Internet-Based Live Courier Tracking And Delivery System

Course: Software Systems Analysis and Design

Submitted by

Abu Huraira (a.huraira@innopolis.university)

Mostafa Khaled Mostafa Mohamed Aref Kira (m.kira@innopolis.university)

Daria Verevkina (d.verevkina@innopolis.university)

Vladimir Rublev (v.rublev@innopolis.university)

Internet-Based Live Courier Tracking And Delivery System

According to Alexander Shvets, "**Chain of Responsibility** is a behavioral design pattern that lets you pass requests along a chain of handlers. Upon receiving a request, each handler decides either to process the request or to pass it to the next handler in the chain." [1]

The main design pattern of our project was implemented is the **Chain of Responsibility.** We used a **Chain of Responsibility** pattern as our implementation uses a tracking system where a Client can make a request and the request goes through a series of Handlers of requests and ultimately executes the Client's request. Chain of responsibility allows to go through all points of track and see what is the current position of the box. By applying such a pattern we can check every step to determine whether the package was lost or has reached the client.

System Details

Users are split into 5 types: Customer, Distribution Center Admin, Airport Admin, Road Transport Admin, Pickup Point Admin. The possible locations of a given Package are implemented as an Enumeration (DISTRIBUTION_CENTER, AIRPLANE, TRUCK, PICKUP_POINT). Each user is able to see the location of the package upon successful execution of authentication.

Code Elements

The class *Package* consists of the information about a package. Also, the class *User* includes the email of the user, the user's password, and a type of user (which is implemented using an Enumeration). We have an abstract class *Handler* with three extended handlers: **ValidateUserExistsHandler** to check, **ValidatePasswordHandler** to check the correctness of the password. The third is **RoleCheckHandler** to determine the user's role in the system. After handling all the segments of the request the system finally executes the request.

As per the project description, the project is further extendible introducing a barcode scanning system, where the client may make a request just by scanning the barcode. After implementing a handler for retrieving the information from the barcode, the data then would be passed on to the existing handlers to deal with the request.

UML diagram

UML diagram for our system shows all classes, abstract classes, interfaces, and relations within our program, which was implemented in the Java programming language.

The diagram can be accessed in the **UML.svg** in the submission file. A link to a public repository on GitHub has also been provided.

Repository

- https://github.com/hur41r4/ssad-team42

References:

1. Shvets, A., 2020. Dive Into Design Patterns. 1st ed. p.256.