IDENTIFICATION AND IMPLEMENTATION OF REQUEST PATTERN IN INTERNET BANKING APPLICATION

[1] B.Chithra, [2] A.Meiappane, [3] Dr. V. Prasanna Venkatesan

[1] PG Student, Department of Computer Science & Engineering, Sri Manakula Vinayagar Engineering College, Puducherry – 605 107.

[2] Research Scholar, [3] Associate Professor, Pondicherry University, Puducherry

[1] chithramec@gmail.com, [2] auromei@yahoo.com

Abstract— Pattern concept deals with providing the solution for the commonly occurring problems. By implementing pattern in any system we can achieve better reusability and also we can provide the common solution for the repeatedly occurring problem. This is the main use of the pattern. The pattern comes under the concept of the POSA (Pattern Oriented Software Architecture),which it deals with the patterns which we can achieve reusability. For all the business organizations and all its applications we use the pattern which will be very useful. Here we took the internet banking application. As banking as lot of responsibilities we mainly concentrate on the internet banking application. In this paper we are going to concentrate on the request pattern in the internet banking application. We discuss about the patterns , design patterns and we mainly concentrate on the request pattern. If we implement request pattern in the internet banking application for any type of request we can call the request pattern instead of developing it from the scratch which mainly achieves reusability. This is the usage of the pattern. We discuss about the request pattern in detail in this paper .

Keywords: POSA , Design Patterns , Request Pattern Chain Of Responsibility

I. INTRODUCTION

In this paper we are going to identify and implement the request pattern in the internet banking application. Before we discuss about the request pattern we generally discus about the Software architecture and the design patterns.

Software Architecture: This is the general definition of the software architecture. "The software architecture deals with the particular system, the elements involved in the system and also it deals about the relationships between the systems [1]". It tells about the properties and responsibilities of each and every system. Considering the several quality attributes and the requirements of the stakeholders the software architecture of the system is built. This is about the software architecture.

Patterns: The pattern is mainly used in the business organizations in order to achieve the reusability. It improves the modularity of the particular system. In order to attain all these quality attributes in the system we go for the pattern concept**.** The general definition of the pattern is that "Giving solution to the recurring problem at particular context [2]". Pattern mainly deals with the three important concepts as follows:

Context: Situation that gives rise to the problem.

Problem: Problems occurring in that situation.

Solution: A proven solution to the given problem.

Types of patterns:

Architectural Pattern: It tells about the fundamental structure of the application**.**

Design patterns: It is the blue print of the particular solution.

Idioms: Describes how to implement. Addresses both design and implementation.

II. DESIGN PATTERN

In this paper we are mainly going to concentrate on design pattern .From that we are going to concentrate only on the chain of responsibility pattern which the request pattern is implemented using the concept of the chain of responsibility pattern.

TYPES OF DESIGN PATTERNS:

Creational Pattern : Creational patterns are used to create the objects for our application whenever there is a need.

Structural Pattern: Structural pattern tells about the relationship between the objects**.**

Behavioral Patterns : Behavioral patterns tells about the communication between the objects.

In this paper we are mainly going to concentrate on the Request pattern which is identified using the chain of

responsibility where it belongs to the behavioral pattern. We will discuss about this chain of responsibility in detail in this paper.

III.Request Pattern:

By using patterns the developers feel very easy to develop a software. They can develop the system by reusing the coding instead of starting from the scratch.

In this paper we are going to concentrate on the request pattern where it avoids the issues and we can do any type of request. By using this request pattern the developer need not want to develop a system from scratch. If he wants to do any type of request in the banking we can call request pattern.

This request pattern uses the chain of responsibility design pattern. This proposed request pattern is executed at the run time .During the run time the customers can select the type of the request he want to do . By using this method we can achieve the enhanced reusability.

There are lot of request we can do in the internet banking application ie : For eg We can request for loan and insurance type of request. If we write the coding for loan we can call that as a request pattern then we can reuse the same code for the insurance request also. Not only for insurance if any other kind of request we need to perform we can call this request pattern.

Structure Of the Request Pattern

The request pattern does not exists so far. In order to make request we should code for each and every time. To avoid this difficulty the pattern gives the best solution. So we are going to implement the request pattern so that we can call the request pattern for all the kind of requests in the internet banking application. We can use the request pattern whenever there is a need.

