Project Design Phase
Solution Architecture

Date	25 JUNE 2025
Team ID	LTVIP2025TMID30830
Project Name	Lease Management
Maximum Marks	

Solution Architecture:

1. Find the best tech solution to solve existing business problems

Evaluate and select the most appropriate technologies, platforms, and tools that align with the organization's goals for lease management. This includes determining whether to build a custom solution or use a commercial off-the-shelf product, and ensuring compatibility with existing infrastructure. The chosen solution should automate manual processes, reduce errors, enhance data visibility, and streamline operations such as lease creation, tracking, notifications, compliance monitoring, and financial reporting.

2. Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders

Create a comprehensive architectural blueprint that defines how the lease management system will function. This includes system modules (e.g., lease contracts, payments, renewals), user roles (e.g., admin, tenant, property manager), data flow, and external integrations (e.g., with ERP or accounting software). It should explain how the system behaves under various conditions (e.g., overdue rent, contract expiration), ensuring clarity for both technical teams and business stakeholders.

3. Define features, development phases, and solution requirements

Specify the essential features of the lease management system, such as contract generation, automated reminders, document storage, audit logs, and customizable reporting. Clearly outline both functional and non-functional requirements, including security, performance, scalability, and usability. Break the project into development phases (e.g., MVP, pilot, full rollout), with defined milestones and timelines to ensure structured and measurable progress.

4. Provide specifications according to which the solution is defined, managed, and delivered

Document the technical specifications that guide system development and deployment. This includes architecture diagrams, database design, API specifications, user authentication methods, data encryption standards, and hosting infrastructure. Also include delivery guidelines—such as version control, testing protocols, DevOps pipelines, and SLAs—to ensure the solution is maintainable, scalable, and meets all compliance and business expectations.