CMPE 202 - Individual Project

1. **Primary Problem**

   To read the credit card numbers, validate them if they belong to a valid account. If valid, create an instance of appropriate credit card class. Here, currently we are designing for Visa, MasterCard, Amex, and Discover cards, but we need to design in order to accommodate new cards if added in future without breaking the existing code.

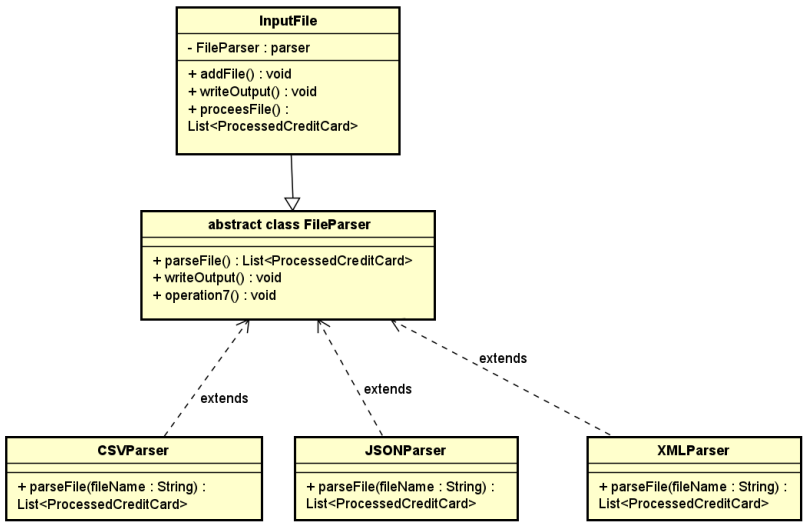
2. **Secondary Problem**

   We need to accept the credit card numbers, so the input format can be anything. We have to make sure the system works for all kinds of files. Currently we are accepting the inputs as CSV file, JSON file, XML file. Thus, in future if we have to accept a new type file, our system should work with minimal addition of code and not breaking the existing code. Thus we can use the strategy pattern to parse the Input file, create the new instance of concrete classes (CSV file, JSON file, XML file) based on the input file given.

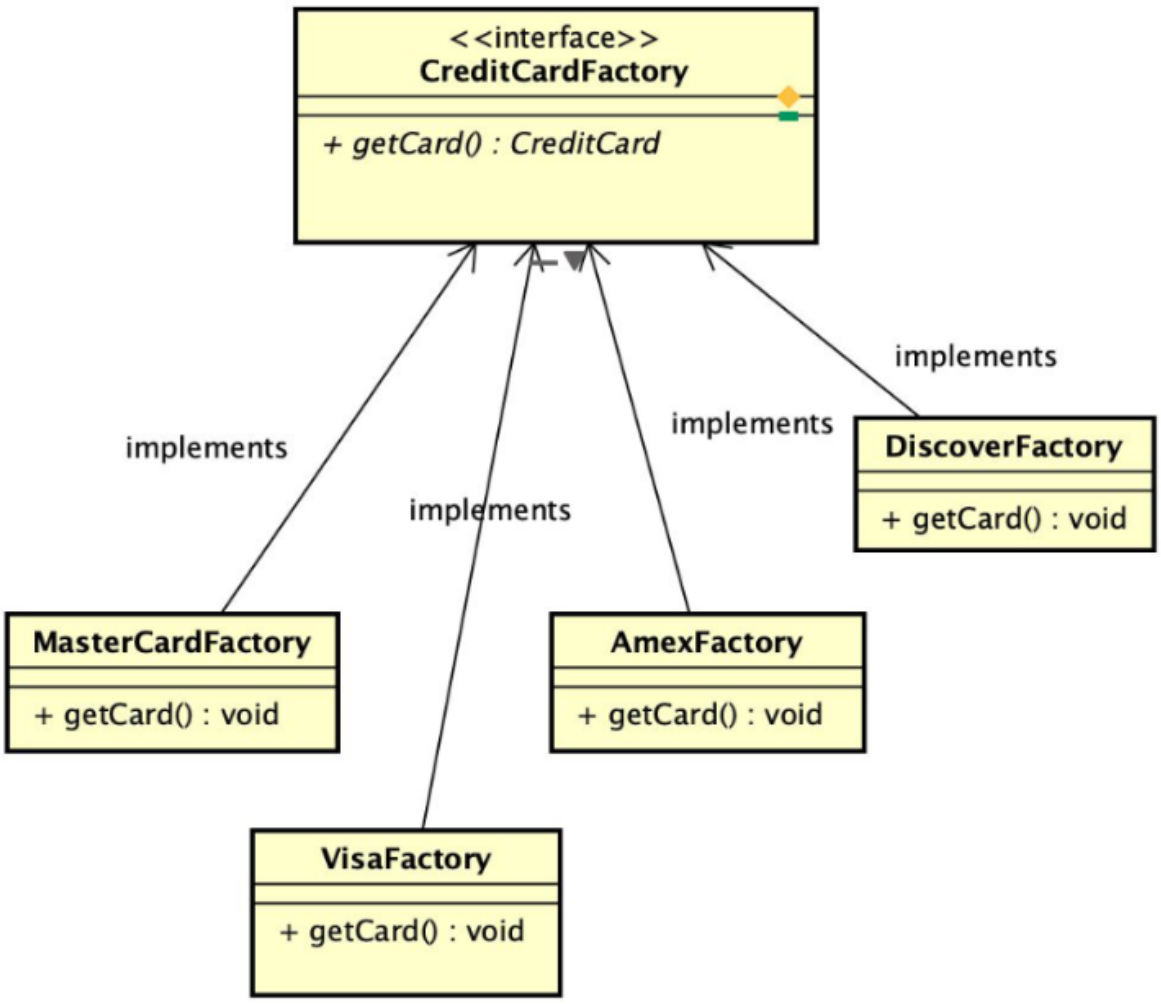
3. **Design Patterns**

   Design patterns used:
   1. Strategy pattern
   2. Factory Design pattern

   Strategy pattern is used to implement multiple file parsers CSV, JSON, XML , the common operation for all these is to parse file with different behavior.



Factory design pattern is used to get different cards from the factory of cards. CreditCardFactory is an interface, all other concrete classes (VIisa, Master, Amex, Discover) implements the interface.



4. **Consequences of Design Patterns:**

Strategy Pattern Consequence : It provides a substitute to subclassing, here defines each File parser behavior within its own class, eliminating the need for conditional statements. It makes it easier to extend and incorporate new behavior without changing the application.

Factory Pattern Consequence: It promotes the loose coupling by eliminating the need to bind application specific classes into code. Code interacts solely with the resultant interface or abstract class, so that it will work with any classes that implement that interface or that extends that abstract class.