Engineering Home Task

Challenge

In finance, it's common for accounts to have so-called "velocity limits". In this task, you'll create a Java Spring boot application that accepts or declines attempts to load funds into customers' accounts in real-time.

Each attempt to load funds will come as a single-line JSON payload, structured as follows:

{
"id": "1234",
"customer_id": "1234",
"load_amount": "$123.45",
"time": "2018-01-01T00:00:00Z"
}
Each customer is subject to three limits:

A maximum of $5,000 can be loaded per day
A maximum of $20,000 can be loaded per week
A maximum of 3 loads can be performed per day, regardless of amount
As such, a user attempting to load $3,000 twice in one day would be declined on the second attempt, as would a user attempting to load $400 four times in a day.

For each load attempt, you should return a JSON response indicating whether the fund load was accepted based on the user's activity, with the structure:

{ "id": "1234", "customer_id": "1234", "accepted": true }
You can assume that the input arrives in ascending chronological order and that if a load ID is observed more than once for a particular user, all but the first instance can be ignored (i.e. no response given). Each day is considered to end at midnight UTC, and weeks start on Monday (i.e. one second after 23:59:59 on Sunday).

Your program should process lines from input.txt and return output in the format specified above, either to standard output or a file. The expected output given our input data can be found in output.txt.

Some things to consider:

- Your program should make use of a relational database (preferably in-memory)
- Follow best practices for error handling and logging
- Implement tests
- Generally we are looking for your ability to design and build production-ready products - so think about simple good design that is maintainable and easy to troubleshoot