



**Daffodil**  
*International*  
**University**

## ***Assignment on*** **Software Engineering**

**Course Title: “Pouchedebo” delivery system**

**Course Code: CIS332**

**Submitted to:**

Mr. Md. Aktaruzzaman Pramanik

Lecturer

Department of Computing & Information System (CIS)

Daffodil International University.

**Submitted by:**

Prosenjit Chowdhury

ID: 183-16-346

Department of Computing & Information System (CIS)

Daffodil International University.

**Date of Submission: 15/08/2020**

### **SubTask-1**

**The functional requirements for online “PoucheDebo” system:**

Functional Requirements Number	Functional Requirements
FRN1	The system will be capable to generate new customer.
FRN2	System can efficient to query for product item and quantity.
FRN3	Products details will operate by the system.
FRN4	System can able to take Order.
FRN5	System can able to connect customer, seller and delivery with system.
FRN6	System can able to make payment.
FRN7	System can able to store transaction details information.

## **Video Scenario on Anderson**

### **How I react to my customer**

I always be grateful to those who are my customer. It doesn't matter whether they are positive or negative. I always try to show appreciation for them take to share their thoughts with me. This way, my regular and potential customers see that my behave on their values frank feedback. That the consumer opinion is important to it.

### **How I react to my Boss**

When I have facing a bad time at work, my project isn't coming together as planned that time I talk to my boss. It is not acceptable, or professional to vent one's anger on others, or to mis-direct anger on those who do not deserve it. My only choice is to stay calm and not take it personally, as long as he stays within limits. I always be grateful to my boss

### **Think about every statement that is made by the customer and the "expert." How would you respond?**

When a client needs to contact a company's support, he/she always wants to hear the yes answer to all their requests. Here, Anderson says no at first. It makes a bad impression for them. In my opinion, he doesn't make the negative approach. If he always says a professionally negative approach it is very detrimental for him. Otherwise, his client becomes very annoyed. So, in professional life don't say any negative talk to your customer and boss. That will help to maintain a good impression.

### **How do you deal with the boss?**

Successfully managing a difficult boss is a challenge but often feasible. First, i should try to understand the reasons for my boss' difficult behavior. Assuming my boss generally behaves in a fairly reasonable manner, and that his difficult behavior seems to be a result of stress overload rather than his character, chances are good that the behavior can be modified. If my boss' behavior seems to reflect a chronically hostile, abusive style of interacting regardless of the amount of stress in the worksite, the chances are less positive that the behavior can change. In fact, I want to consider seeking counsel from a trusted mentor or human resources professional to evaluate my options.

Second, I have to manage my own negative emotions regarding his behavior so that I do not engage in self-defeating behavior.

## SubTask-2

**Design any two classes by following the SRP principle**

### **Class Name: Customer.**

For the system “PoucheDebo” Customer class is essential. Without customer the system is valueless. That’s why at first, I design Customer Class.

Code: (Using NetBeans platform)

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package pouchedebo.system;

import java.util.Scanner;

/**
 *
 * @author Prosenjit
 */
public class Customer {
    private String Id;
    private String Role;
    static Scanner sc = new Scanner(System.in);
    static Scanner sc2 = new Scanner(System.in);
    public String getId() {
        return Id;
    }
    public String getRole() {
        return Role;
    }
    public static Scanner getSc() {
        return sc;
    }
    public int getId() {
```

```
        return id;
    }
    public String getName() {
        return name;
    }
    public int getAge() {
        return age;
    }
    public String getPhoneNumber() {
        return phoneNumber;
    }
    public String getEmail() {
        return email;
    }
    public String getUsername() {
        return username;
    }
    public String getPassword() {
        return password;
    }
    public String getAddress() {
        return address;
    }
    public void setId(int id) {
        this.id = id;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }
    public void setEmail(String email) {
```