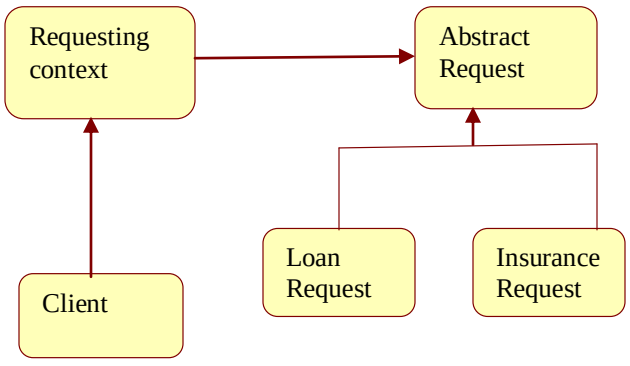


Fig 1 : Structure Of the request Pattern

This is the structure Of the request pattern. The client will directly call the requesting pattern where the abstract request will have all type of request .and when customer do any type of request then they can call from the abstract context as shown in the above figure.

This request pattern uses the concept of the chain of responsibility pattern.

Chain Of Responsibility:

The chain of responsibility design pattern consist of the command objects and the series of processing objects. The series of request is passed through the processing object. If the first handler could not able to process the request the request is passed to the next handler and so on till the request gets completed.

Each and every request of the internet banking application is going to be of this type and hence we create the request pattern and call it for all type of request. This is the use of the request pattern which it is implemented by chain of responsibility design pattern .

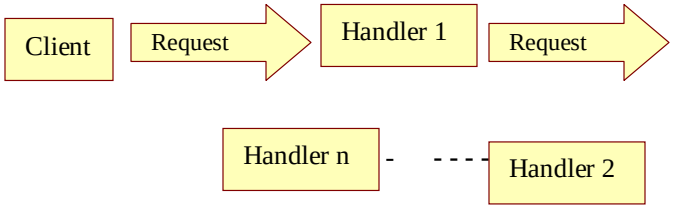


Fig2: Request handling by different handlers.

Structure Of COR Pattern

The structure of COR pattern is described as below.

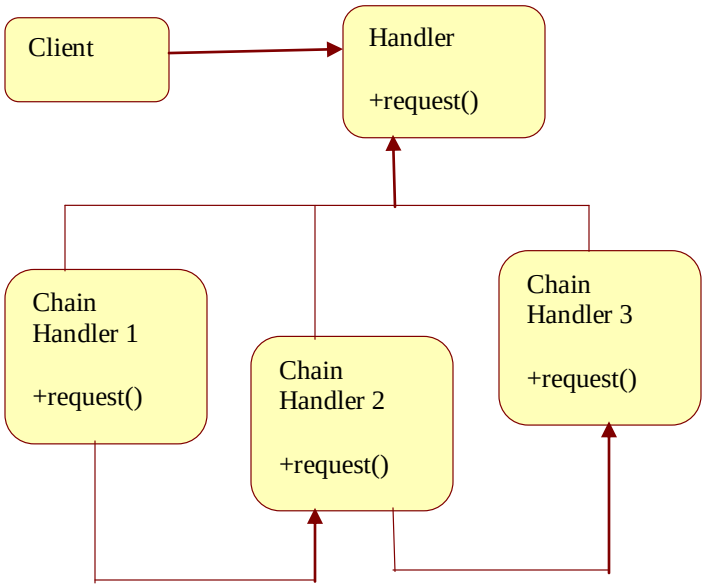


Fig 3: Structure Of the COR Pattern

The client provides the request to the system , the handler first handles the request if he could not finish the process the request is handled to the chain handler 1 if he could not then the request is passed to the next handler and so on.

Participants:
Handler:
It defines the interface for request handling. In order to call many request we can call that interface.

RequestHandler:
It will handle the request if not possible then the request is passed to the next request handler and so on,likewise the request is passed to the successor until the request gets completed.

Client:
Client first sends the request to the first handler and then the request is passed to the successors if it cannot be handled.

Uses:

If the person do any banking request , they can call the request pattern . The type of the request the users want to perform is decided and selected at the runtime.
The various requests are override at the runtime. The user should give the type of the request he wants to perform.
In case of any type of request the input should given. And the request pattern is called

IV. Implementation Of Request Pattern

Intent : It Avoid coupling the sender of a request to its receiver by giving more than one object a chance to handle the request. Chain of responsibility is used when we have series of processing which will be handled by a series of handler logic. Here we are going to discuss about loan and insurance processing requests.
Problem : How to process or pass request in loan or insurance giving process
Solution : In the internet banking application The Requests for loan approving is process and insurance process are implemented.

