|  |  |
| --- | --- |
| **Date** | 3 August 2025 |
| **Team ID** | LTVIP2025TMID30830 |
| **Project name** | Lease management |
| **Maximum Marks** |  |

**📌 Project Name:**

**Lease Management System**

**🧩 Team ID:**

**LTVIP2025TMID30830**

**📅 Date:**

**3 August 2025**

**📝 1. Introduction**

This document outlines the architecture of the Lease Management System built on the Salesforce platform. The architecture is designed to deliver a modular, automated, and scalable solution that addresses the key business needs of lease tracking, tenant management, rent collection, and automated communication. It ensures performance, reliability, security, and future extensibility.

**⚙️ 2. Architectural Overview**

The Lease Management System is implemented using the **Salesforce Lightning Platform** and is composed of the following layers:

**🔹 Architecture Layers:**

| **Layer** | **Component** | **Purpose** |
| --- | --- | --- |
| **Presentation Layer** | Lightning App & Tabs | User Interface for admins, managers, and staff |
| **Application Layer** | Apex Classes, Triggers, Flows | Enforce business rules, automation, and validations |
| **Data Layer** | Salesforce Custom Objects | Store data related to Properties, Tenants, Leases, and Payments |
| **Communication Layer** | Email Templates & Alerts | Send reminders, approvals, rejections, and confirmations |
| **Scheduler Layer** | Apex Scheduler | Automate monthly rent reminders |
| **Security Layer** | Profiles, Roles, Field-level Security | Control access and ensure data protection |

**📦 3. System Components**

**🔹 Custom Objects**

* **Property**: Stores property details (name, type, address, area)
* **Tenant**: Tenant information and lease status
* **Lease**: Lease agreements with start and end dates
* **Payment for Tenant**: Payment status, amount, and date

**🔹 Relationships**

* **Master-Detail and Lookup** relationships between:
  + Tenant → Property
  + Lease → Tenant
  + Payment → Tenant & Property

**🔁 4. Automation and Logic**

**🔸 Apex Triggers**

* Prevent multiple tenants from being assigned to the same property

**🔸 Validation Rules**

* Lease end date must be after the start date
* Mandatory fields must be filled before submission

**🔸 Approval Process**

* Approval workflow for tenant requests to end or extend leases

**🔸 Scheduled Apex**

* Sends monthly email reminders to tenants on the 1st of each month

**📩 5. Communication System**

| **Email Template** | **Use Case** |
| --- | --- |
| Tenant Leaving | Request for lease termination |
| Leave Approved | Approval of lease end request |
| Leave Rejected | Rejection with reasons |
| Monthly Rent Reminder | Auto-sent on the 1st of each month |
| Payment Confirmation | Sent after successful rent payment |

**🔐 6. Security Model**

* **Role-based access control**
* **Profile-level permissions**
* **Object and Field-Level Security (FLS)**
* **Audit Trails and History Tracking**

**🌐 7. Scalability & Extensibility**

**Current Scale:**

* Supports 1000+ properties and tenants

**Future-Ready Integrations:**

* Payment gateways (e.g., Razorpay, Stripe)
* Mobile App
* AI/ML for lease forecasting and analytics
* Tenant self-service portal

**🧾 8. Summary Diagram**

Would you like a **visual diagram** showing:

* Component interactions
* Data flows
* User roles
* Object relationships?

I can generate it for you.

**✅ 9. Conclusion**

The Salesforce-based architecture provides a structured, flexible, and scalable foundation for lease management. It ensures automation, reliability, and ease of use—eliminating manual errors and paving the way for digital transformation in property administration.

Bottom of Form