```

        this.email = email;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public void showInsertedInfo(Customer C)
    {
        System.out.println("\n*****Your inserted informations*****");
        System.out.println("ID Number: "+p.getId());
        System.out.println("Patient Name: "+p.getName());
        System.out.println("Age: "+p.getAge());
        System.out.println("Mobile No: "+p.getPhoneNumber());
        System.out.println("Email: "+p.getEmail());
        System.out.println("Address: "+p.getAddress());
        System.out.println("User Name: "+p.getUsername());
        System.out.println("Password: "+p.getPassword());
        System.out.println("❤❤❤ Welcome ❤❤❤\n");
    }
    public void insertInfo()
    {
        Customer C = new Customer();
        System.out.println("Enter Id: ");
        this.id = sc2.nextInt();
        System.out.println("Enter Name: ");
        this.name = sc.nextLine();
        System.out.println("Enter Age: ");
        this.age = sc2.nextInt();
        System.out.println("Enter Mobile Number: ");
        this.phoneNumber = sc.nextLine();
    }

```

```

        System.out.println("Enter Email: ");
        this.email = sc.nextLine();
        System.out.println("Enter Address: ");
        this.address = sc.nextLine();
        C.setld(id);
        C.setName(name);
        C.setAge(age);
        C.setPhoneNumber(phoneNumber);
        C.setEmail(email);
        C.setAddress(address);
        C.setUsername(username);
        C.setPassword(password);
        this.showInsertedInfo(p);
    }
}

```

## **Class Name: Products.**

For the system “PoucheDebo” Products class is also important. That’s why at first, I design Products Class.

Code: (Using NetBeans platform)

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package pouchedebo.system;
import java.util.Scanner;
/**
 *
 * @author Prosenjit
 */
public class Products {
    private String Productld;
    private String ProductNumber;
    private String ProductType;
    static Scanner sc = new Scanner(System.in);

```

```

static Scanner sc2 = new Scanner(System.in);
public String getProductId() {
    return ProductId;
} public String getProductNumber() {
    return ProductNumber;
}
public String getProductType() {
    return ProductType;
}
public static Scanner getSc() {
    return sc;
}
public String ProductDescription() {
    return ProductDescription;
}
public String ProductItems() {
    return ProductItems;
}
public void setProductId(int ProductId) {
    this.ProductId = ProductId;
}
public void setProductNumber(String ProductNumber) {
    this.ProductNumber = ProductNumber;
}
public void setProductType(int ProductType) {
    this.ProductType = ProductType;
}
public void setProductDescription(String ProductDescription) {
    this.ProductDescription = ProductDescription;
}
/**
 *
 */
public void addProducts()
{

```



```

    }

    public void editProducts()
    {
    }

    public void deleteProducts()
    {
    }

    public void searchProducts()
    {
    }
}

```

## Class Diagram:

Customer Class
+user_id: int +user_role_id: int +user_name: string +user_email: string +user_mobile: varchar +user_address: string
+addUser() +editUser() +deleteUser() +searchUser()

Products Class
+product_Id: int +product_Number: String +product_Type: String +product_Description: String +products_Items: String +product_Customer_Id: int
+add Products () +edit Products () +delete Products () +search Products ()

## Data Flow Diagram for the given platform “PoucheDebo”

In this given scenario there are three actors. They are Admin, Customer Type-1, Customer Type-2. Customer Type-1 is who sent his/her product. Customer Type-2 is who carry the product for delivery.

### Context level DFD – 0 level

The context level data flow diagram (dfd) is describe the whole system. The (o) level dfd describe the all user module who operate the system. Below data flow diagram of “PoucheDebo” shows the two user can use the system and Admin operate users.