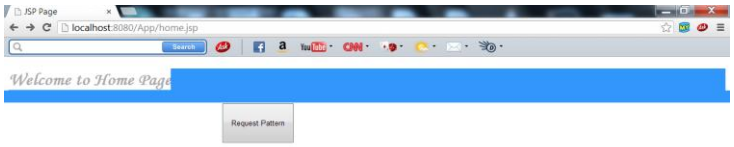
This is about the implementation of the request pattern

Hence we can develop coding for the loan request and if the person gives insurance request we do not want to develop it from the scratch. We can reuse it for any kind of the request**.**

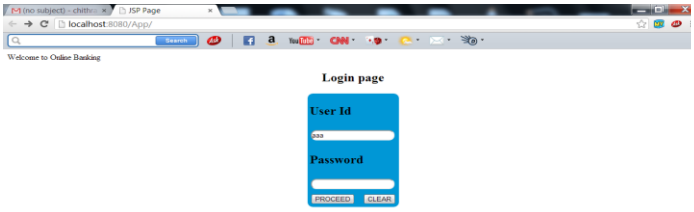
V.Sample Coding:

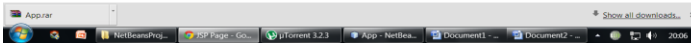
```
import java.io.*;
import java.io.Reader.*;
import java.io.InputStreamReader.*;
abstract class LoanRequest
{
   protected final double base=500;
   protected LoanRequest successor;

   public void setSuccessor(LoanRequest successor)
   {
      this.successor = successor;

   }

   abstract public void processRequest(LoanRequest1 request);

   }
class BranchManager extends LoanRequest
{
   private final double ALLOWABLE =10*base;
   public void processRequest(LoanRequest1 request)
   {
         if (request.getAmount()<ALLOWABLE)
            System.out.println("Branch Manager will approve your loan amount"+request.getAmount());
   else if(successor !=null)
               successor.processRequest (request);
}}


class ZonalManager extends LoanRequest
{
   private final double ALLOWABLE =20*base;
   public void processRequest(LoanRequest1 request)
   {
         if (request.getAmount()<ALLOWABLE)
            System.out.println("Zonal Manager will approve your loan amount"+request.getAmount());
   else if(successor !=null)
               successor.processRequest (request);
}}

class HeadOffice extends LoanRequest
{
   private final double ALLOWABLE =30*base;
   public void processRequest(LoanRequest1 request)
   {
         if (request.getAmount()<ALLOWABLE)
            System.out.println("head office will approve your loan amount"+request.getAmount());
   else if(successor !=null)
               successor.processRequest (request);

   else
      System.out.println("your request for"+request.getAmount()+"needs a board meeting");
```

```
}}

 public class CheckAuthority {


   public static void main(String[] args) throws Exception
   {
   BranchManager manager=new BranchManager();
   ZonalManager Zmanager =new ZonalManager();
   HeadOffice Hmanager=new HeadOffice();

   manager.setSuccessor(Zmanager);
   Zmanager.setSuccessor(Hmanager);
   while(true)
   {
      System.out.println("*******LOAN REQUEST***");
      System.out.println(" Enter the smount to check who
should approve your loan amount");
      double d=Double.parseDouble(new BufferedReader
(new InputStreamReader(System.in)).readLine());
      manager.processRequest(new
LoanRequest1(0,d,"General"));


   }

 }
}
```

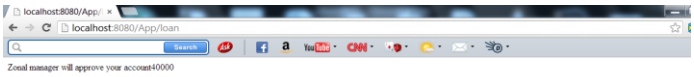


VI.Screen Shots:





After giving username name and password the following screen will be opened.



This is the screenshot of the above coding. In internet banking we are doing loan request if we claim some amount according to that we can know who will approve the loan. If the loan is below 20000 the branch manager will approve the amount. If it is between 20000 and 50000 zonal manager will approve and if 60000 head office will approve else needs to meet with customer.

In This We Have Developed The General Request Codings And Have Called For All Type Of Loan And Insurance Requests.

VII.Conclusion:

In this paper we have discussed about the patterns , Design Patterns and also we have discussed in detail about the request pattern which follows the chain of responsibility pattern. By identifying all these patterns in our organization or application we can achieve better reusability and scalability. In the request pattern we develop the coding for common type of request and can call the request pattern whenever any type of request we need to make in the internet banking application. Our future work is to identify some more patterns in the internet banking application

References:

[1] The Past, Present, And Future Of Software Architecture, *IEEE Software* special issue, 2006.
Philippe Kruchten Henk Obbink, Judith Stafford

[2] A survey On design Patterns and its evaluation methods, A.Meiappane, B.chithra, Dr .V. Prasana venkatesan,E-gov 2012

[3]"Visitor Pattern: Implementation Of Enquiry Pattern For Internet Banking" Mr.A.Meiappane, Ms.J.Prabavadhi, Dr.V.Prasanavenkatesan ,International Journal of Engineering Research and Technology (IJERT), May – 2012.

[4] "strategy pattern: payment pattern for internet banking" , Mr.A.Meiappane, Ms.J.Prabavadhi, Dr.V.Prasanavenkatesan , International Journal of Information Technology and Engineering (IJITE), March 2012.

[5] "Applied Dynamic Chain of Responsibility in Web Application Security" Mahdi Negahi Shirazi, Maryamsadat Hejazi, and Hossein Dolatabadi, International Journal of Computer Theory and Engineering, Vol. 4, No. 6, December 2012