Functional and Non-Functional Requirements

1. FUNCTIONAL REQUIREMENTS

1.1 User Management

- User Registration: The system shall allow users to register with their details, including username, password, email, mobile number and full name.
- User Authentication: The system shall provide login functionality using the registered email and password.
- Profile Management: Users shall be able to view and update their profiles.

1.2 Announcement & Discussion Management

- Create Announcements: Admins shall be able to create announcements with a title, content, and timestamp.
- Role-Based Access: The system shall support different roles (e.g., Admin, Member) with varying levels of access to functionalities.
- Manage Discussions: Users shall be able to create discussion groups, invite members, and assign admins.
- Poll Creation: Admins shall be able to create polls within announcements or discussion groups with multiple options and an end time.
- Voting: Users shall be able to vote in polls, with the system recording and displaying the results.
- Messaging: Members of a discussion group shall be able to send and view messages within the group.
- Invite Members: Admins of a discussion group shall be able to invite other users to join the group.

1.3 Feedback & Document Management

- Submit Feedback: Users shall be able to submit feedback, including text content and a timestamp.

1.4 Media Management

- Upload Documents: Users shall be able to upload documents, which will be stored in an S3 bucket.
- Document Management: The system shall allow users to view, download, and manage uploaded documents.
- Event Calendar Creation: Admins shall be able to create events with a name, date, and associated photo galleries.

- Photo Gallery Management: Users shall be able to create photo galleries, upload photos, and manage them (add, remove, update).
- Photo Upload: The system shall support uploading photos to a gallery, with each photo containing metadata such as URL, name, and upload timestamp.

1.5 Parking & Guest Management

- Guest Management: The system shall allow users to register guests.
- Parking Slot Allocation: The system shall allow users to allocate and manage parking slots for registered guests, including start and end times for parking.
- Parking Management: Admins shall be able to view and manage parking slots, ensuring optimal usage.

1.6 Payment Management

- Payment Processing: The system shall allow users to make payments for services, fees, or other charges.
- Payment Status Tracking: Users shall be able to view their payment history, including payment IDs, amounts, dates, and statuses.
- Invoice Generation: The system shall generate invoices for payments made and provide them to users.

1.7 Notification Management

- Emergency Notifications: The system shall allow admins to send emergency notifications to all users.
- Group Notifications: The system shall support sending notifications to specific discussion groups or user segments.
- Event Reminders: Users shall receive reminders for upcoming events in the event calendar.

1.8 Voting and Polling

- Create Polls: Admins shall be able to create polls with multiple options within announcements or discussion groups.
- Vote in Polls: Users shall be able to vote on polls within the allowed time frame.
- View Poll Results: Users shall be able to view the results of polls after voting has ended.

1.9 System Integration

- Integration with Third-Party Services: The system shall integrate with third-party services such as payment gateways, email services, and S3 storage.
- API Integration: The system shall provide RESTful APIs for third-party integration and allow other services to interact with it.

2 . NON-FUNCTIONAL REQUIREMENTS

2.1 Security

- Basic Encryption: Sensitive data should be encrypted in transit using standard HTTPS.
- Role-Based Access: The system should ensure that only authorized users can access certain features.

2.2 Usability

- User Interface: The application should be easy to navigate on desktop.
- Responsiveness: The application should be responsive, providing a consistent and optimal user experience across various screen sizes and devices.
- Basic Accessibility: The application should be accessible to a wide range of users, with simple and clear layouts.

2.3 Maintainability

- Code Simplicity: The code should be well-organized and documented to allow easy updates and maintenance.

2.4 Scalability

- Basic Scalability: The system should be able to add more users with minimal performance issues by scaling the infrastructure as needed.

2.5 Interoperability

- API Integration: The system should provide simple RESTful APIs for integration with other services.

2.6 Compliance

- Basic Data Protection: The system should comply with basic data protection regulations, ensuring user privacy is respected.