

SuperDataWorker

Lavender is the company working on real estate. They have lists of customers, apartments and the rental contracts. However, their employees create and manage different files so it is hard to report. The director wants to have a tool to migrate all data files into the database so he could use the report application to analyze and make decisions better.

Duration: 5 days

Team size: 1 people

Required Technologies

You must use below technologies:

- Java 8
- Hibernate
- Maven 3
- Log4j 4
- Apache Common
- Mysql Database
- H2 Database

Functional Requirements

The main requirement is to load data files into the database.

Each employee has three different files to store customers, apartments and contracts.

The application could import multiple files at the same time.

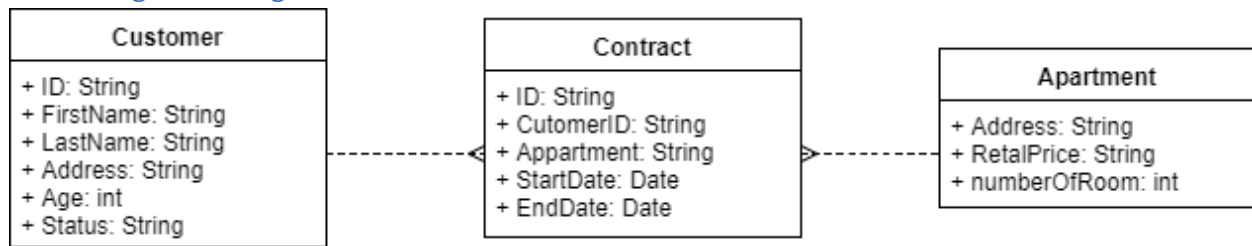
The application is a runnable jar file with configured parameters on the command line. For instance, **java -jar SuperDataWorker DATA_TYPE file1.csv file2.csv**

- DATA_TYPE is the parameter to define data to import. It could be customers, apartments or contracts
- file1.csv, file2.csv, etc: the data files listed out on the command line

The application must detect and avoid duplication data (rows) in different files.

The application must detect missing data in different files. For instance, the importing contracts do not have information of customers or apartments. In this case, the failure contracts must be store in a file after the loading process.

Class Diagram Design



Requirements

- Create model classes **Customer**, **Apartment** and **Contract** as the class diagram.
- Create service classes to read data files and convert to different model objects. Let's say that there are three methods: loadCustomers, loadApartments and loadContracts in three different classes: CustomerService, ApartmentService and ContractService.
- Create three DAO classes to CR (do not need to UD) models to/from the database. These DAO classes must inherit from an abstract DAO class, named ModelDAO using the generic class design. Therefore, we do not need to implement details inside three sub-DAO classes.
- The SuperDataWorker class has the main method to start the process.

Testing Requirements

You must have test cases to:

- Load non-existed files (wrong filenames)
- Load existed files (Test with students.csv)
- Load existed files with missed values (Test with students-missing.csv)
- Save a null model list
- Save an empty model list
- Save a non-empty model list
- Get all models from the database
- Save duplicated models
- Save contracts with missing customers or apartments

Your code must pass all test cases (listed on the attached file).

Non-Functional Requirements

- Use log4j to log all necessary information (info and error in exception cases)
- Error messages must define and use as final variables (see the message file)
- Use properties files to configure the databases and log4j
- Use Apache Common Configuration to load properties files
- In testing code, you must use H2 database

Coding Quality Requirements

You must follow and guarantee below coding quality requirements:

- Code coverage must be more than 75% (both main and test code)

- You must not violate more than 5 times based on the FA Checkstyle