***Develop a Java application utilizing Spring Boot and a streaming tool, incorporating both SQL and NoSQL databases. Ensure all functionalities are covered with JUnit tests, adopting a Test-Driven Development (TDD) approach. Set up a deployment pipeline, preferably through GitHub Actions, or alternatively, Jenkins. This pipeline should include automated regression testing and an Open Source Software vulnerability scan, with the build failing upon detection of critical or blocker vulnerabilities. If feasible, please use PlantUML to create necessary sequence, class, or design diagrams.***

**Problem Statement**

Imagine a situation where **thousands** of trades are being sent to a single store, through any method of **transmission**. Our objective is to develop a trade store that organizes and stores each trade in a specific sequence.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Trade Id | Version | Counter-Party Id | Book-Id | Maturity Date | Created Date | Expired |
| T1 | 1 | CP-1 | B1 | 20/05/2020 | <today date> | N |
| T2 | 2 | CP-2 | B1 | 20/05/2021 | <today date> | N |
| T2 | 1 | CP-1 | B1 | 20/05/2021 | 14/03/2015 | N |
| T3 | 3 | CP-3 | B2 | 20/05/2014 | <today date> | Y |

The assignment requires several validations to be implemented:

1. If a trade with a lower version is received during transmission, the store will reject it and generate an exception. Trades with the same version will replace the current record.
2. The store must reject any trade that has a maturity date earlier than today's date.
3. When a trade's maturity date is surpassed, the store should automatically mark the trade as expired.