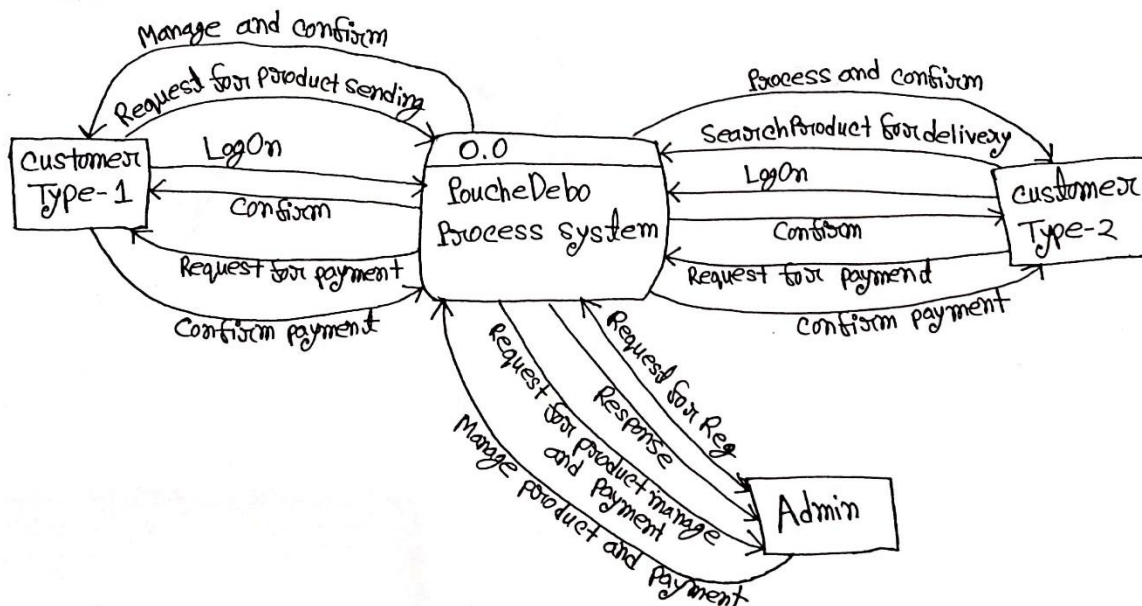


Diagram: 0 level DFD.

