

Database Design and Development Report

Date	22-07-2024
Team ID	SWTID1720162063
Project Name	Cab Booking Application
Maximum Marks	

Project Title: [Cab Booking Application]

Date: [22-07-24]

Prepared by: [Jaswanth(Team lead),Mothish(Team Member),Rishi(Team Member),Prasad(Team member).]

Objective

The objective of this report is to outline the database design and implementation details for the Cab Booking Application project, including schema design and database management system (DBMS) integration.

Technologies Used

- **Database Management System (DBMS):** MongoDB
- **Object-Document Mapper (ODM):** Mongoose

Design the Database Schema

The database schema is designed to accommodate the following entities and relationships:

1. Users

- Attributes: _id, name, email, password, createdAt, updatedAt

2. Cabs

- Attributes: _id, cabNumber, model, capacity, available, createdAt, updatedAt

3. Drivers

- Attributes: _id, name, licenseNumber, cab (references Cab), createdAt, updatedAt

4. Bookings

- Attributes: _id, user (references User), cab (references Cab), driver (references Driver), pickupLocation, dropLocation, pickupTime, status, createdAt, updatedAt

Implement the Database using MongoDB

The MongoDB database is implemented with the following collections and structures:

Database Name: cab_booking_app

5. Collection: users

```
{  
  "_id": ObjectId,  
  "name": String,  
  "email": String,  
  "password": String,  
  "createdAt": Date,  
  "updatedAt": Date  
}
```

2. Collection: cabs

```
{  
  "_id": ObjectId,  
  "cabNumber": String,  
  "model": String,  
  "capacity": Number,  
  "available": Boolean,  
  "createdAt": Date,  
  "updatedAt": Date  
}
```

3. Collection: drivers

```
{
```

```
"_id": ObjectId,  
"name": String,  
"licenseNumber": String,  
"cab": ObjectId (references cabs),  
"createdAt": Date,  
"updatedAt": Date  
}
```

6. Collection: bookings

```
{  
  "_id": ObjectId,  
  "user": ObjectId (references users),  
  "cab": ObjectId (references cabs),  
  "driver": ObjectId (references drivers),  
  "pickupLocation": String,  
  "dropLocation": String,  
  "pickupTime": Date,  
  "status": String,  
  "createdAt": Date,  
  "updatedAt": Date  
}
```

Integration with Backend

- **Database connection:** Provide a screenshot of the database connection done using Mongoose.
- The backend APIs interact with MongoDB using Mongoose ODM. Key interactions include:
 - **User Management:** CRUD operations for users.
 - **Cab Management:** CRUD operations for cabs, with availability status updates.

- **Driver Management:** CRUD operations for drivers and their assigned cabs.
- **Booking Management:** CRUD operations for bookings associated with users, cabs, and drivers.