

# Ducati Store Management System

## Team Members:

- Artyom Shpringer

## Topic:

Ducati Store Management System

## Aim:

To develop an integrated, user-friendly management system for Ducati motorcycle sales, inventory tracking, customer management, and service records to streamline operations and enhance customer satisfaction.

## Goals:

1. Build a centralized system for efficient management of motorcycles, inventory, and orders.
2. Provide real-time insights into inventory and sales data.
3. Enhance customer experience through personalized service and efficient tracking.
4. Reduce operational inefficiencies in service and inventory management.

## Analysis of Existing Technologies:

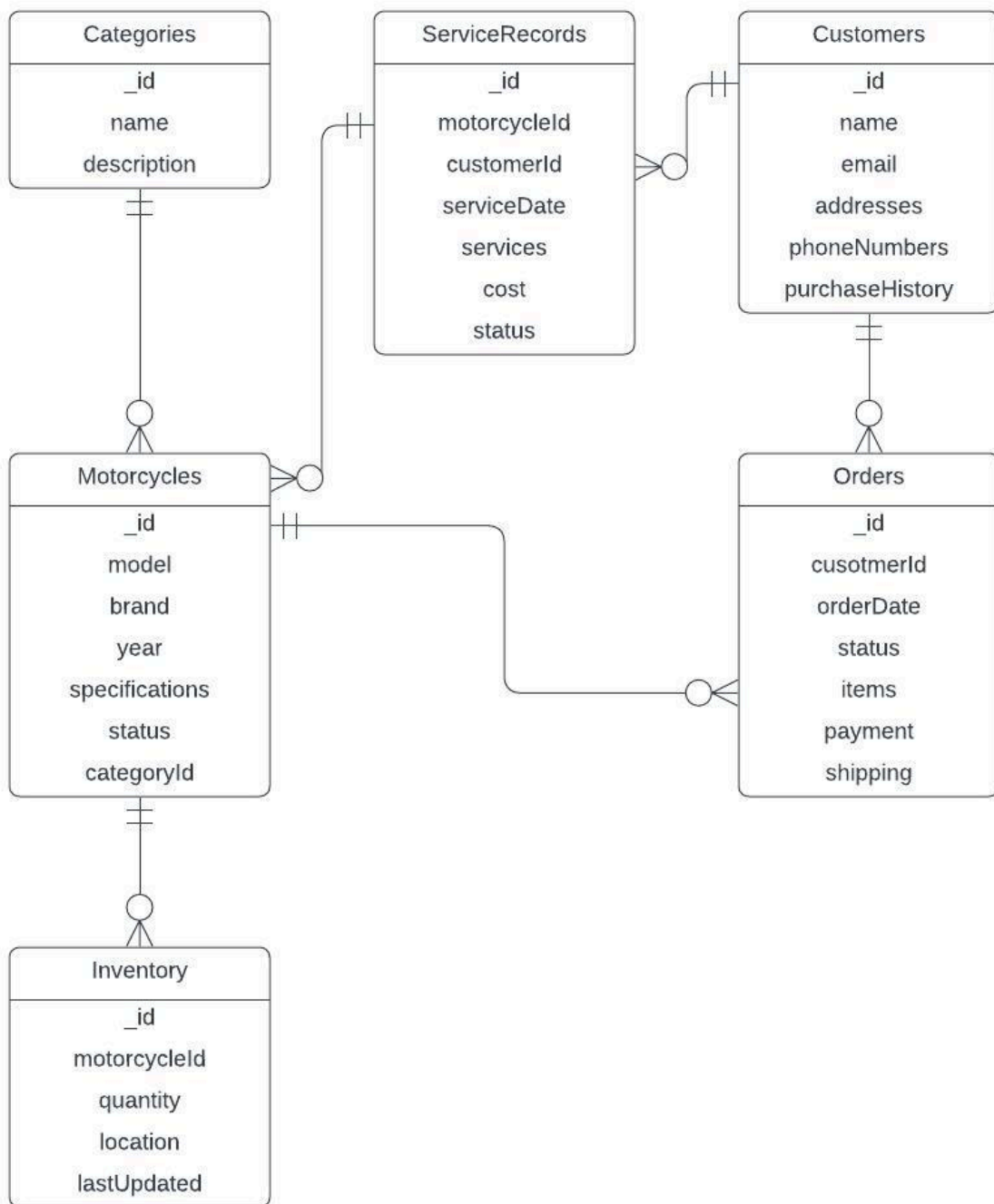
Feature	SAP Business One	Shopify	QuickBooks	Ducati System
Motorcycle Inventory	Yes	Limited	No	Yes
Customer Management	Yes	Yes	Yes	Yes
Service Records	No	No	No	Yes
Cost	High	Moderate	Low	Moderate

Customization	Moderate	Limited	Limited	High

SAP Business One is too expensive for small-scale dealerships, Shopify lacks automotive-specific features, and QuickBooks is great for accounting but lacks inventory tracking. Our Ducati System provides an integrated solution optimized for motorcycle sales and services.

## Architecture of the Project:

- Model-View-Controller (MVC)
- Model: MongoDB database for storing motorcycles, inventory, customer data, and service records.
- View: React-based frontend for a seamless user interface.
- Controller: Golang middleware for handling business logic and API requests.



## NoSQL Features Usage:

- Document-Oriented Structure: JSON-like documents for efficient data retrieval.
- Embedded Documents: Reducing expensive JOIN operations.
- Indexes: Implemented on customer email, motorcycle model, and order status for faster lookups.

- Replication & Sharding: Uses MongoDB replica sets and sharded clusters to handle large datasets.

## Data Handling and Queries:

Example CRUD Operations:

### Insert a New Motorcycle

```
db.motorcycles.insertOne({
  model: "Ducati Panigale V4",
  brand: "Ducati",
  year: 2024,
  price: 25000,
  specifications: ["1103cc engine", "214 hp", "6-speed gearbox"],
  status: "Available",
  categoryId: ObjectId("654c1d0b2f9a1e3d6f8a1f2b")
});
```

### Find All Available Motorcycles

```
db.motorcycles.find({ status: "Available" });
```

### Update Customer Phone Number

```
db.customers.updateOne(
  { email: "customer@example.com" },
  { $set: { phoneNumbers: ["+1 555 123 4567"] } }
);
```

### Delete an Order

```
db.orders.deleteOne({ _id: ObjectId("654c1d0b2f9a1e3d6f8a1f3c") });
```

## Performance & Security Measures:

- Optimized Queries with Indexing.
- Role-Based Access Control (RBAC).
- Data Encryption for customer data.
- Injection Attack Prevention using parameterized queries.

## Conclusion:

The Ducati Store Management System is designed to provide scalable, high-performance, and user-friendly motorcycle inventory and service management. By leveraging NoSQL

features, we ensure efficient data handling, fast queries, and high availability, making it an ideal solution for Ducati dealerships.