### Wonka Shop

### Problem Statement:

Wonka the Chocolate Shop is an exclusive outlet for members only. They issue membership cards for a year which can be renewed. Our assignment is to implement a billing system for their self-checkout counter.

The customer can select chocolates from the shelves, place them in the cart, and proceed to the checkout counter.

1. Initialize the necessary in-memory data structures in the system from the following data sources:
2. The chocolate shop product master is given in the file “Wonka\_Products.txt”. The file is in the tab-separated format:

**Item Code Item Name Unit Price (Rs.) Available Count**

**Read this file and create an in-memory data structure with the contents of the file.**

**Populate this data into Products\_tbl in the database wdb. [More info about table is given at the end of the assignment]**

**Use following method to implement:**

**Public List<Product> readProductsFromFile(String fullFilePath)**

**This method has been provided to you in Client.java. Please do not change the method signature. You are free to add more data members and methods.**

**The Product bean class has been provided to you. Use the class as provided. You are free to add more methods or data members.**

1. **Authentication:** When a customer is ready to check out, he/she will go to a counter and initiate the procedure by entering a valid membership id.

Some of the members you can test the system with are given in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Card Number** | **Password** |
| 1 | Mr. Cadbury | 12345 | kuch-meetha-ho-jaaye |
| 2 | Mrs. Lindt | 67890 | rin-ki-shakti |

Create in-memory data structure of all the members using following method signature:

**Public List<User> initUsers()**

**This method has been provided to you in Client.java. Please do not change the method signature. You are free to add more data members and methods.**

**The Bean User has been provided to you.**

1. Authenticate a user based on users created above.

If the verification fails due to an invalid membership number, the system should throw a user defined exception “InvalidUserException”. For invalid password, system should throw “InvalidPasswordException”. If authentication successful return status as true.

Use following method to authenticate:

**Public boolean authenticate(User user, List<User> users) throws InvalidUserException, InvalidPasswordException**

**This method has been provided to you in Client.java. Please do not change the method signature. You are free to add more data members and methods.**

1. **Order Entry:** System should display a list of available chocolates along with their prices. Customer will enter product number and quantity. System will keep on prompting to enter product number and quantity till the time user wants to continue. The order must be processed using following method:

**public void processOrder(List<Transaction> transactions)**

**This method has been provided to you in Client.java. Please do not change the method signature. You are free to add more data members and methods.**

As soon as the customer enters the quantity of the last item, system should:

1. Clean up any existing bill item entries for the customer from the database table,
2. Enter the currently selected item details/Transactions (product code, quantity and item subtotal i.e., quantity x price as applicable) into the database table,
3. Update (decrement) the available count for each item in the Products\_tbl.
4. Read the items from the database table and calculate the final bill and display to the customer.

**Public List<Transaction> getPurchaseTransactionsFromDb()**

.

**Checkout Counter Rules:**

1. Customer can choose items shown in the menu only. An error message is shown if customer chooses a wrong item.
2. Entered quantity must be more than 0. If not a user defined exception is thrown and use should see an error message with one more prompt to enter correct quantity.
3. The bill must be printed in ascending order by quantity i.e., with the item having the greatest quantity at the top. **This ordering should be done in Java and not in SQL.**
4. VAT @ 1% is applicable on the Sub Total (total of individual items)

**Public double calculateFinalBillAmount(List<Transaction> purchaseList)**

*Note that the bill alignment is not a crucial issue. The line items shown, the field values, and the totals shown will count in the grading of the assignment.*

**Expected Output on Screen:**

Welcome to Wonka’s Exclusive!

Please enter your credentials:

Membership number: 1234567890123

Password: kuch-meetha-ho-jaaye

Sorry that card number is invalid.

Please enter your credentials:

Membership number: 123456789012341234

Password: kuch-meetha-ho-jaaye

Welcome Mr. Cadbury

Here is what we offer:

Item Code Name Unit Price (Rs.)

================================================

1 Milky Way 12.00

2 Dark Secrets 25.00

3 Swiss Rendezvous 32.00

4 Pulp Fiction 22.00

What would you like today? 5

Sorry, item number 5 is not present in the menu.

What would you like today? 3

Please enter the quantity for – Swiss Rendezvous: -1

INFO: -1 is invalid quantity, Please enter quantity above 0.

Please enter the quantity for – Swiss Rendezvous: 14

Only 12 Swiss Rendezvous are available in the inventory, please enter a valid quantity for - Swiss Rendezvous: 12

Do you want to buy more?(y/n): n

Thank you. Your bill is printed below:

==================================================================

Member Name: Mr. Cadbury Date: 14/03/2011

==================================================================

Item Code Name Unit Price (Rs.) Qty Total

==================================================================

3 Swiss Rendezvous 32.00 12 384.00

1 Milky Way 12.00 4 48.00

==================================================================

Sub Total: 432.00

VAT @ 1%: 4.32

==================================================================

Total: 436.32

==================================================================

Thank you, Mr. Cadbury. Please visit again. Good Bye!

**Implementation Details**

* Use the database table having the schema given below to store the data as indicated in the Order Entry module.
* Use the following queries in MySql command prompt to create database and Table.[Programmatically not required.].

Create database **wdp;**

use **wdp**;

create table **Products\_tbl**(item\_code int, item\_name varchar(50), unit\_price decimal(4), avaialable\_count int);

create table **Bill\_tbl**(member\_id bigint,item\_code int,qty int,total decimal(4));

Note: Please follow the above commands strictly to create db and tables. Any other name/types used will not be accepted.

**Database Information:**

Url: jdbc:mysql://localhost:3306/wdp

Driver class: com.mysql.jdbc.Driver

Username: root

Password: root