**Individual Project**

CMPE-202 Software System Design

Supriya Meduri (015262767)

**PART - 01**

1. Describe what is the primary problem you try to solve

In the given requirement, I recognized the following as the primary problem

* To check whether a card is valid or not by checking with different types of credit cards available
  + Master Card
  + Visa
  + American Express
  + Discover
* To determine the card issuer based on the credit card number.
* To find the relevant objects based on card type

1. Describe what are the secondary problems you try to solve (if there are any).

The secondary problem is to identify the proper design patterns to accommodate

New credit classes that can be added in the future for different credit card types.

**PART-02**

1. What design patterns could you use for the problem statement

**Design Patterns:** Chain of Responsibility, Strategy

**Strategy Design Pattern:**

1. Strategy Design pattern allows to change the behaviour of application based on selected strategy
2. Since there are different file types, objects for each file type are created and appropriate strategies are used based on the input file
3. I used a strategy design pattern to support different file formats. I created three different interfaces - Reader, Writer, CreditCardHandler.
4. Reader Interface - has a readFile method, to read input file based on the type of input file
5. Credit Card Handler Interface - has a checkCreditType method which takes the parsed input file of a type and checks it falls under any credit card type category
6. Writer Interface - has a writeToFile method which takes the input of type output entry(which has the final output format defined) and writes the data to the expected file format.
7. This design pattern helps in swapping the behaviour at runtime. Since each strategy follows open/close principle we can introduce new strategies without affecting client
8. Hence this design patterns helps to accommodate newer file formats without disturbing the existing code/design

**Chain of Responsibility Design Pattern:**

1. After parsing the file (either of xml,json,csv) we need to check the type of creditcard of the four available credit card types
2. The file should be validated to find the credit card type out of the available four credit card types
3. So here we can use chain of responsibility design pattern where the file is validated starting with MasterCard handler
4. If the file does not belong to Master Card handler, the file is passed on to next category
5. To moderate the whole process, we include credit card handler, which passes the file to check with all the other handlers
6. All the different credit card handler implement the main handler and pass the file accordingly

4.Describe the consequences of using this/these pattern(s).

**Chain of Responsibility**

Pros:

1. Separates the sender of the request from the recipients.
2. With this design pattern, the client doesn't need to worry about the chain structure and maintains direct references to its methods.
3. At any point we can change the order of the chain or remove/add.

Cons:

1. Hard to observe the run-time characteristics and debug.

**Strategy Design Pattern:**

Pros:

1. Strategy design has the ease of adding any number of strategies as and when required.

Cons:

1. User should know multiple strategies and the difference between each of them
2. At a time there is a possibility of having more objects and hence can be cumbersome and redundant.

Class Diagram:

