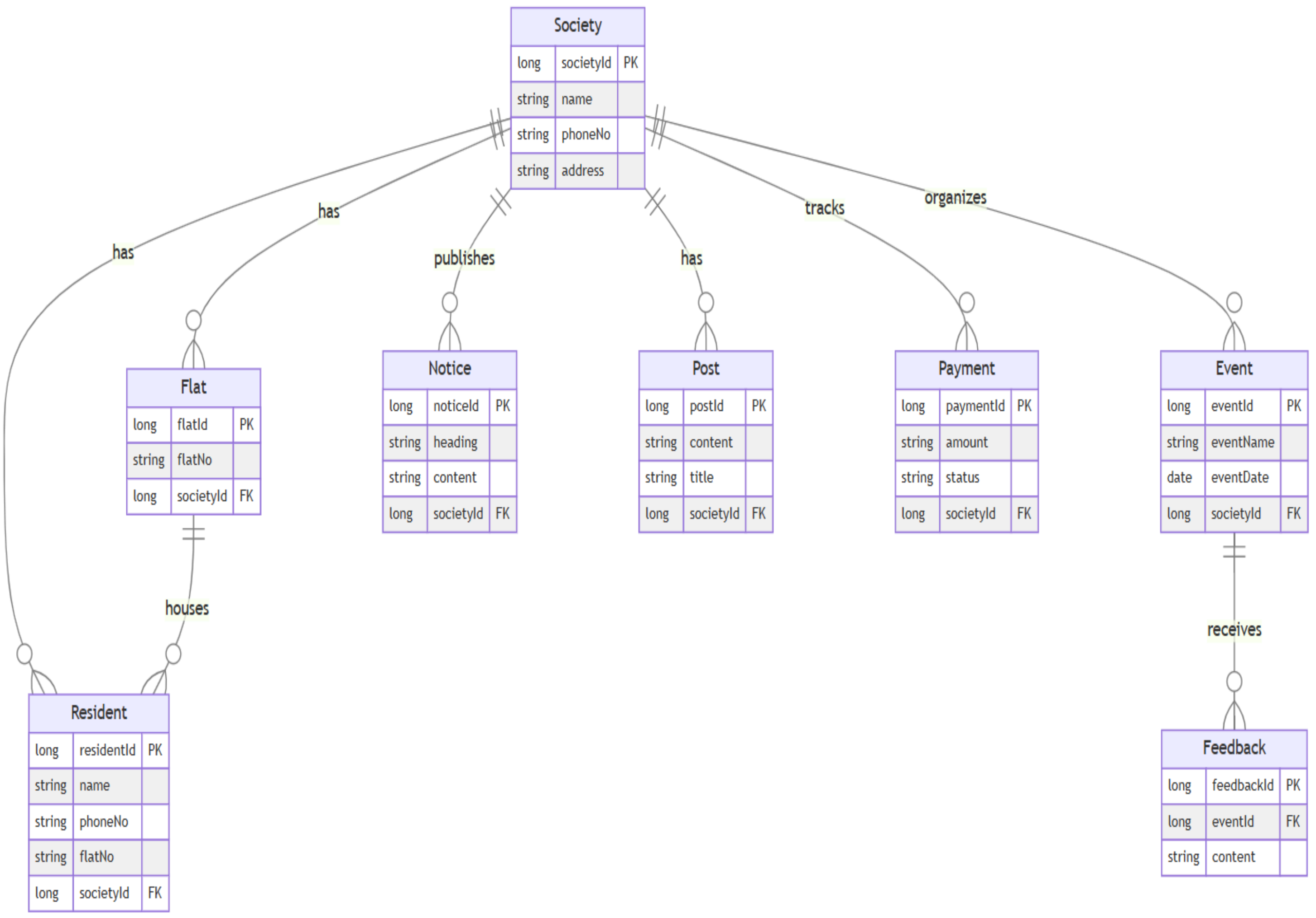
9. **Performance Monitoring**
- Real-time monitoring of system performance
- Regular performance reports and analytics
- Monitoring of payment transaction success rates
- User activity and engagement metrics tracking

