Term Project Report

Chukwubuikem Ume-Ugwa

CIS 566

Route Mining

In order to help the marketing department better determine the return on the investment, I have developed a route mining system that is capable of retrieving, computing basic statistic and export carrier route information for given addresses.

The project is implemented as a web application with a simple and intuitive user interface. The project lets the user input the addresses via two options. The first option is by uploading a well-structured excel file which an example is available on the home page of the web application. The second option is to manually enter the addresses using a web form.

Once the data is submitted, the processing begins and when completed, the user is redirected to the results page.

The project was implemented using the Python programming language and the Flask framework. To solve the problem, the following design patterns where incorporated.

Builder Pattern:

The builder pattern was used due fact that the business model class which the address, is a complex class and the creation of the objects of the class warrants abstracting it away and the builder pattern fits the bill perfectly.

Abstract Factory Pattern:

This pattern was utilized out of necessity due to fact that during development I have freedom to write to the filesystem, however that’s not the case when it is deploy on Google cloud’s App Engine. As result, I needed a way to create a FileIO object based on the environment the application is running in. This helps abstract away the environment issue.

Chain of Responsibility Pattern:

This is pattern was utilized in order to add extensibility to the system. This will allow the system to be able to handle new file format by just implementing a suitable FileHandler and adding it to the chain.

Intercepting Filter Pattern:

This pattern was used to handle post-processing of the data. Essentially a pipeline is created that processes the address once the file handlers are finished with pre-processing the file. This pattern allows for adding more post-processing steps in the future. The current post-processing steps include address validation, carrier route retrieval and storing the processed data for easy report generation.

All these pattern where put together to solve the route mining problem. Below in Figure 1 is the class diagram

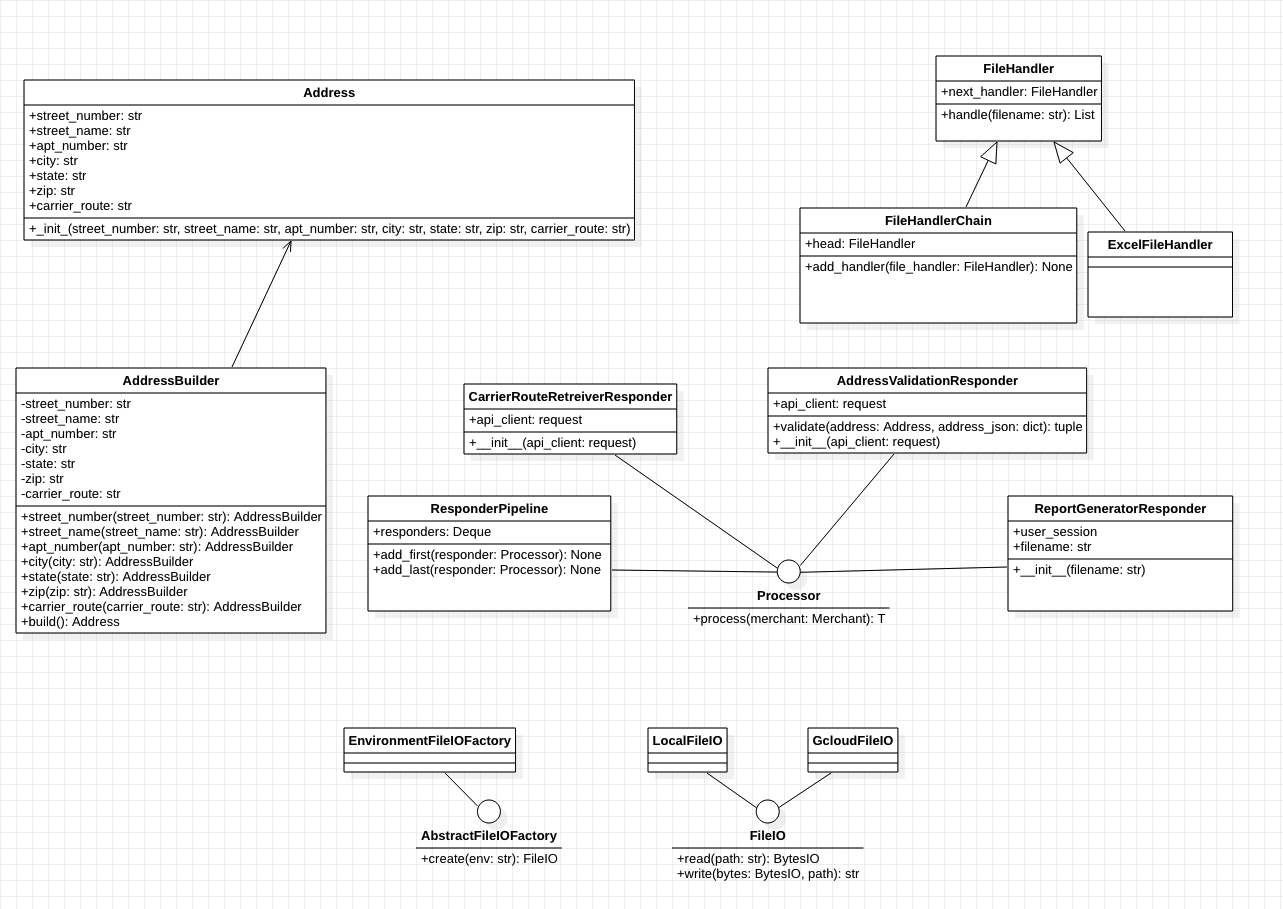


Figure 1: Route Mining Class Diagram