    - Create, view, edit, and delete property records.
    - Track details: property name, address, type, ownership, status.
2. **Tenant Management**
    - Maintain tenant profiles with contact info, lease history, and documents.
    - Link tenants to properties and leases.
3. **Lease Agreement Tracking**
    - Generate new lease agreements.
    - Track start/end dates, duration, terms, rent amount, status (active/expired).
    - Upload scanned copies or PDFs.
4. **Rent/Payment Tracking**
    - Record monthly/quarterly payments.
    - Set due dates and automatic payment reminders.
    - Integrate with Salesforce Flow to notify tenants or admin for late payments.
5. **Renewals & Terminations**
    - Track lease expiry dates and send renewal alerts.
    - Allow early termination with reason tracking.
6. **Reporting and Dashboards**
    - Generate reports for:
        - Active/expired leases
        - Payment history
        - Property occupancy
    - Visual dashboards for lease health and payment status.
7. **User Roles and Permissions**
    - Admin: Full access
    - Property Manager: Manage leases & payments
    - Tenant (Community User): View lease & payment status

Non-Functional Requirements

These refer to **how** the system should perform.

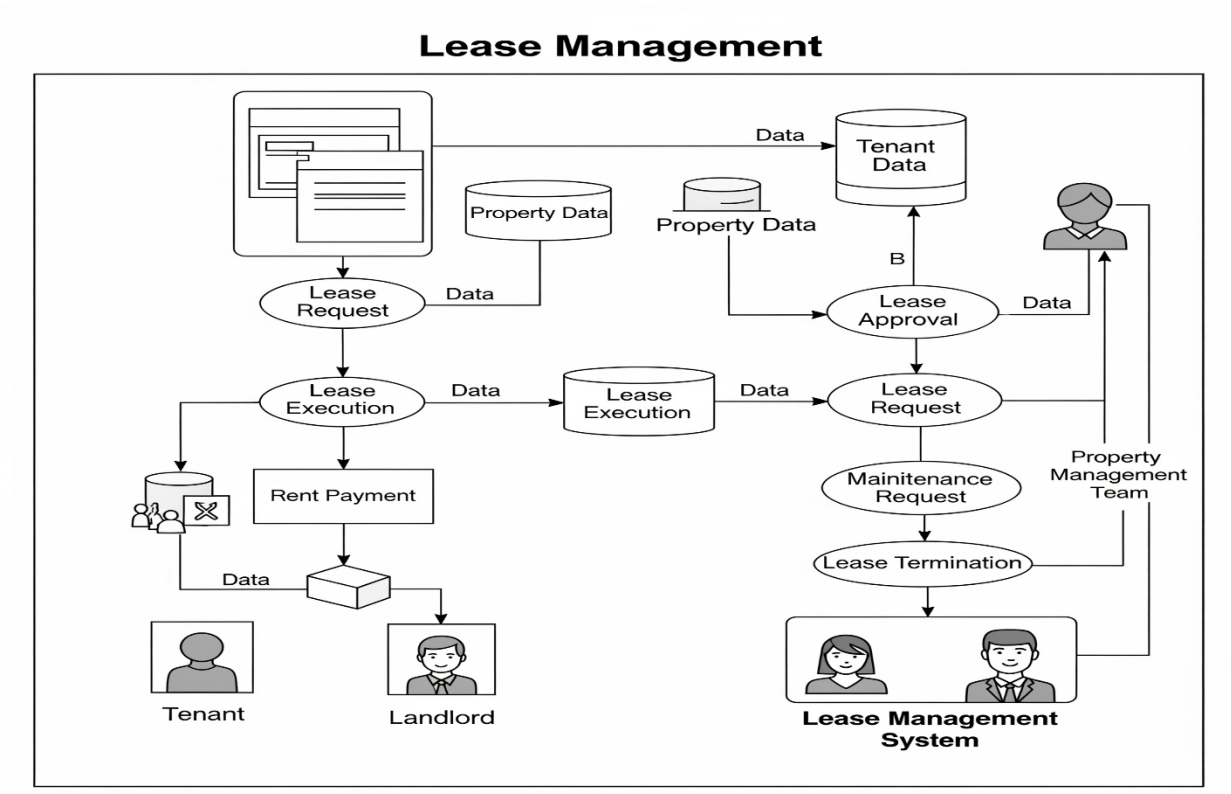
1. **Scalability**
    - Should support 100+ properties and tenants with no performance lag.
2. **Usability**
    - User-friendly interface with clear navigation for both admin and tenants.
3. **Data Security**
    - Only authorized users can access/edit lease and payment records.
    - Role-based sharing and field-level security enabled.
4. **Availability**
    - 99.9% uptime – accessible 24/7 via Salesforce Cloud.
5. **Integration**
    - Integration with email (Outlook/Gmail) and calendar for reminders.
    - Optional: Payment gateway (e.g., Razorpay, Stripe).
6. **Automation**

- Salesforce Flows to automate:
  - Payment reminders
  - Lease renewal notices
  - Monthly reports