10. **Storage and Capacity**
- Sufficient storage capacity for documents and notifications
- Efficient management of database growth
- Archive mechanism for old records
- Support for document uploads with size limitations

6. Project Architecture



7. Database Schema Overview:



8. Assumptions:

The CommUnity platform assumes that all housing societies have reliable internet connectivity and residents have access to internet-enabled devices (smartphones, tablets, or computers) to access the system. It is assumed that society administrators have basic technical literacy to manage the platform and that residents can navigate web-based applications. The system presumes that each housing society has a formal organizational structure with designated administrators who will have elevated privileges within the platform. Furthermore, it is assumed that the society maintains accurate and up-to-date records of all residents, including their contact information and flat details, which will be necessary for initial system setup and ongoing management.

The platform assumes that all financial transactions for maintenance payments will be processed through digital payment methods via Razorpay integration, and residents have access to digital payment options (credit cards, debit cards, or UPI). It is presumed that the housing society has established maintenance fee structures and payment cycles that can be standardized within the system. For security purposes, it is assumed that each resident will maintain the confidentiality of their login credentials and promptly report any unauthorized access. The system also assumes that the society has standardized processes for handling complaints and service requests, and has established relationships with vendors and service providers who can be integrated into the complaint management workflow. Additionally, it is presumed that the society follows local regulations and guidelines regarding data privacy and financial transactions, and all users consent to their data being stored and processed within the system.

9.Constraints:

1. Technical Constraints:

- The system must work with standard web browsers (Chrome, Firefox, Safari, Edge) and their previous two major versions only
- The platform must handle multiple concurrent users with a response time not exceeding 3 seconds for regular operations
- Database transactions must comply with ACID properties to ensure data consistency
- The system must maintain encrypted storage of sensitive data (passwords, payment information) using industry-standard encryption methods
- File uploads for notices and events must be limited to common formats (PDF, JPG, PNG) with a maximum size of 10MB per file

2.Security Constraints:

- User passwords must meet minimum complexity requirements (8 characters, including uppercase, lowercase, numbers, and special characters)
- User sessions must timeout after 30 minutes of inactivity
- Failed login attempts must be limited to 3 tries before temporary account lockout
- All API endpoints must be secured with JWT authentication
- User data must be backed up daily with a retention period of at least 30 days

3.Operational Constraints:

- The system must support a maximum of 1000 societies with up to 500 residents each
- Maintenance payment records must be retained for a minimum of 7 years for audit purposes

4.Administrative Constraints:

- Each society must have at least one primary administrator and one backup administrator
- Only society administrators can add or remove residents from the system
- Changes to maintenance fee structures require approval from authorized administrators
- Service request resolution time must not exceed 48 hours
- Historical data must be archived after 2 years of inactivity

5. Legal and Compliance Constraints:

- The system must comply with local data protection laws and regulations
- User consent must be obtained for data collection and processing
- System logs must be maintained for all financial transactions for 7 years
- Personal information can only be accessed by authorized personnel

6. UI/UX Constraints:

- All critical features must be accessible within 3 clicks from the dashboard
- The mobile interface must support both portrait and landscape orientations
- Error messages must be clear and provide actionable information to users

7.Data Constraints:

- Resident profiles must include mandatory fields (name, flat number, contact number, email)
- Event notices must be posted at least 48 hours before the scheduled time
- Complaint descriptions must be between 50-500 characters
- Payment records must maintain audit trails of all modifications
- Document storage must not exceed 5GB per society

8.Performance Constraints:

- The system must support up to 1000 concurrent users
- Page load time must not exceed 3 seconds on 4G connections
- Database queries must be optimized to execute within 100ms
- API endpoints must handle requests with a maximum latency of 200ms
- The system must maintain 99.9% uptime during operational hours

9.Integration Constraints:

- Payment gateway integration must be limited to Razorpay API
- SMS and email notifications must use approved service providers
- Third-party integrations must be versioned and maintain backward compatibility
- API rate limiting must be implemented to prevent abuse
- External service dependencies must have fallback mechanisms

