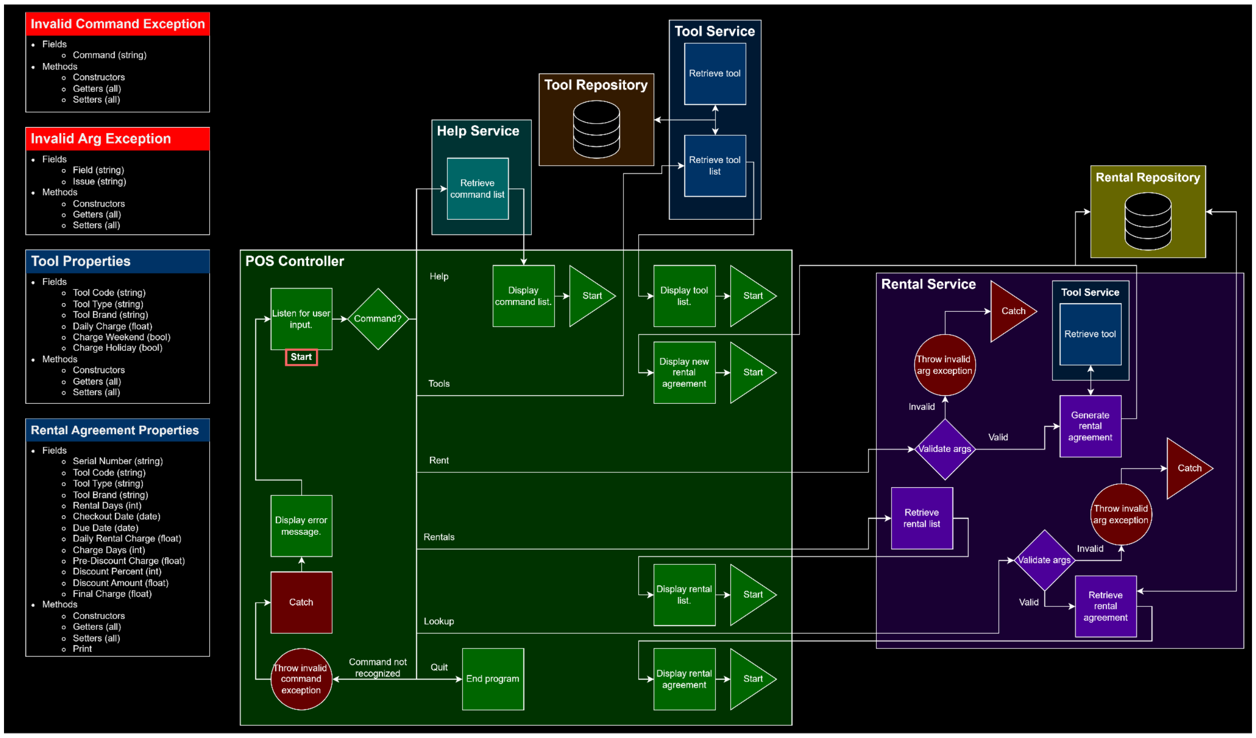
Tool Rental

Table of Contents

Visual Aid



Problem

As an applicant, I want to design and implement a technical solution to the prompt provided by Cardinal Financial.

Specs

- Application
    - Technologies
        - Java
        - Junit
        - No UI required.
        - No database required.
    - Purpose
        - POS app for a hardware store.
        - A customer rents tools for a number of days.
        - Upon rental, the application produces a Rental Agreement.
        - Each day of rental accrues a daily rental fee, varied by tool type.
        - Certain tools are free-of-charge on weekends and holidays.
        - Clerks may apply discounts to customers on certain days of the rental period, reducing the overall due amount.

Questions

- ~~What VO’s will be required?~~
    - Tool
    - Rental agreement
    - Invalid arg exception
    - Invalid command exception
- ~~What classes will be required?~~
    - POS controller
    - Help service
    - Tool service
    - Rental service
- ~~How will Junit be implemented in a private product like this demo?~~
    - JUnit is installed by default in the version of Gradle I am using.

Solution

- Preface
    - Considering the application would work much like the backend API for a POS UI, I will be structuring it as such.
    - Without a database component, I will be using an initialization method to populate what would be the database layer. These will be basic objects with getters and setters like standar VO’s.
- Controller
    - Acts as the entry point for the application. This will contain the IO loop required to interface with the application in the absence of a UI component.

- This controller class will receive user requests, direct the request to the proper service class, catch any exceptions, and output proper responses.

- Help Service
    - The help service is a product of the constraints of the project.
    - In the absence of a UI, the user may need some additional assistance in interfacing with the application properly.
    - The help service will provide the instructions list required for the user.
- Tool Service
    - The tool service will handle retrieving the list of tools and individual tools by tool code.
- Rental Service
    - The rental service will handle requests regarding renting tools.
    - The rental service will handle the generating and looking up Rental Agreements.
    - Invalid requests will throw an exception that will be bubbled up to the controller for handling.
- Tool Repository
    - The tool repository is the stand in for the tool database component.
    - The tool repository will store tool information and be used for retrieving tool details to generate rental agreements and lookup tool details.
- Rental Repository
    - The rental repository is the stand in for the rental agreement database component.
    - The rental repository will store generated rental agreements and be leveraged when looking up rental agreements.