

Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	2 November 2025
Team ID	NM2025TMID03296
Project Name	Lease Management System
Maximum Marks	4 Marks

Technical Architecture

The **Lease Management System Architecture** ensures smooth automation of property and lease operations using Salesforce's **Lightning Platform**. It integrates UI, business logic, and cloud data layers within a single ecosystem.

When a user creates a **lease record**, validation rules verify input data (like start and end dates). If valid, the record is processed and linked to the respective **tenant and property**. A **workflow rule** triggers an email notification to the admin for approval. Once approved, payment records are created and reflected in reports and dashboards for real-time insights.

The system's architecture emphasizes **automation, scalability, and reliability** — reducing manual effort and improving data integrity across all lease operations.

Example – Solution Architecture Diagram

Figure 1: Architecture and data flow of the Lease Management System using Salesforce Lightning.

Reference:

<https://developer.salesforce.com/docs/platform>

Technical Architecture: Lease Management System
Built on Salesforce Lightning Platform

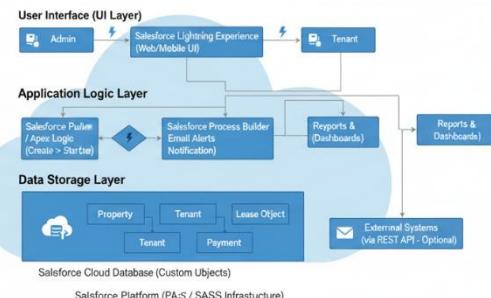


Table-1: Components & Technologies

S.No.	Description	Technology
S.No. 1	User Interface	Salesforce Lightning Experience
S.No. 2	Database	Salesforce Cippleset / Relationships
S.No. 3	External API (Optional)	Salesforce REST API Relationships Infrastructure
S.No. 4	Infrastructure	(PaaS / SaaS)

Table-2: Application Characteristics

S-No.	Description	Characteristics	Technology
S-No. 1	Scalable Architecture Salesforce Multi-Tenant Cloud Architecture Salesforce REST API Integration Infrastructure	Scalable Architecture Description Salesforce Profiles, Permission Sets	Implementation Scenarios

Table-1: Components & Technologies

S.No	Component	Description	Technology
1.	User Interface	Admin and tenants interact via the Salesforce Lightning App UI.	Salesforce Lightning Experience
2.	Application Logic-1	Handles creation, update, and validation of property, tenant, and lease records.	Salesforce Flow Builder, Apex Logic
3.	Application Logic-2	Executes validation rule: <i>End Date must be greater than Start Date.</i>	Validation Rule (Salesforce Declarative Tool)
4.	Application Logic-3	Manages tenant approval process and sends email notifications.	Salesforce Process Builder, Email Alerts
5.	Database	Stores all object data (Property, Tenant, Lease, Payment) and relationships.	Salesforce Cloud Database (Custom Objects)
6.	Cloud Database	Fully managed Salesforce backend ensures scalability and availability.	Salesforce Cloud Infrastructure
7.	File Storage	Used to store attachments like lease documents or payment receipts.	Salesforce File Storage / Content Library
8.	External API-1 (Optional)	Enables integration with third-party payment systems or property portals.	Salesforce REST API Integration
9.	External API-2	Not applicable for the current implementation.	—
10.	Machine Learning Model	Not applicable for current use case; can be extended later for predictive rent analysis.	—
11.	Infrastructure (Server / Cloud)	Hosted, maintained, and secured on Salesforce Cloud (SaaS Platform).	Salesforce Cloud (PaaS / SaaS)

Table-2: Application Characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Not applicable — Salesforce uses proprietary Lightning and Apex frameworks.	—
2.	Security Implementations	Role-based access controls, record-level security, and authentication for admin users.	Salesforce Profiles, Permission Sets, Authentication Policies
3.	Scalable Architecture	Cloud-based architecture with scalability for multiple tenants and property records.	Salesforce Multi-Tenant Cloud Architecture
4.	Availability	Highly available through Salesforce's globally distributed and load-balanced servers.	Salesforce Cloud Hosting (SaaS)
5.	Performance	Optimized through indexed queries, asynchronous processes, and Flow Builder automation.	Apex Triggers, Flow Builder, Indexed Database Tables