# ELTP Java – Shopping Mania

**Marks: 100; Time: 3.5 Hours**

## Guidelines:

1. Use **eclipse-jee-helios-SR2-win32-x86\_64** available on your machine.
2. You have to implement the problem statement in the Java project provided to you.
3. Copy the participant\_workspace.zip onto your machine at C:\ and unzip it.
4. Then, choose the workspace as **C:\participant\_workspace.**
5. A project containing all required files is already provided to you. You have to import the project into your workspace and implement code in this project.
6. Use the project as provided to you **and do not create new project**.
7. The project contains **Client.java** which has all methods that need to be implemented by you.
8. The method signatures have already been provided to you in Client.java. Please do not change the method names or their signatures.
9. **Please read carefully what a method takes as an argument and what it must return. You have to strictly adhere to the method signatures as provided.**
10. You are free to add more classes and methods.
11. Please do not change workspace name, project name, package name, class names or method names provided to you. **If any discrepancy of names is present, your code will not be graded**.
12. If you comment out any methods which are provided in Client.java, or change the names of methods, or change the method signatures your code will not be graded.
13. Some bean classes, exception classes are already provided to you in each project. Please use the same and DO NOT re- create those bean classes.
14. You are free to create more classes and methods as required.
15. Please make sure your code is compiling successfully before submitting.
16. **To submit the code, please upload the entire participant\_workspace.**
17. For your reference, Java Documentation for Java 6 is made available on your machine.

**Evaluation of Assignments**

1. The assignment carries 100 marks, distributed across the various methods asked to implement.
2. Every method must be implemented correctly and completely as per the given specifications. **No marks will be given for partial implementation of a method**.
3. Marks will be given method wise and 100% correctness is expected along with required validations and expected functionalities.

## Problem Statement:

Shopping Mania is an online Shopping System with a wide range of Products. You can create a shopping cart having selected products as per your requirement. You are asked to develop the APIs for this system and provide a console-based test for them in the prototype version.

Here is the functionality for the initial version of your prototype:

## Functional Specifications:

1. **DB Initialization**

The system reads the details of all available Product Categories from a database table called category\_tbl at startup.

For implementing this requirement, you can first create the table using the following DDL statement (not through Java Program, should be done only through MySQL console):

You have to create the database shoppingdb and tables ‘category\_tbl’ and ‘shoppingCarts\_tbl’ using the commands below.

***mysql>*** *create database shoppingdb;*

***mysql>*** *use shoppingdb;*

***mysql>*** *create table* category\_tbl *(category\_id int(11),file\_name varchar(20),category\_name varchar(20));*

*Also populate data into it using the following DML statement (*not through Java Program, should be done only through MySQL console*):*

***mysql>*** *INSERT INTO category\_tbl (category\_id, file\_name,category\_name) VALUES (1, 'ElectronicItems.txt', 'Electronic'), (2, 'Grocery.txt', 'Grocery');*

***mysql>*** *create table shoppingCarts\_tbl(user\_name varchar(20),product\_name varchar(30));*

**The connection details are:**

***Url:jdbc:mysql://localhost:3306/******shoppingdb***

***Driver class:com.mysql.jdbc.Driver***

***Username: root***

***Password: root***

To create a DB connection use the class DatabaseConnectionManager that implements the interface DBConnectionUtil provided to you.

1. **Create Categories**

Implement the functionality to read all the rows from the category\_tbltable using following method signature. Create and close the connection within the method.

|  |  |
| --- | --- |
| Method that you should implement for this: | **public List<Category> readAllCategoriesFromDb()** |
| Classes provided to you: | Bean class called **Category** |
| Note: | Please do not change the method signature.  You can add more members to the bean or client, as needed. |
| Marks: | 10 |

1. **Populate Categories**

Get the category file names from List of Categories obtained from the previous method.

Refer to these names and read the files provided in the same project in Categories folder.

The text files containing the details of the products in the category in a comma-separated format as:

*Product\_Name, Brands (colon[:] separated names)*

Make sure you do not change the path and names of these files. Use the relative path to access the Category files.

**Implement the following method to read the Category files:**

|  |  |
| --- | --- |
| Method that you should implement for this: | **public void populateCategoryProducts(List<Category> categoryList)** |
| Classes provided to you: | Bean class called **Product** |
| Note: | Please do not change the method signature.  You can add more members to the bean or client, as needed. |
| Marks: | 25 |

The system should then proceed to present a menu to the user for the following functionalities. Write your own method for creating this menu.

***Welcome to Shopping Mania!!***

***Please select an option:***

***1. Create a Shopping Cart***

***2. Display a Shopping Cart***

***3. Exit***

**All the above initialization must be done before the user is presented with the menu.**

1. **Create a Shopping Cart**

If this menu option is chosen, the system should prompt the user to enter the following:

1. **Take User Input as per following rules**

* Write your own method in Client.java to implement functionality to take user input for user name and products.
* User name is used to identify the Shopping Cart.
* A Shopping Cart can have multiple occurrences of the same product.

