CMPE 202 Software Systems Engineering

Individual Project

Name: Rishikesh Andhare

SJSU ID: 016726203

Deliverables:

GitHub - https://github.com/gopinathsjsu/individual-project-RishikeshKeshav

Describe what is the primary problem you try to solve.

Answer :-My main aim was to develop a solution that allows for the verification of the authenticity of widely-used credit cards, specifically those issued by MasterCard, Visa, American Express, and Discover. The ability to perform this validation is crucial for preventing fraudulent activities and ensuring the accuracy and security of transactions.

Describe what are the secondary problems you try to solve.(If any)

Answer :- Another issue I aimed to address was identifying the suitable design patterns that can accommodate the addition of new credit card types in the future. By implementing such patterns, the system can easily adapt to changes and additions in the credit card landscape without requiring significant modifications to the existing codebase.

Describe what design pattern(s) you use how

- Following are the Design Patterns that have been used to formulate the solution:

1. **Iterator Pattern:**
   - For a method of sequentially accessing a collection object's elements, use the iterator pattern.
   - Each kind of file has several records, hence this pattern aids in processing each record by iterating through each record.

2. **Strategy Pattern:**
   - Design patterns for strategies enable applications to change their behavior in accordance with the chosen strategies.

· Due to the variety of file types, specific objects are created for each type and appropriate methods are employed depending on the input file.
· To support various file formats, I employed a strategy design pattern.
· CsvFileparser, Jsonfileparser, and Xmlfileparser are the three interfaces I've developed. A file parser is added to all of these. As a result of the input, the behavior modifies

3. Factory Pattern:

· Each file type has multiple records; hence this pattern helps with processing each record by processing each record twice.
· Since there are many kinds of cards, the cardFactory interface is implemented in order to return a newly created object of the card type.

Describe the consequences of using this/these patterns:

1. Iterator pattern:

· Advantages: When working with collections, enables you to work at a higher level of abstraction.
· Disadvantages**:** Iterator is too much to handle for small applications, Iterator is too extravagant.

2. Strategy Pattern:

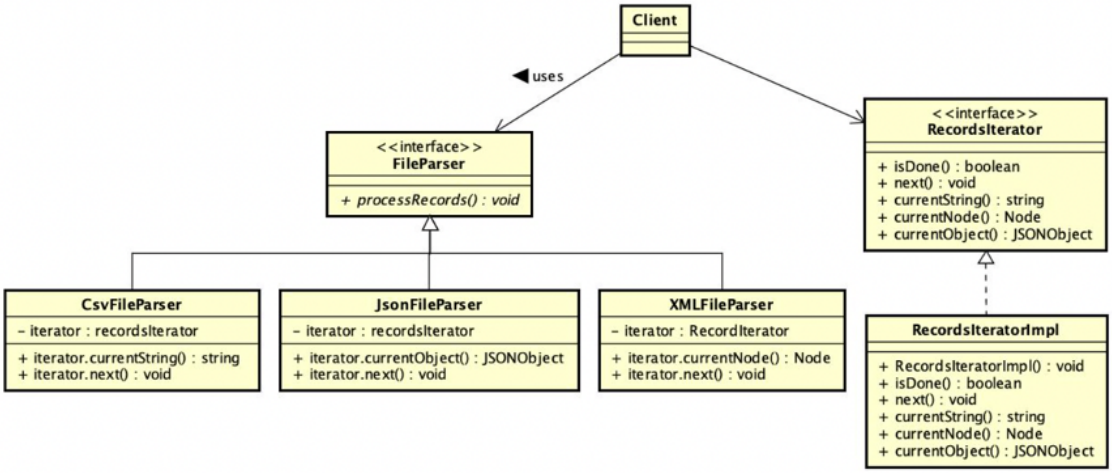
· Advantages**:** It is simple to add as many strategies as necessary when designing a strategy.
· Disadvantages**:** Users should know multiple strategies and the difference between each of them

3. Factory Pattern:

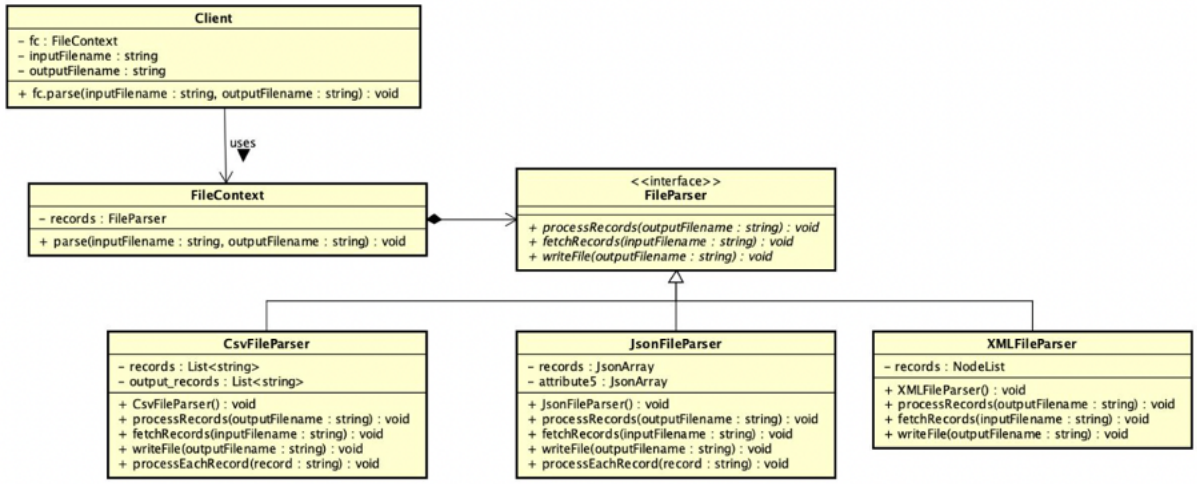
· Advantages**:** The application's ability to be expanded modularly and this is much easier to test it.
· Disadvantages**:** There would a lot of classes which are necessary**.** The application's extension is extremely complex.

Design Class Diagrams:

Iterator Pattern:



Strategy Pattern:



Factory Pattern: