**Tool Rental Agreement - Project Documentation**

***Anuradha Vasudevan – 11th July 2024***

### **Tool Rental Application** - Spring Boot Implementation

#### **Overview**

This project demonstrates a simple tool rental application implemented using Spring Boot. The application serves as a point-of-sale system for a store that rents out various tools such as ladders, chainsaws, and jackhammers. Customers can rent tools for a specified number of days, and upon checkout, a rental agreement is generated that includes the rental charges, applicable discounts, and due dates.

#### **Features**

1. \*\*Tool Rental\*\*: Customers can rent tools for a specified number of days.

2. \*\*Rental Agreement Generation\*\*: Upon checkout, a detailed rental agreement is produced.

3. \*\*Daily Rental Charges\*\*: Each tool type has a different daily rental fee.

4. \*\*Weekend and Holiday Charges\*\*: Some tools have different charges on weekends and holidays.

5. \*\*Discounts\*\*: Clerks can apply discounts to the total rental charges.

#### **Tools**

The application supports the following tools:

- Chainsaw (Stihl)

- Ladder (Werner)

- Jackhammer (DeWalt and Ridgid)

#### **Database**

For ease of testing and development, the application uses an **H2 in-memory database**. This allows the application to run without needing an external database setup, simplifying the testing process.

A screenshot of a computer

Description automatically generated

#### **Configuration**

The H2 database configuration is defined in the `application.properties` file:

```properties

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=password

spring.h2.console.enabled=true

spring.h2.console.path=/h2-console

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.jpa.hibernate.ddl-auto=update

```

#### **Initial Data**

Sample data for tools is initialized using a **command-line runner**:

```java

@Component

public class DataInitializer implements CommandLineRunner {

    @Autowired

    private ToolRepository toolRepository;

    @Override

    public void run(String... args) throws Exception {

        Tool chainsaw = new Tool("CHNS", "Chainsaw", "Stihl", BigDecimal.valueOf(1.49), false, true);

        Tool ladder = new Tool("LADW", "Ladder", "Werner", BigDecimal.valueOf(1.99), true, false);

        Tool jackhammerD = new Tool("JAKD", "Jackhammer", "DeWalt", BigDecimal.valueOf(2.99), false, false);

        Tool jackhammerR = new Tool("JAKR", "Jackhammer", "Ridgid", BigDecimal.valueOf(2.99), false, false);

        toolRepository.save(chainsaw);

        toolRepository.save(ladder);

        toolRepository.save(jackhammerD);

        toolRepository.save(jackhammerR);

    }

}

```

#### **REST API**

The application exposes a REST API for checking out tools. Below is an example curl command for checking out a tool:

```sh

curl -X POST "<http://localhost:8080/api/rentals/checkout?toolCode=LADW&rentalDays=3&discountPercent=10&checkoutDate=2020-07-02>"

Output:

A screenshot of a computer

Description automatically generated

```

#### **Testing**

To validate the functionality, you can use the following curl commands for different test cases:

1. \*\***Test Case 1:**\*\*

   ```sh

   curl -X POST "<http://localhost:8080/api/rentals/checkout?toolCode=JAKR&rentalDays=5&discountPercent=101&checkoutDate=2015-09-03>"

Output:

A screenshot of a computer

Description automatically generated

   ```

2. \*\***Test Case** 2:\*\*

   ```sh

   curl -X POST "<http://localhost:8080/api/rentals/checkout?toolCode=LADW&rentalDays=3&discountPercent=10&checkoutDate=2020-07-02>"

Output:

A screenshot of a computer

Description automatically generated

   ``

`

3. \*\***Test Case** 3:\*\*

   ```sh

   curl -X POST "<http://localhost:8080/api/rentals/checkout?toolCode=CHNS&rentalDays=5&discountPercent=25&checkoutDate=2015-07-02>"

Output:

A screenshot of a computer

Description automatically generated

   ```

4. \*\***Test Case 4:\*\***

   ```sh

   curl -X POST "<http://localhost:8080/api/rentals/checkout?toolCode=JAKD&rentalDays=6&discountPercent=0&checkoutDate=2015-09-03>"

Output:

A screenshot of a computer

Description automatically generated

   ```

5. \*\***Test Case 5**:\*\*

   ```sh

   curl -X POST "<http://localhost:8080/api/rentals/checkout?toolCode=JAKR&rentalDays=9&discountPercent=0&checkoutDate=2015-07-02>"

Output:

A screenshot of a computer

Description automatically generated

   ```

6. \*\*Test Case 6:\*\*

   ```sh

   curl -X POST "<http://localhost:8080/api/rentals/checkout?toolCode=JAKR&rentalDays=4&discountPercent=50&checkoutDate=2020-07-02>"

Output:

A screenshot of a computer

Description automatically generated

   ```

The Same above Outputs can be formatted in a UI, or in any other power BI oriented fotmayts as a Table, as per need.

### **Conclusion**

This project demonstrates a simple yet comprehensive tool rental application using Spring Boot. The use of an H2 in-memory database makes it easy to test and develop without requiring an external database setup.