## 1st Level DFD

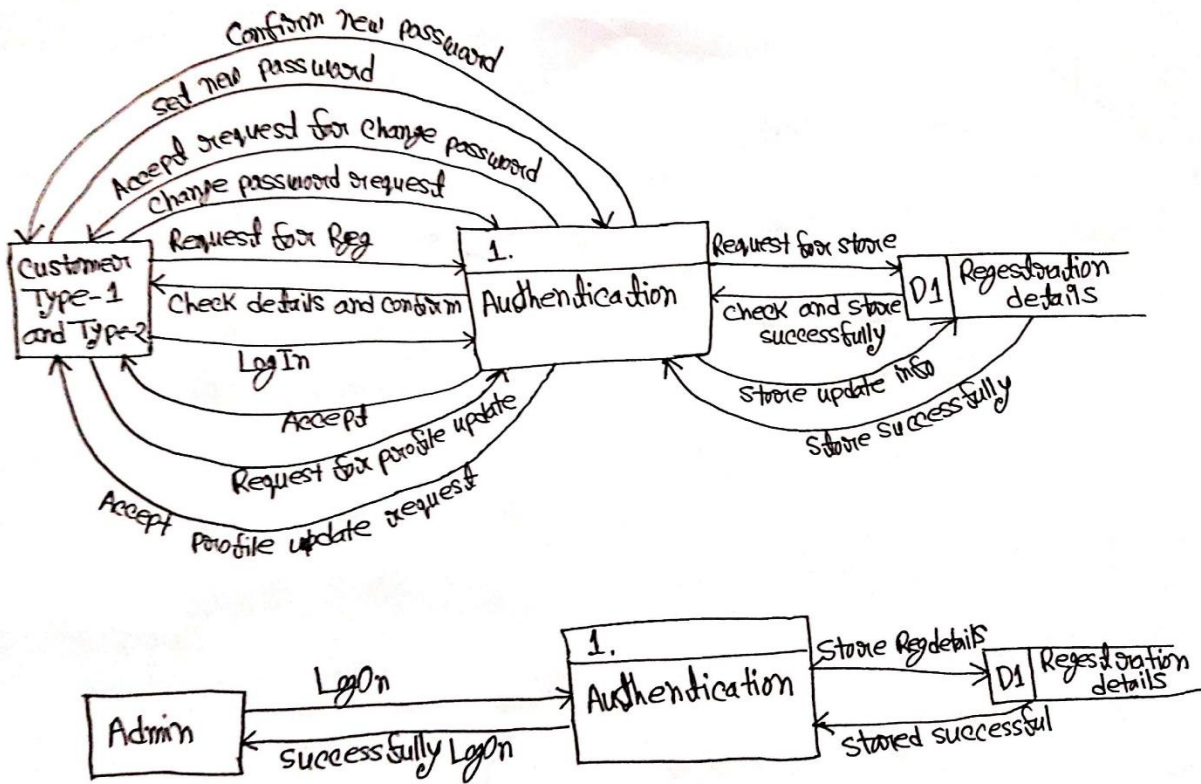


Diagram: 1st level DFD for authentication.

