

CMPE 202 Software Systems Engineering

Individual Project

Name: Charles Fathima. B
SJSU ID: 016773055

Deliverables:

Describe what is the primary problem you try to solve:

- The main challenge addressed in this project was validating the authenticity of various available credit cards, including MasterCard, Visa, American Express, and Discover.

Describe what are the secondary problems you try to solve:

- The auxiliary challenge tackled was identifying suitable design patterns that would facilitate the integration of additional credit card types in the future.

Describe what design pattern(s) you use how:

- The solution incorporated the following design patterns:
 1. Iterator Pattern:
 - This pattern is utilized for sequentially accessing elements within a collection object.
 - It is particularly useful in this context because each file type comprises multiple records, and the pattern facilitates processing each record individually.
 2. Strategy Pattern:
 - Strategy design patterns allow for dynamic behavioral changes in the application based on the selected strategy.
 - Given the diversity of file types, specific objects for each file type are created, with the methods employed being contingent on the type of the input file.
 - The strategy design pattern was employed to accommodate different file formats, leading to the creation of three interfaces: CsvFileparser, Json

File Parser, and Xmlfileparser. The behavior of the file parser changes depending on the input.

3. **Factory Pattern:**

- This pattern aids in handling multiple records in each file type by processing each record individually.
- Due to the variety of card types, the cardFactory interface is implemented to generate a new object for each card type.

Describe the consequences of using this/these patterns:

1. **Iterator Pattern:**

Pros: Enhances the level of abstraction when managing collections.

Cons: May be overly complex for smaller applications.

2. **Strategy Pattern:**

Pros: Facilitates the addition of numerous strategies within the design.

Cons: Requires users to be familiar with various strategies and their distinctions.

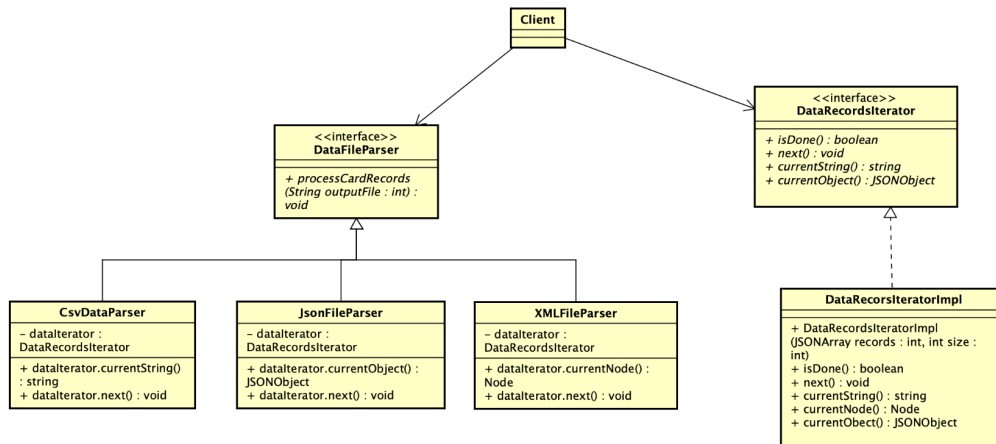
3. **Factory Pattern:**

Pros: Enhances the modularity and testability of the application.

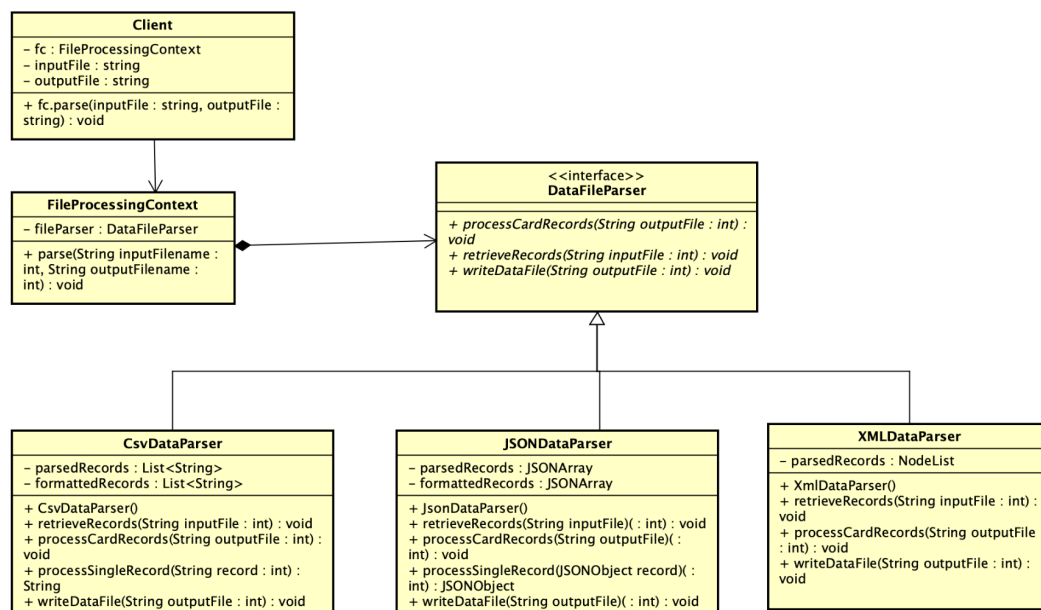
Cons: Necessitates a significant number of classes and can make extending the application quite intricate.

Design Class Diagrams:

Iterator Pattern:



Strategy Pattern:



Factory Pattern:

