LEASE MANAGEMENT — Solution Architecture & Project Document

Date: November 01, 2025

1. Project Overview

A lease management project creates a system to efficiently handle leasing processes for real estate, equipment, or other assets. The purpose is to streamline and automate lease lifecycle tasks (from listing and negotiation to renewal and termination), ensure accurate record-keeping, guarantee regulatory compliance, and improve communication between landlords, tenants, property managers, and external stakeholders.

2. Goals of the Architecture

- Centralized lease lifecycle management
- Data integrity & consistency
- Automated workflows
- Auditability & compliance
- Scalability & availability
- Integration-friendly

3. Key Components

User Interface (Web / Mobile), API Layer, Application Services/Business Logic (Lease Management, Tenant Onboarding, Billing & Payment, Document Management, Notifications, Reporting), Database(s), Workflow Engine, Notification Service, Auth & Authorization, Integration Adapters, Monitoring & Observability.

4. Development Phases (Suggested)

1. Requirements & Scope Definition
2. Design & Architecture
3. MVP Implementation
4. Workflow & Automation
5. Integration
6. Testing & Compliance
7. Deployment & Rollout
8. Monitoring & Iteration

5. Solution Architecture Description

Centralizes lease and party data in a transactional database and separates documents into object storage. API layer enforces business rules. Workflow engine runs time-based jobs. Security via centralized authentication and RBAC. Modular design with bounded contexts.

6. Flow Diagram (Visual)

Embedded below is the high-level flow diagram showing front-end, API, business logic, workflow engine, document store, notifications, accounting, and reporting.

7. Functional Requirements (Key)

Create/edit properties and units, Create leases with clauses and rules, Party management, Document management (upload, e-signature, versioning), Billing & payments, Workflow automation, Search & reporting, Role-based access & audit trail.

8. Non-functional Requirements

Performance: API response time < 300ms, Availability: 99.9% uptime, Scalability: horizontal scaling, Security: encryption at rest & transit, Compliance: audit logs retention and export.

9. Data Model (Simplified)

Core entities: Property, Unit, Lease, Party, Payment, Document, AuditLog.

10. Assumptions & Constraints

Payment gateway PCI-compliant, E-signature provider supports REST API, Tenant screening is 3rd party, Multi-currency/tax rules out of scope for MVP unless specified.

11. Risks & Mitigations

Versioning risk -> version control & approvals. Payment failures -> retries & notifications. Data leakage -> encryption & strict RBAC.

12. Conclusion

Modular, secure, and auditable platform enabling incremental delivery of core leasing operations and key integrations.

Flow Diagram