

## Case Study

### ➤ Problem Statement- Retail Store Management System

Objective- The Spencer Home Appliances shop wants to automate the following processes:

- Booking a product by customer
- Bill generation for customer
- Inventory status check and updation

### ➤ About the Shop-

- Spencer Home Appliances shop owns various models of products, for eg . LG Television T101, LG Refrigerator R601, LG Micro Oven M701 etc.
- The shop keeps stock of bulk of products in same model to meet the customer demand. For eg:  
20 pieces of Sony 24" TV is available in the shop store. The product Ids of each piece will be different but product name is same for all pieces.
- The customer should check the availability of a product before booking the product. As per the availability of the product in the retail store, the customer can book one or more products based on the demand of the customer. Booking has to be done based on Product names.
- The customer can book only one product type at a time.

### ➤ Design and Implement following classes as per given description:

- ✓ **Customer-** Customer should know about his customerID, customerName, contactNumber and the details of products booked by the customer.  
Following are the responsibilities of Customer class:

**Customer** (int customerID,String name,int contactNo):

The Constructor creates Customer object with the given id, name & contact no.

**addBookedProduct** - Public void addBookedProduct(Product obj) :

The method adds the product which is booked by the customer to his bookedProductList.

- ✓ **Product-** Product should know about productID,productName,productStatus and productPrice. (note: productStatus represents the status of the product. It can be "Available" or "Booked". )

Following are the responsibilities of Product class:

**Product** (int productID,String productName, double ProductPrice)

The constructor initializes the value of productID, name & price with the given values .Initially the status of the product is set to "Available".

- ✓ **RetailStore-** RetailStore class should know about the storeName, customers registered with the store and product details available in the store. RetailStore holds following responsibilities. Implement the below mentioned responsibilities for RetailStore class.

**RetailStore(String storeName):**

The constructor initialize the storeName with the given input value.

**Methods-**

- public int **generateCustomerID()**- The method generates the customerID by incrementing the value of previous customer's ID by 1. The value of Customer Id is initially set to zero. For the first customer added to the system the ID should be 1,for the second customer, ID should be 2 and so on. The method should return the ID generated for the customer.
- public int **generateProductID()** - The method generates the productID by incrementing the value of previous product's ID by 1. The value of productId is initially set to zero. For the first product added to the system the ID should be 1,for the second product , ID should be 2 and so on. The method should return the ID generated for the product.
- public void **addCustomer**(String name,int contactNo)- The method creates a customer ID , with that it creates a customer object and then it adds to the customerList of RetailStore.  
Hint:To create the customerId,use generateCustomerID method.
- public void **addProduct**(String name,String status,double price) - The method creates a product ID , with that it creates a product object and then it adds to the productList of RetailStore  
Hint:To create the productId,use generateProductID method.
- public int **checkProductAvailability**(String ProductName) - The method returns the count of products which are in status "Available" with the given name .
- public int **bookProduct**(int CustomerID,String ProductName, int NumberOfProduct) - The method checks for the availability of the given product in the productList of retail store .If the required number of products are available in the store , then that many products are booked which means that ,that many products are being added to the bookedProductList of customer.

If the required number of products are not available then the available quantity of products is being booked.

The method should return the total number of products booked.

If the customer tries to book 7 products and only 3 are available, it should book 3 and the method should return 3.

If the given product is not available in the product list or no stock available, it should return 0.

If the given customer is not available in the customer list, then the method should return -1.

- `public double calculateNetAmount(int customerID, double discount)`- The method takes a customer ID and discount percentage as input parameters and calculates and returns the net amount that the customer need to pay for the entire booking he has made. Net amount= sum of price of all products - discount applicable (note: discount is applied to the sum of price of all products not on individual product price)  
If the given customer is not available in the customer list, then the method should return -1.

- ***Note- Provide necessary getters and setters for all three classes.***