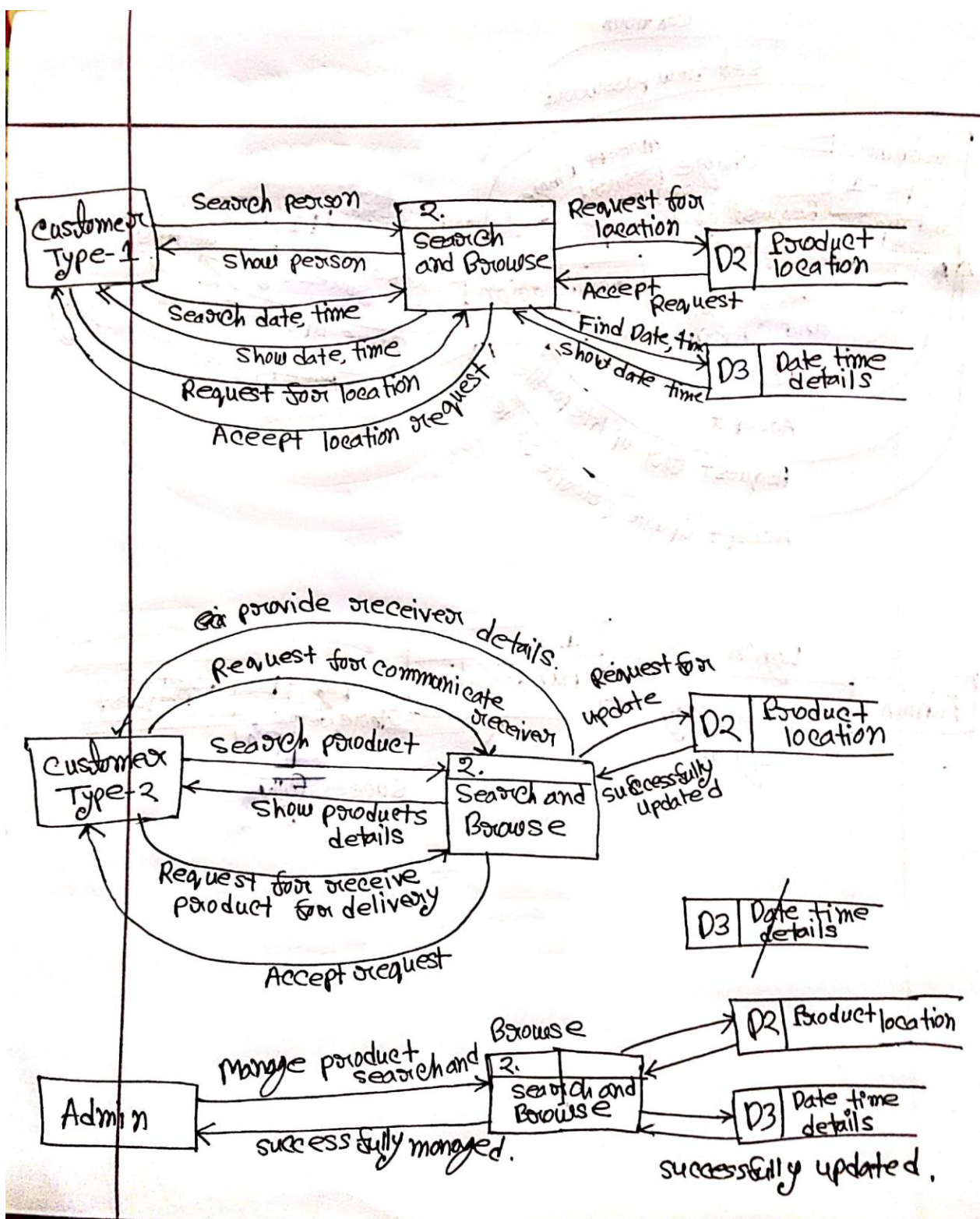


Diagram: 1st level DFD for search and browse.

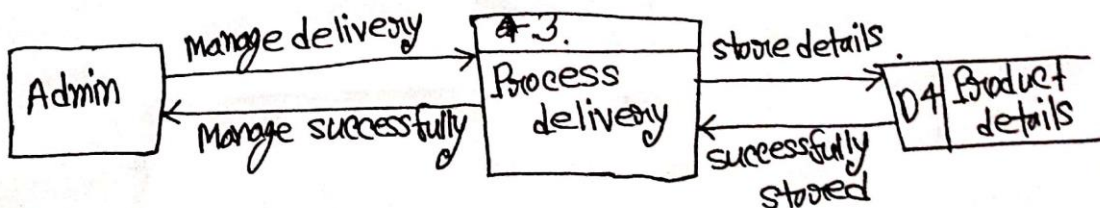
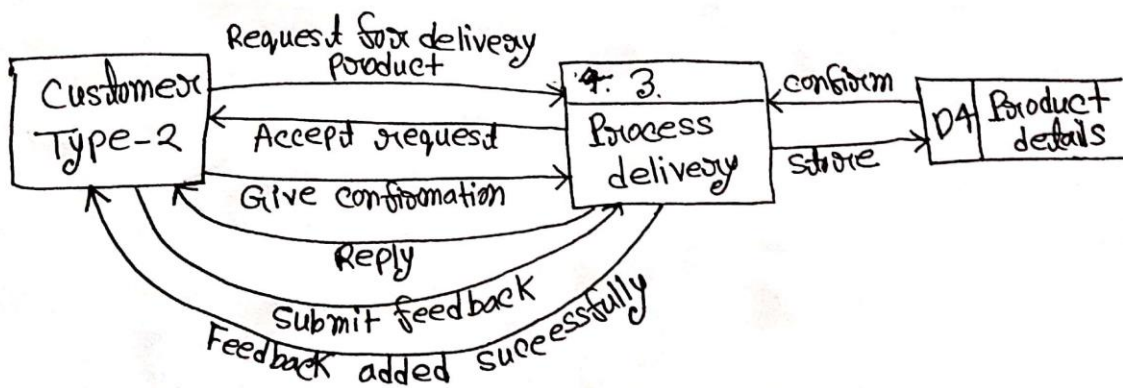
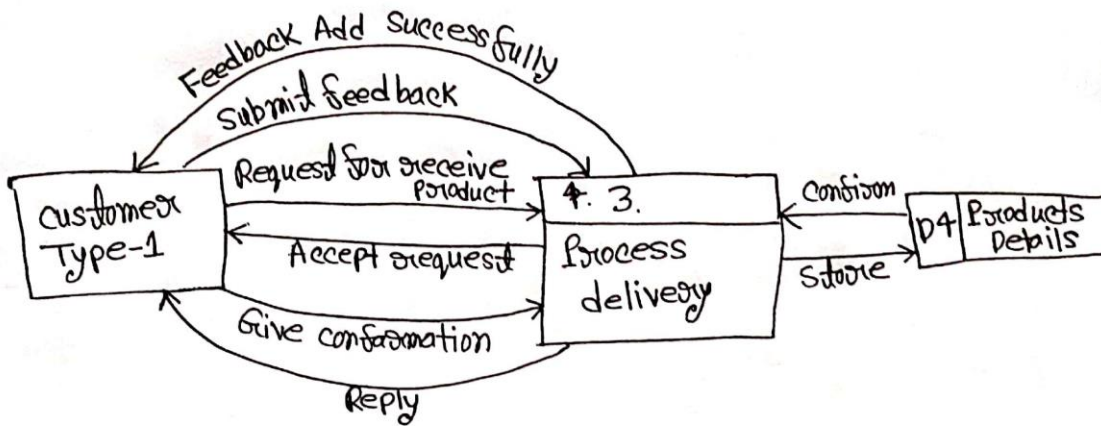


