**Solution Architecture Document for ToolRentalApplication**

**Table of Contents**

1. Introduction
2. System Overview
3. Architecture Design
   * 3.1 Logical View
   * 3.2 Physical View
   * 3.3 Process View
   * 3.4 Development View
4. Technology Stack
5. Data Model
6. Security
7. Deployment

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to describe the architecture of the **ToolRentalApplication**. This includes the high-level design, system components, and how these components interact to fulfill the requirements.

**1.2 Scope**

This document provides an overview of the system architecture, including its logical, physical, process, and development views. It also details the technology stack, data model, security considerations, and deployment strategy.

**2. System Overview**

The **ToolRentalApplication** is a system designed to manage the rental of tools. It provides functionalities for tool checkout, tool and cost management, and maintaining a cache of tool information.

**3. Architecture Design**

**3.1 Logical View**

The logical view outlines the main components of the system and their interactions.

**Components**

1. **ToolRentalApplication**: The main entry point of the application.
2. **InteractiveRunner**: Handles user interaction and commands.
3. **CheckoutService**: Manages the tool checkout process.
4. **ToolService**: Provides tool-related services.
5. **CacheService**: Caches tool and tool cost information.
6. **ToolRepository**: Data access object for tools.
7. **ToolCostRepository**: Data access object for tool costs.

Class Diagram

A diagram of a service

Description automatically generated

**3.2 Physical View**

The physical view describes the deployment of the software components on the hardware nodes.

* **Client**: User interface (Command Line Interface).
* **Application Server**: Runs the Spring Boot application.
* **Database Server**: H2 in-memory database.

**3.3 Process View**

The process view explains the runtime behavior of the system.

1. **Startup**: The application initializes and loads the tool and tool cost data into the cache.
2. **User Interaction**: Users interact with the application through the **InteractiveRunner** class, which processes commands and interacts with the **CheckoutService**.
3. **Checkout Process**: The **CheckoutService** validates tool codes, calculates rental charges, and generates rental agreements.
4. **Caching**: The **CacheService** loads tool and tool cost data into memory to optimize lookup performance.

**3.4 Development View**

The development view describes the software's static organization in its development environment.

* **Packages**:
  + **com.toolrental**: Main application package.
  + **com.toolrental.service**: Contains service classes like **CheckoutService**, **ToolService**, and **CacheService**.
  + **com.toolrental.model**: Contains entity classes like **Tool** and **ToolCost**.
  + **com.toolrental.repository**: Contains repository interfaces **ToolRepository** and **ToolCostRepository**.

**4. Technology Stack**

* **Programming Language**: Java 11
* **Framework**: Spring Boot
* **Database**: H2 (in-memory)
* **Build Tool**: Maven
* **Logging**: SLF4J with Logback

**5. Data Model**

The data model includes two main entities: **Tool** and **ToolCost**.

* **Tool**
  + **code**: String (Primary Key)
  + **toolType**: String
  + **brand**: String
* **ToolCost**
  + **toolType**: String (Primary Key)
  + **dailyCharge**: Double
  + **weekdayCharge**: Boolean
  + **weekendCharge**: Boolean
  + **holidayCharge**: Boolean

**6. Security**

Since this is a simple command-line application, security considerations include:

* **Data Validation**: Ensuring input data is correctly validated.
* **Error Handling**: Proper error handling to avoid exposing sensitive information.
* **Dependencies**: Keeping dependencies up-to-date to avoid known vulnerabilities.

**7. Deployment**

**7.1 Deployment Steps**

1. **Build the application**: **mvn clean package**
2. **Run mvn build** mvn spring-boot:run -Dspring-boot.run.arguments="--runInteractive=true"
3. **Run the application**: **java -jar target/tool-rental-app-1.0-SNAPSHOT.jar**

**7.2 Configuration**

The application uses **application.properties** to configure the H2 database and other settings. Ensure this file is correctly set up before deployment.