10.Data Requirements:

1. **Resident Profile Data**
   - Unique user ID
   - Full name (first name, middle name, last name)
   - Email address (unique)
   - Mobile number
   - Password (encrypted)
   - Role (resident/admin/super admin)
   - Registration date
   - Last login timestamp
2. **Society Data Requirements**
   - Unique society ID
   - Society name
   - Complete address
   - Registration number
   - Total number of flats
   - Number of blocks/wings
   - Society registration documents
   - Contact information
   - Banking details for maintenance payments
3. **Flat/Unit Data Requirements**
   - Unique flat ID
   - Flat/Unit number
   - Block/Wing identifier
   - Floor number
   - Type (1BHK/2BHK/3BHK)
   - Current occupancy status
   - Parking slot numbers assigned
   - Maintenance rate category
4. **Event Management Data**
   - Event ID
   - Event title
   - Description
   - Date and time
   - Venue
   - Organizer details

   - Maximum participants
   - Registration deadline
   - Event type (social/cultural/emergency/meeting)
   - Event status (upcoming/ongoing/completed)
   - Participant list
   - Feedback and ratings
   - Associated documents/images
5. **Notice Board Data**
   - Notice ID
   - Title
   - Content
   - Priority level (urgent/normal)
   - Publication date
   - Expiry date
   - Target audience (all/specific blocks)
   - Attachments
   - Posted by (admin details)
   - Read receipt tracking
6. **Complaint/Service Request Data**
   - Ticket ID
   - Category (maintenance/security/housekeeping)
   - Subject
   - Description
   - Submitted by
   - Submission date and time
   - Priority level
   - Status (open/in-progress/resolved/closed)
   - Assigned to
   - Resolution details
   - Resolution time
   - Feedback rating
   - Attached images/documents
   - Communication history
7. **Maintenance Payment Data**
   - Payment ID
   - Flat ID
   - Bill period (month/year)

- Amount due
- Due date
- Payment status
- Payment date
- Transaction ID
- Payment method
- Late payment penalties
- Receipt number
- Payment history
- GST details
- Payment breakdown (water/electricity/maintenance)

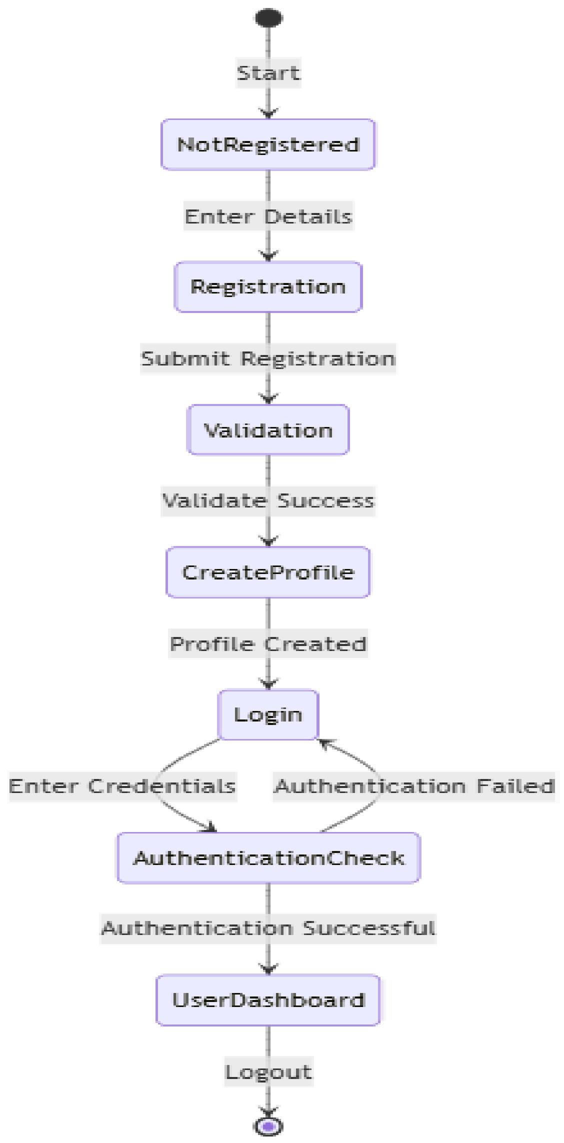
8. **Security Management Data**
    - Guard details (ID, name, shift timings)
    - Visitor logs
    - Entry/exit timestamps
    - Vehicle logs
    - Security incident reports
    - Emergency contact numbers
    - CCTV footage references
    - Patrol schedules
    - Access card details
9. **Communication Data**
    - Message ID
    - Sender details
    - Recipient details
    - Message type (email/SMS/in-app)
    - Content
    - Timestamp
    - Status (sent/delivered/read)
    - Priority level
    - Attachments
    - Delivery reports
10. **Document Management Data**
    - Document ID
    - Title
    - Category
    - Upload date