Diagram: 1st level DFD for process and delivery.



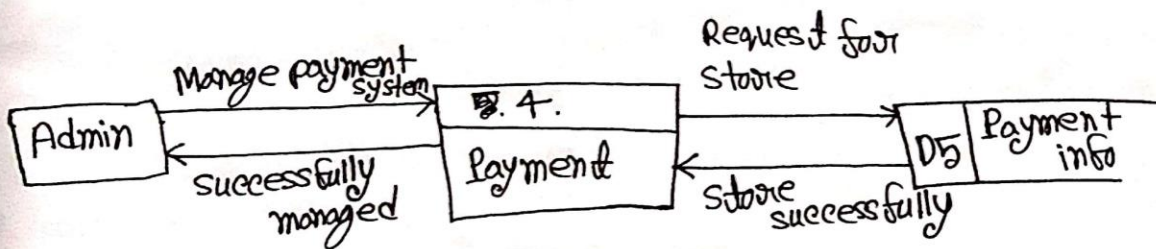
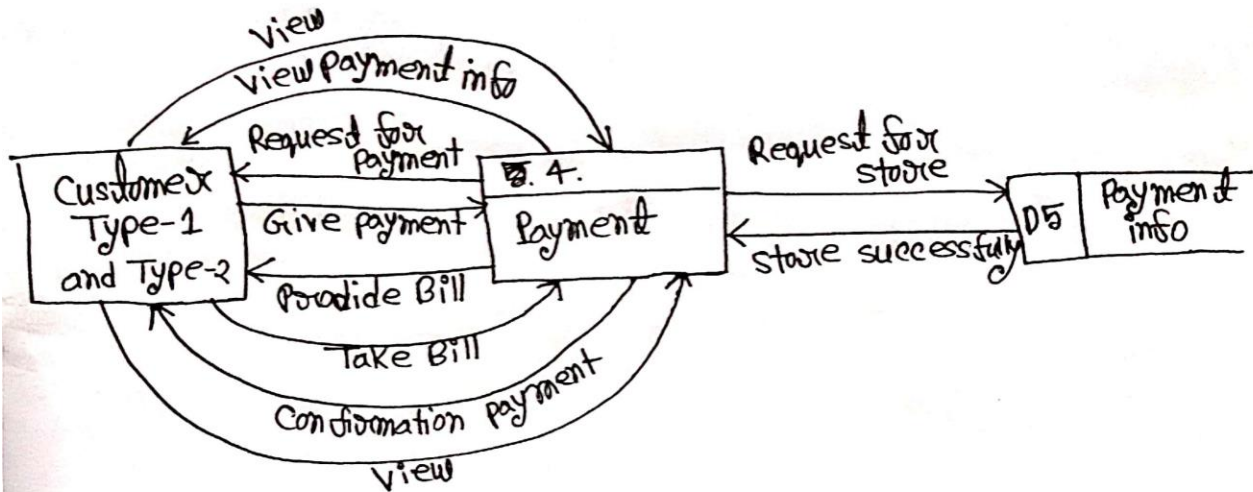


Diagram: 1st level DFD for payment.

### **SubTask-3**

Video testing for Junit, here is my video [Link](#)

Here is my Javadoc documentation, [Link](#)

-----: The End: -----