***Please enter user name for the Shopping Cart:***

***ShriDevi***

***Please enter the product name:***

***Washing Machine***

***Add more products to Shopping Cart? (y/n)***

***y***

***Please enter the product name:***

***LED***

***Add more products to Shopping Cart? (y/n)***

***y***

***Please enter the product name:***

***LED***

***Add more products to Shopping Cart? (y/n)***

***y***

***Please enter the product name:***

***Air Conditioner***

- After accepting the list of products, create a Shopping Cart using the next method given below.

1. **Create a Shopping Cart using following method.**

In method below, ‘**userInputProductNames’** is the list which contains the products accepted from the console.This product names list may contain the products which do not exist in the repository. Within the method, apply validations to ignore the product names which do not exist in the categories’ Product list.

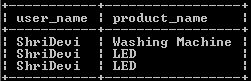
|  |  |
| --- | --- |
| Method that you should implement for this: | **public ShoppingCart createShoppingCart( String userName, List<String> userInputProductNames,List<Category> categories)** |
| Classes provided to you: | Bean class called **ShoppingCart** |
| Note: | Please do not change the method signature.  You can add more members to the bean or client, as needed. |
| Marks: | 15 |

1. **Saving Shopping Carts**

After the Shopping Cart is created in above method, it should be saved in a table named *shoppingCarts\_tbl*.

Note that a Shopping Cart is identified by the user\_name. A user can add the same product to his cart as many times. Refer to the image below for clarification.

If there are multiple products entered by a user then there should be a new row for each product entered.



**Implement the functionality to save the product details into shoppingCarts\_tbl table using following method:**

|  |  |
| --- | --- |
| Method that you should implement for this: | **public void storeShoppingCartIntoDB(ShoppingCart cartObj )** |
| Classes provided to you: | Bean class called **ShoppingCart** |
| Note: | Please do not change the method signature.  You can add more members to the bean or client, as needed. |
| Marks: | 10 |

1. **Display a Shopping Cart of a particular User**

If this menu option is chosen, the system should prompt to enter the name of user whose Shopping Cart has to be loaded.

***Welcome to Shopping Mania!!***

***Please select an option:***

***1. Create a Shopping Cart***

***2. Display a sorted Shopping Cart***

***3. Exit***

***2***

***Please enter the name of the user whose Shopping Cart needs to be shown:***

***Madhuri***

***Failed to load Shopping Cart: The system cannot find the user requested***

***Please enter the name of the user whose Shopping Cart needs to be shown:***

***ShriDevi***

***Shridevi has following products in her cart:***

***WashingMachine***

***LED***

***LED***

If a username is found in database, return all products of Users Shopping Cart in it. If the username is not found, method must throw ShoppingCartNotFoundException with appropriate error message without exiting the application. Do not catch exception ShoppingCartNotFoundException in the method.

Sort the products of the Shopping Cart obtained in descending order of product name.

**Implement the following method in Client.java. This method should retrieve the Shopping Cart for a particular user, sort the Shopping Cart by product name and return Sorted Product list.**

**Write your own method to display the contents of the cart.**

|  |  |
| --- | --- |
| Method that you should implement for this: | **public List<Product> sortShoppingCart(String userName) throws ShoppingCartNotFoundException** |
| Classes provided to you: | Exception class called **ShoppingCartNotFoundException** |
| Note: | Please do not change the method signature.  You can add more members to the bean or client, as needed. |
| Marks: | 40  5 marks for loading the Shopping Cart with valid user name  10 marks for handling the case with invalid username  20 marks for Sorting |

**Sample Output:**

***Welcome to Shopping Mania!!***

***Please select an option:***

***1. Create a Shopping Cart***

***2. Display a sorted Shopping Cart***

***3. Exit***

***1***

***Please enter user name for the Shopping Cart:***

***ShriDevi***

***Please enter the product name:***

***Washing Machine***

***Add more products to Shopping Cart? (y/n)***

***y***

***Please enter the product name:***

***LED***

***Add more products to Shopping Cart? (y/n)***

***Y***

***LED***

***Add more products to Shopping Cart? (y/n)***

***n***

***Done, the Shopping Cart is created. Thank you!***

***Please select an option:***

***1. Create a Shopping Cart***

***2. Display a sorted a Shopping Cart***

***3. Exit***

***2***

***Please enter the name of the user whose Shopping Cart needs to be shown:***

***Madhuri***

***Failed to load Shopping Cart: The system cannot find the user requested)***

***Please enter the name of the user whose Shopping Cart needs to be shown:***

***ShriDevi***

***Shridevi has following products in her cart:***

***WashingMachine***

***LED***

***LED***

***Please select an option:***

***1. Create a Shopping Cart***

***2. Load a sorted a Shopping Cart***

***3. Exit***

***3***

***Good bye!***