- Uploaded by
- File type
- File size
- Access permissions
- Version history
- Retention period
- Document status (active/archived)

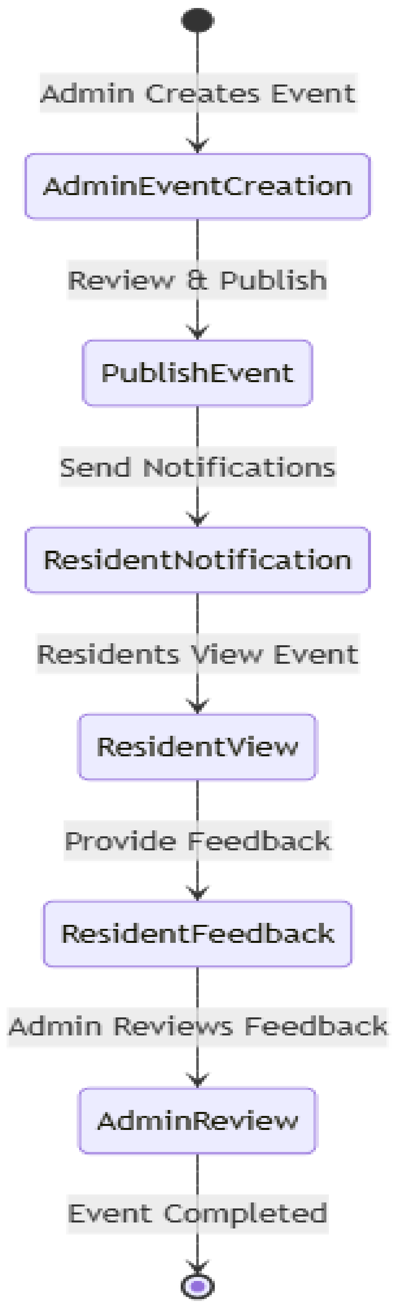
11. **System Configuration Data**
    - System parameters
    - Email templates
    - SMS templates
    - Notification settings
    - User role permissions
    - Feature toggles
    - API configurations
    - Integration settings
    - Backup schedules