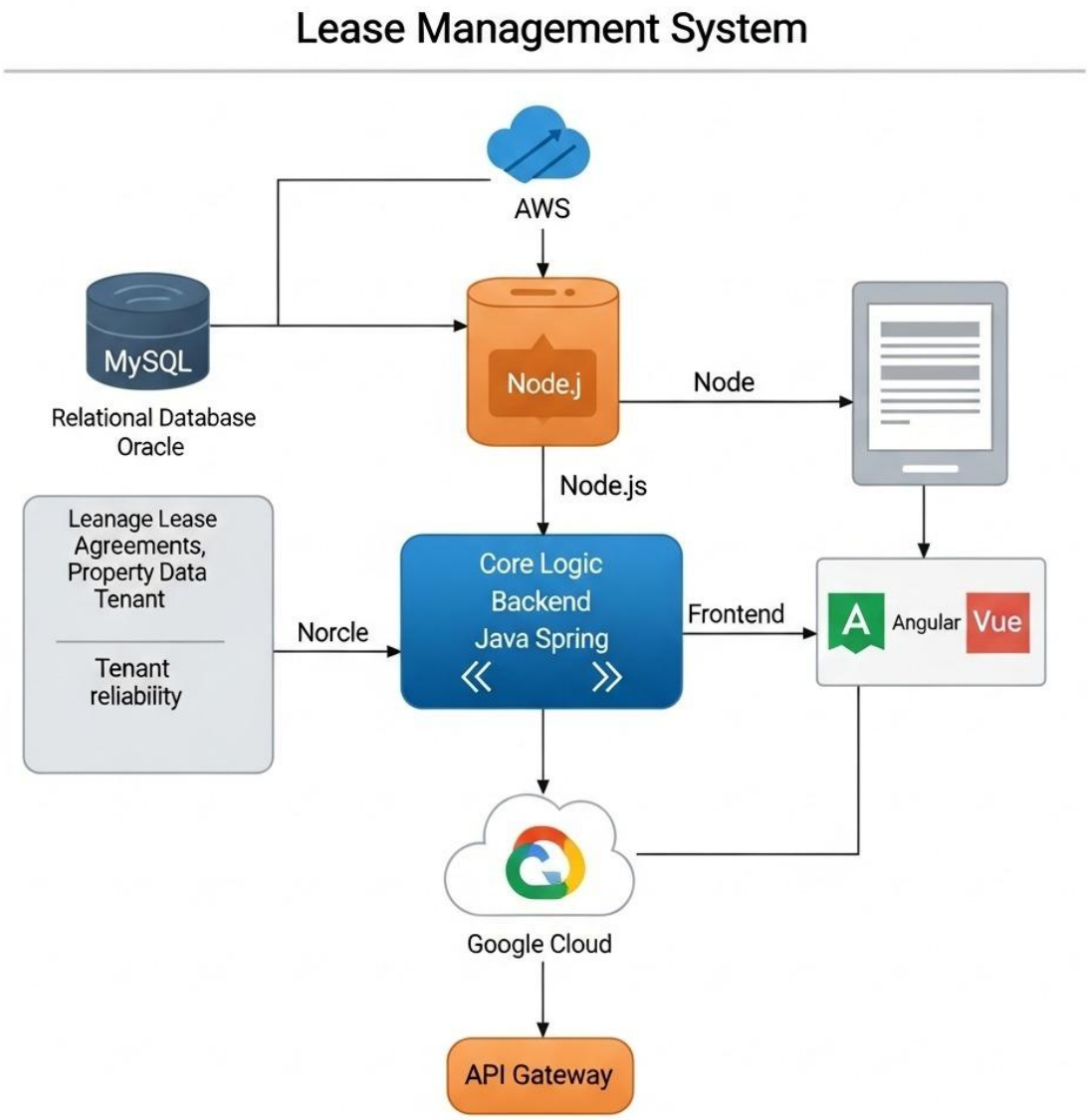
System Requirements (Salesforce Specific)

Component	Requirement
Salesforce Edition	Enterprise/Developer
Objects Used	Custom Objects: `Property`, `Tenant`, `Lease`, `Payment`
Automation Tools	Flows, Process Builder, Approval Processes
Reports & Dashboards	Custom report types and dashboards
Page Layouts	Record pages customized using Lightning App Builder
App Type	Lightning App

3.3 Data Flow Diagram



3.4 Technology Stack:



The Lease Management System is built using tools and technologies provided by Salesforce. The main technologies used are:

1. Salesforce Platform

- Main platform used for building the application.
- Provides tools for development, automation, and data storage.

2. Frontend (User Interface)

- **Lightning App Builder** – to design pages.

- **Lightning Web Components (LWC)** – to create custom components.
- **Tabs & Page Layouts** – to organize data views.

3. Backend (Logic & Automation)

- **Apex** – used to write custom logic (classes, triggers).
- **SOQL** – used to query data from Salesforce objects.
- **Flow Builder** – to automate tasks without code (e.g., send emails, update fields).

4. Database

- **Custom Objects** – like Lease, Tenant, Property, and Payment to store project data.
- **Standard Objects** – like Account and Contact, used if needed.

5. Integration (if required)

- **REST API / SOAP API** – to connect with other apps.
- **Named Credentials** – to store external API login info securely.

6. Development Tools

- **Developer Console** – for writing and testing code.
- **VS Code + Salesforce CLI** – for professional development.
- **Change Sets** – for moving changes from sandbox to production.

7. Testing Tools

- **Apex Test Classes** – to test code.
- **Debug Logs** – to find and fix errors.
- **Workbench** – to run queries and test APIs.