RENTO TECH – Rent a Car Management System

NAME: ABDUL MANNAN

ID : F2024065243

SUBJECT: SOFTWARE ENGINEERING

NAME OF PROJRCT: RenTo Tech

- Scope:
- The system will manage car rentals, including vehicle inventory, customer records, bookings, payments, and returns.
- It will streamline workflows by automating reservations, tracking availability, and generating reports for efficient fleet and rental operations.

Software Architecture Models:

Models used in my project:

- Layered Architecture
- Client–Server Architecture
- Component-Based Architecture (Partial)
- MVC Architecture (UI Level)

Layered Architecture:
Why i used:

- Layered Architecture is used because it helps to organize a system into clear and separate layers.
- It is best for management systems because it keeps the user interface, business rules, and data handling separate. This separation makes the system easier to maintain, debug and update.
- Role in Project:
- Booking rules
- Availability logic
- Payment processing
- Secure database access

Client–Server Architecture:
Why i used:

- Client–Server Architecture is used because it allows multiple users to access the system at the same time using a centralized database.
- When a client requests a booking, the server checks availability, applies rules, stores data securely, and then responds. This architecture is naturally implemented in the system because it supports multi-user access, security, and smooth communication between users and the system.
- **Role in Project:**
- Client requests booking
- Server processes & responds

Component-Based Architecture:

- Component-Based Architecture focuses on building a system using separate components, where each component handles a specific function.
- The main benefit of this approach is better code organization and easier understanding of the system. It also helps in future scalability, as these components can later be converted into fully independent components if needed.
- **COMPONENTS HANDLE:**
- Booking Component
- Vehicle Management Component
- Customer Component
- Payment Component
- Reporting Component

MVC Architecture :

- MVC (Model–View–Controller) Architecture is mainly used to organize the user interface of the system.
- In MVC, the Model represents data such as cars, customers, and bookings. The View shows screens and forms to the user, and the Controller handles user actions like button clicks or form submissions,MVC is not used as the main architecture because it is limited to the UI part only.
- **MVC MAPPING:**
- Model → Data (Cars, Customers, Bookings)
- View → Screens / Forms
- Controller → User actions handling

Why not used:

- **Microservices Architecture**
- Microservices is not used because the project is small to medium in scale. This architecture needs complex deployment and management, which is unnecessary for an academic project.

- **Event-Driven Architecture**
- This architecture is not used because the system does not require real-time event processing. A simple and direct workflow is enough for the rental system.

- **SOA (Service-Oriented Architecture)**
- SOA is not used because there is no integration with external services. The system works as a standalone application.

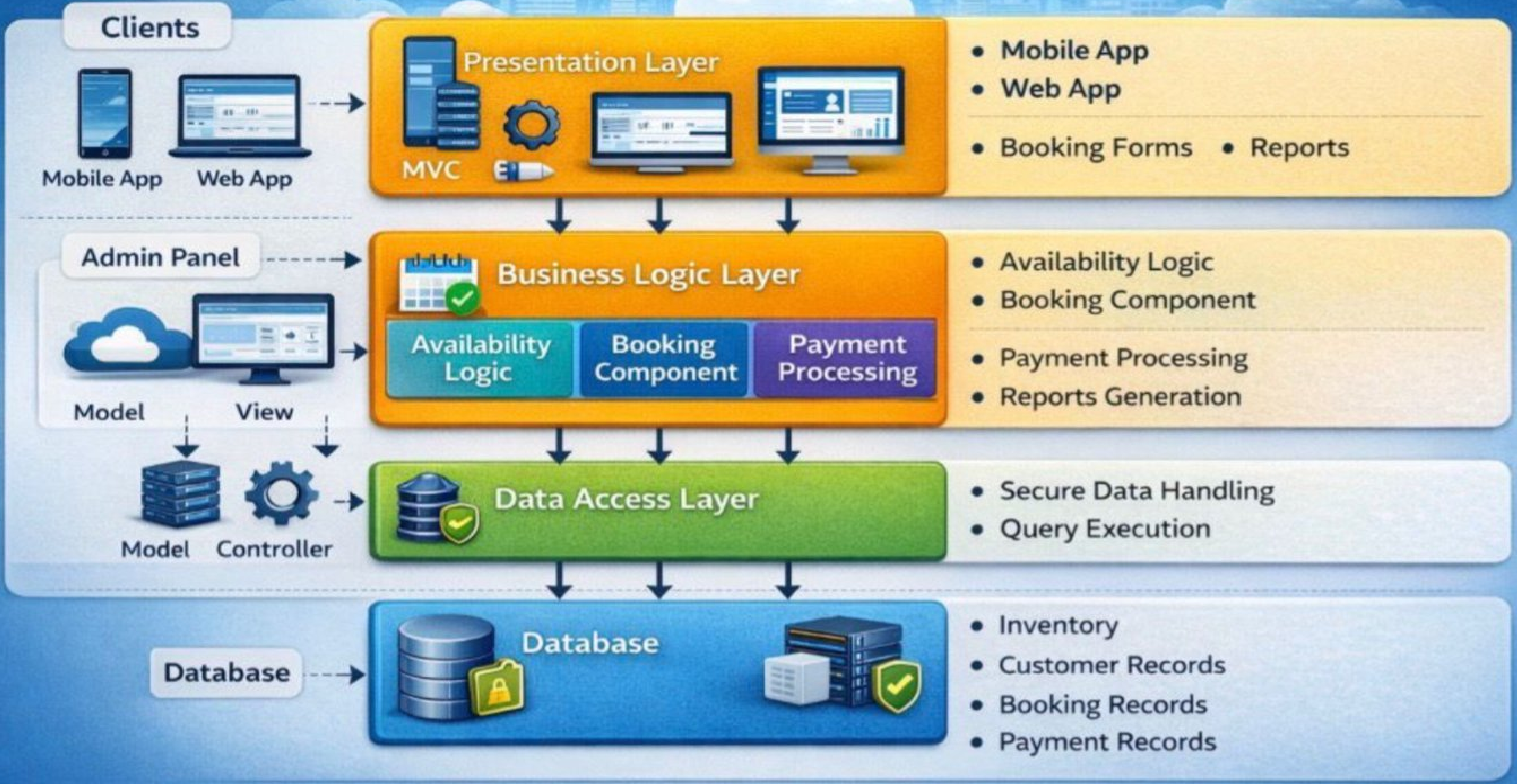
Why not used:

- **Peer-to-Peer Architecture**
- Peer-to-peer is not suitable because the system is centralized. Clients do not communicate with each other; all communication goes through the server.

- **Pipe-and-Filter Architecture**
- This architecture is not used because the system does not handle continuous data streams. The rental system is transactional, not data-flow based.

- **Broker Architecture**
- Broker architecture is unnecessary because the system is not distributed. Direct communication between client and server is sufficient.

Layered Architecture for Rent a Car Management System


Clients
Mobile App
Web App
Presentation Layer
MVC
• Mobile App
• Web App
• Booking Forms • Reports
Admin Panel
Business Logic Layer
Availability Logic
Booking Component
Payment Processing
Model
View
• Availability Logic
• Booking Component
• Payment Processing
• Reports Generation
Model
Controller
Data Access Layer
• Secure Data Handling
• Query Execution
Database
Database
• Inventory
• Customer Records
• Booking Records
• Payment Records