11. Workflow Diagrams:

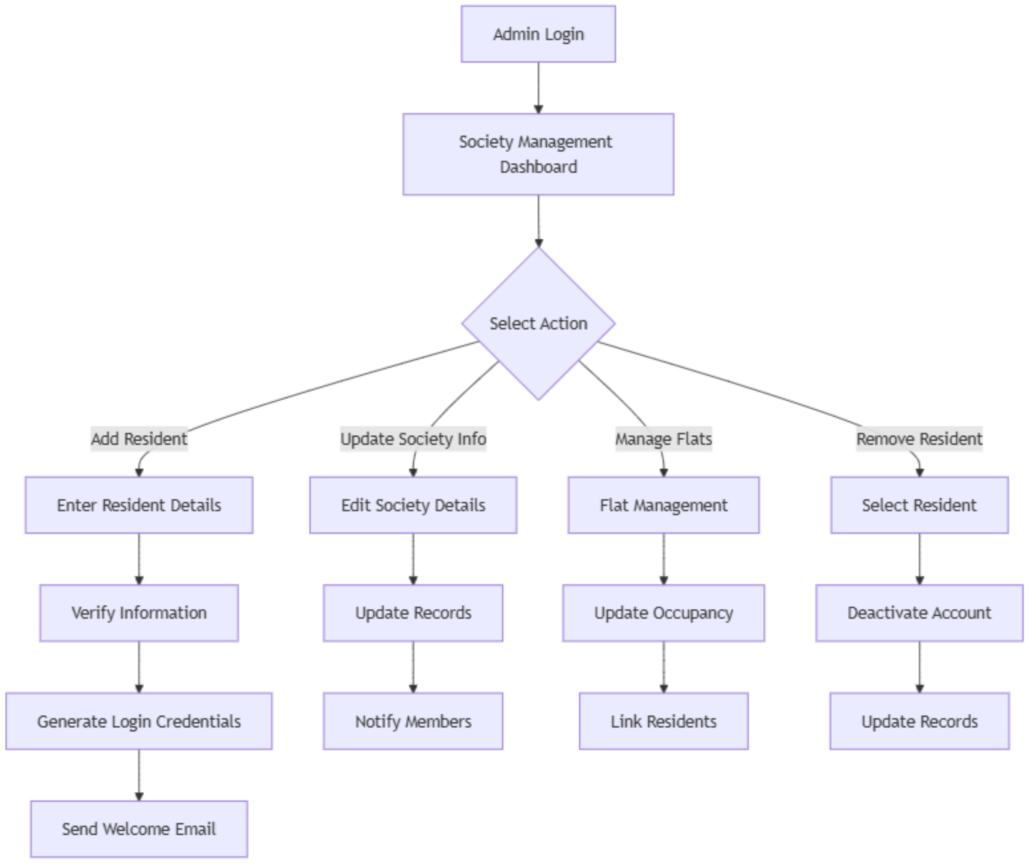
1. User Authentication Workflow



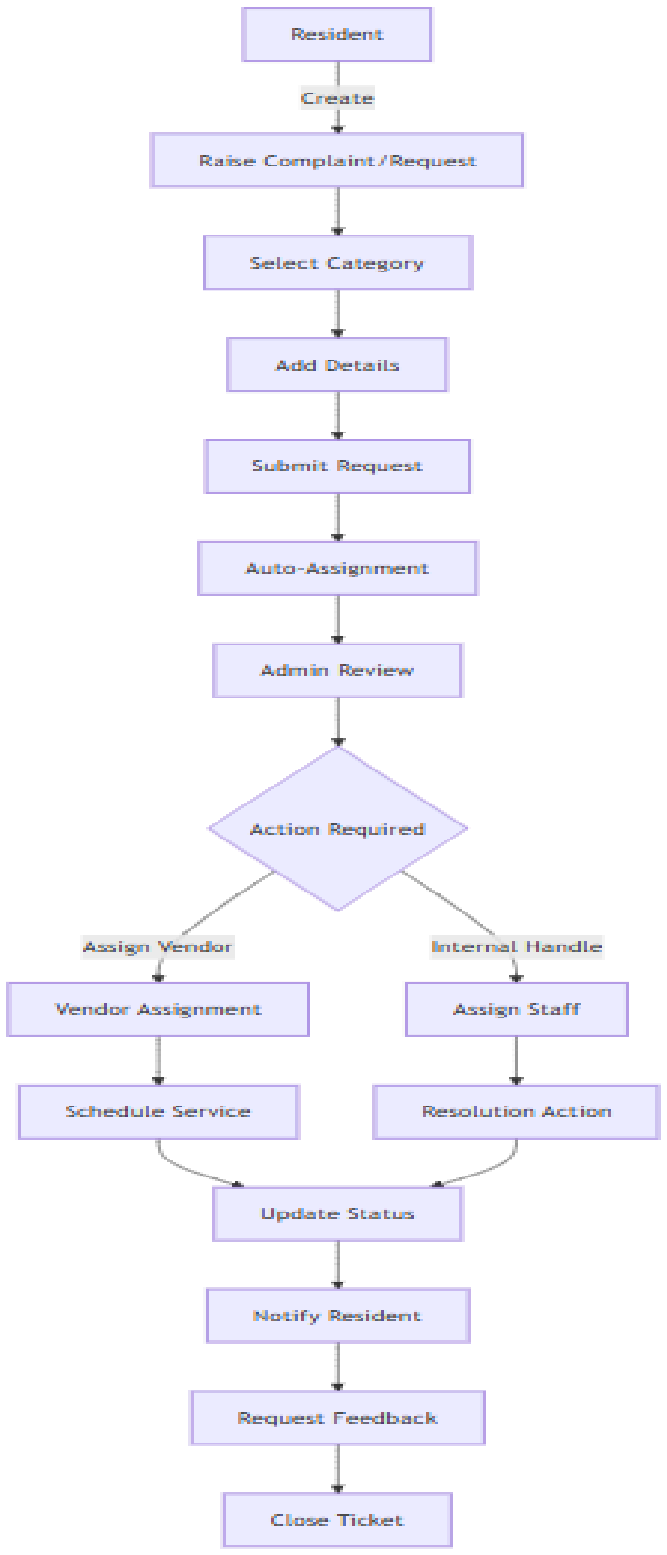
2. Event Management Workflow



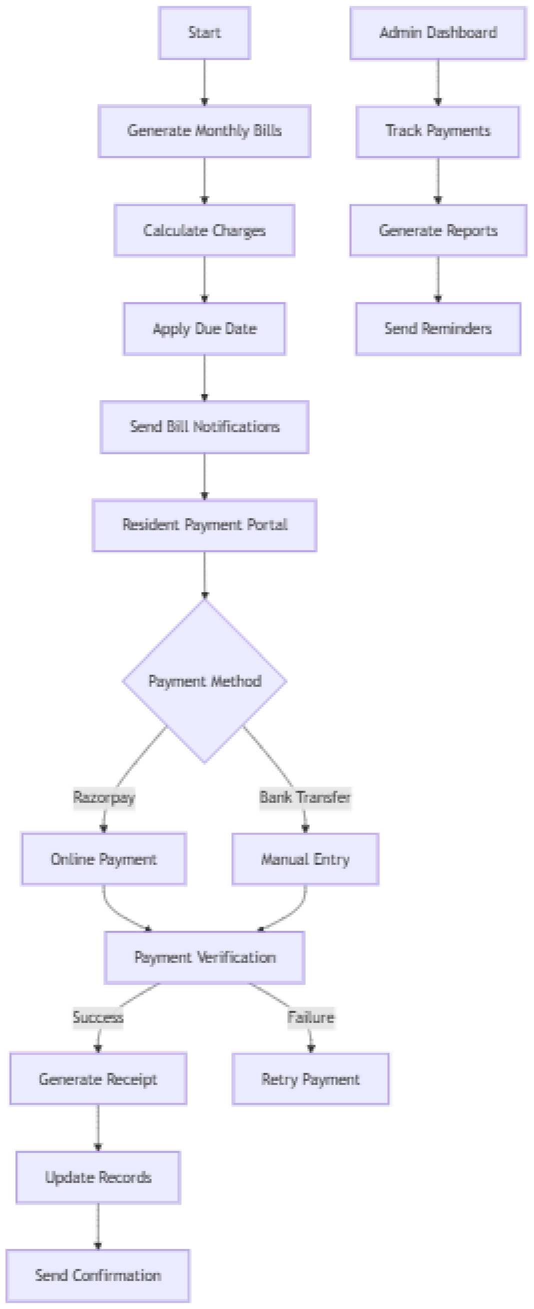
3.Resident and Society Management workflow:



4. Complaint and Service Request Management:



5.Maintenance Payment System Workflow:


Start
Generate Monthly Bills
Calculate Charges
Apply Due Date
Send Bill Notifications
Resident Payment Portal
Payment Method
Razorpay
Bank Transfer
Online Payment
Manual Entry
Payment Verification
Success
Failure
Generate Receipt
Retry Payment
Update Records
Send Confirmation
Admin Dashboard
Track Payments
Generate Reports
Send Reminders