

Solution Requirements

Date	01/11/2025
Team ID	NM2025TMID02331
Project Name	Lease management

Functional Requirements

Functional requirements define what the system **must do** — the specific functions, features, and business rules.

Lease Management

1. The system must allow users to create, view, update, and delete lease records.
2. Each lease record must include property details, tenant information, start and end dates, rent amount, and payment frequency.
3. The system should automatically calculate lease duration and rent totals based on input values.
4. The system must allow uploading and attaching lease documents (PDF, Word, images).

Tenant Management

1. The system must store and manage tenant details including contact information, address, and ID proofs.
2. The system should track tenant history, previous leases, and payment performance.
3. The system should send automated notifications to tenants for rent dues or renewal reminders.

Payment and Billing

1. The system should generate automated rent invoices based on the lease agreement.
2. The system must support payment recording and maintain a transaction log for all payments made.
3. The system should calculate penalties or late fees for overdue payments.
4. The system must generate monthly and yearly financial statements.
5. Users should be able to export payment history in Excel or PDF formats.

Lease Renewal and Termination

1. The system must notify property managers about upcoming lease expirations.
2. The user should be able to process lease renewal or termination through the dashboard.
3. Upon renewal, a new record should be created automatically with updated dates and terms.

Data Analytics and Reporting

1. The system must analyze lease data to provide insights such as total revenue, occupancy rates, and pending payments.
2. It should generate dynamic reports and visual charts showing trends over time.
3. The reports should be customizable by date, property, or tenant.

Search and Filter

1. Users should be able to search records by lease ID, tenant name, property, or date.
2. Advanced filters should allow narrowing results by payment status, lease type, or duration.

Access Control and User Roles

1. The system should support multiple user roles — Admin, Property Manager, Accountant, and Viewer.
2. Each role should have defined permissions (e.g., view-only, edit, delete).
3. Admins should be able to add, update, or deactivate user accounts.

System Integration

1. The system should integrate with accounting or CRM tools (e.g., Salesforce, QuickBooks).
2. It should allow data import/export using CSV or API.

Notification and Alert System

1. Automatic alerts should be generated for due payments, upcoming renewals, or expired leases.
2. Notifications should be sent via email or dashboard messages.

Audit and Logging

1. Every user activity (record creation, update, or deletion) should be logged.
2. The system should maintain an audit trail for compliance and tracking purposes.

Non-Functional Requirements

Non-functional requirements define **how** the system performs — focusing on quality, performance, and reliability.

Performance

1. The system should handle at least 500 active leases and 100 concurrent users without performance degradation.

2. Response time for most operations (like searching, updating) should be less than 3 seconds.

Security

1. All sensitive data (tenant info, payments) must be encrypted in the database.
2. Passwords must be stored using hashing and salting techniques.
3. The system should implement role-based access and restrict unauthorized actions.
4. All data transmission should occur over HTTPS.

Reliability

1. The system should have an uptime of 99.9%.
2. Backup should occur automatically at least once a day.
3. In case of a failure, recovery time should be less than 1 hour.

Usability

1. The system interface must be simple, intuitive, and consistent.
2. Help tooltips and user guides should be provided for each module.
3. Color codes and alerts should highlight pending payments or expired leases.

Scalability

1. The system should easily scale to support more properties, users, or geographical locations.
2. Cloud-based deployment should be supported for expansion.

Maintainability

1. The system should be modular with separate components for lease, tenant, and payments.
2. Code documentation and configuration settings should allow easy updates or bug fixes.

Compatibility

1. The system should support major browsers (Chrome, Firefox, Edge).
2. It should be responsive and accessible from desktops, tablets, and mobile devices.

Data Integrity

1. The system should validate all input fields to prevent data duplication or errors.
2. Data consistency must be maintained across all modules.

Compliance

1. The system should comply with local property laws and data protection regulations (e.g., GDPR).
2. Audit logs should be preserved for at least 12 months.

Availability

1. Scheduled maintenance should not exceed 2 hours per month.
2. The system should support failover mechanisms in case of server downtime.