Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	31 October 2025
Team ID	NM2025TMID05681
Project Name	Garage Management System
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/

Guidelines:

This architecture provides a scalable, secure, and efficient Garage Management System that integrates with external services and leverages machine learning for predictive analytics and decision-making.

Garage Management System

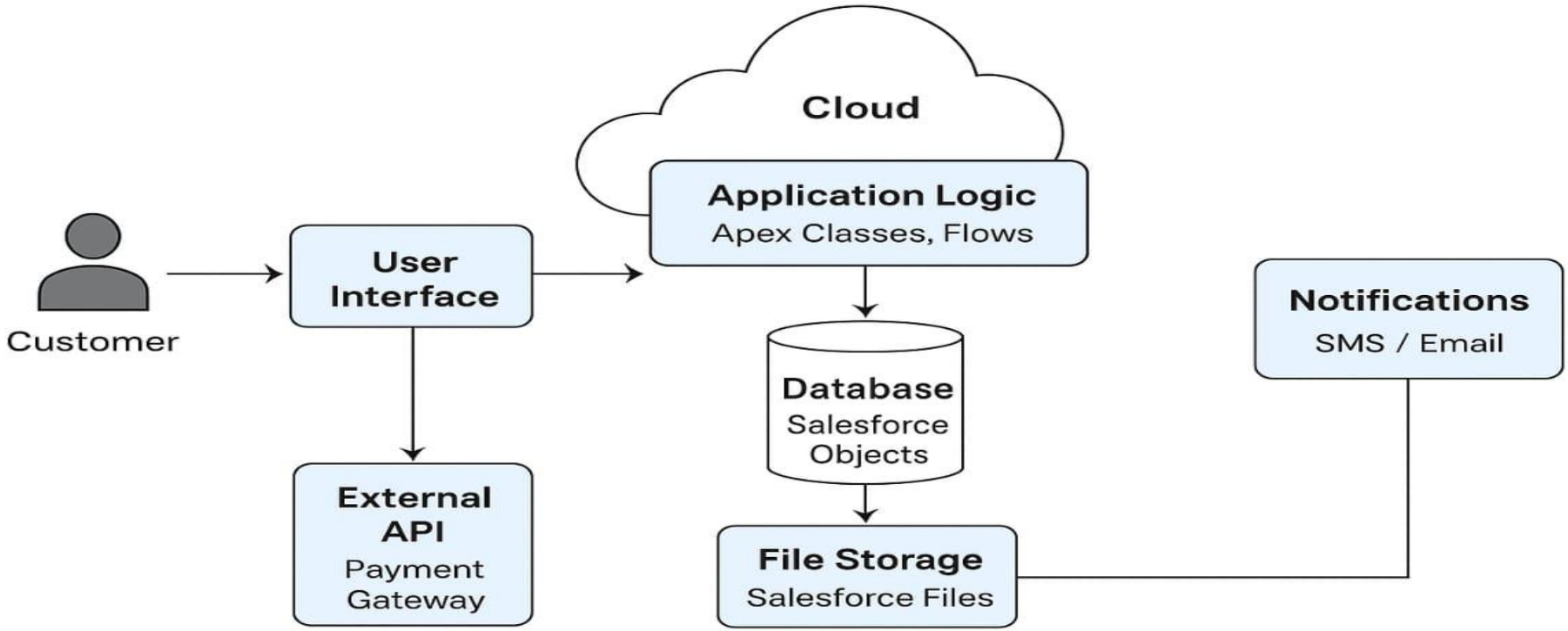

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Service advisors interact via web dashboard	Web UI (React or Angular)
2.	Application Logic-1	Manages service requests and assignments	Node.js, Express.js
3.	Application Logic-2	Validates vehicle history and service status	Database queries (SQL)
4.	Application Logic-3	Sends notifications for service updates	Notification service (Twilio or Nexmo)
5.	Database	Stores customer, vehicle, and service data	Relational database (MySQL or PostgreSQL)
6.	Cloud Database	Managed database service for scalability	AWS RDS or Google Cloud SQL
7.	File Storage	Stores vehicle inspection reports and documents	Cloud storage (AWS S3 or Google Cloud Storage)
8.	External API-1	Integrates with OEM parts suppliers for inventory updates	REST API (Supplier API)
9.	External API-2	Utilizes machine learning models for predictive maintenance	TensorFlow or PyTorch
10.	Machine Learning Model	Not applicable for current use case	-
11.	Infrastructure (Server / Cloud)	Hosted on a cloud platform for scalability	AWS or Google Cloud Platform

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilizes open-source frameworks for flexibility	React, Node.js, Express.js
2.	Security Implementations	Role-based access control, data encryption, secure APIs	OAuth, SSL/TLS, Access Control Lists
3.	Scalable Architecture	Cloud-based, horizontally scalable	AWS or Google Cloud Platform, Load Balancing
4.	Availability	Highly available with cloud hosting and redundancy	Load-balanced servers, Auto Scaling
5.	Performance	Optimized via caching, indexing, and efficient database queries	Redis, MySQL or PostgreSQL, Query Optimization