# Mini Project Case Study – Third Party Recharge Application

**Development Environment**

JDK, Notepad or Eclipse

**Type of application (Web /Standalone)**

Standalone

**Efforts required**

2 person-days

**Skills pre-requisite**

Core Java and OOPs features

# Topic: Third Party Recharge Application

1. Requirements 2

1.1. Business 2

1.2. Functional 2

2. Purpose 2

2.1. Project Objectives 2

3. CODING 3

3.1. Brief of coding standard 3

3.2. Error Handling 3

**1.** Requirements

* 1. Business:-

This project aims at creating a third party recharge application. This application allows users to recharge any service providers online. Using this application user can recharge their mobiles with the denominations available for the respective service provider.

## 1.2 Functional:-

Following are the functional needs of the system. More functionality can be added to enhance the application.

1. Accept inputs like mobile number and service provider.
2. Choose from the respective denominations shown and enter the amount appropriately.
3. Service provider specific message should appear if recharge is successful or exception message should be displayed.
4. The user should also be able to check his/her balance irrespective of the service provider.

# 2. Purpose

The purpose of the project is to implement Abstraction and Polymorphism in real time.

Also, the pattern should be programmed in such a way that tomorrow if a new Service provider come(Jio), there’ll not be any changes in the business logic code(**TestImplementation** class).

## 2.1 Project Objectives:-

Methods required for the recharge application.

* + - Interface PayTM
      * void recharge(long mob,int amt)
      * void denominations()
    - All service providers implementing the interface
      * balEnquiry()
* Exception class
  + InvalidAmountException

**Classes Involved: -**

|  |  |
| --- | --- |
| **Class/Interface Name** | **Description** |
| PayTM | Interface that creates a method for recharge that should be defined in all the service provider classes. |
| Airtel,Vodafone,Idea,Jio | Implementation classes |
| TestImplementation | Static method that contains only one call to the recharge method.(This method will have no changes involved even if a new service provider comes up)  Hint: This method will have a parameter of type PayTM. |
| InvalidAmountException | Custom exception classes for exception handling. |
| TestPayTM | Class containing the main method with cases created for every service provider. This class will also call the method in TestImplementation class based on the case selected. |

# 3. Coding

## 3.1 Brief of coding standards:-

The JavaCoding standards as suggested by Sun Microsystems/Oracle will be used while coding.

Some of the important coding standards that needs to be followed:-

1. Package name should contain lowercase letters.
2. Class/Interface names should start with capital letters and every word of the name should start with capital letters.
3. Minimize use of on-demand (.\*) imports.

## 3.2 Error Handling:-

Appropriate user defined error handling should be done in the project. All exceptions are to be caught and appropriate error messages should be displayed. Use only user-defined checked exceptions.

1. If the recharge amount is greater than 500 or less than 10, an exception should be thrown – InvalidAmountException
2. One more innovative exception case of your choice.

4. Questions (To be answered after successfully running the project)

* 1. Where in your project do you see abstraction and polymorphism feature.
  2. Explain where is inheritance and encapsulation in your project.

Hint: You don’t have to write separate code for this. Explore for these features which comes in automatically.