University of Technology, Jamaica
School of Computing and Information Technology
CIT3009 – Advanced Programming

Facilitator(s):	Christopher Panther | Horrett Scarlett		
Assessment:	Group Assignment (Group size, no less than 3 persons, maximum 4 persons)		
Given week of:	September 27, 2022	**Due week of:**	November 11, 2022

Jan's Wholesale and Retail is a small supermarket located in Papine. The management of the supermarket is not satisfied with the service they are delivering to their customers especially on the weekends and at the end of the month when the supermarket becomes crowded. The lines in the supermarket are extremely long because the clerks have to manually write up the invoices before the customer pays for their items at the cashier. The cash register is currently only used to collect the money for payment. It does not keep a record of the items being checked out so that the inventory may be updated.

The management has decided to computerize their system to help to improve the customer experience and reduce the lines. They hope that the system will also improve their own internal procedures.

You have been contacted to develop the system for Jan's Wholesale and Retail. After meeting with the management and the other employees the requirements reveal the following modules:

Customer Database

The management wants to have a good rapport with their customers and thus would like to maintain strong communication. They wish to maintain a database that will store contact details of their customers. The management allows the general public to shop in the store but customers with membership will be given a ten percent (10%) discount subtracted from the total before tax (15%) is added.

Store the following data on the customers:

- Customer ID – string
- Name – string

- DOB – date
- Address – string
- Telephone – string
- Email – string
- Date of Membership – date
- Date of Membership Expiry - date

Staff Database

Management also wants to maintain data on their staff for their own bookkeeping protocols. There are **3 types of employees, Manager, Supervisor and Line worker.**

The 3 departments in the business are:

- Management
- Inventory
- Accounting and Sales – Cashiers also fall within this department.

Store the following data on the departments:

- Department Code – string
- Name - string

Stock and Inventory

Keeping track of the inventory is very important to the running of the business. The management wishes to use the system to view the stock of particular products in the warehouse to determine when to replenish the supply from their suppliers. The system should also aid in the stock-taking process so that the inventory operators may update product information, delete product information, insert new products and their information.

Store the following data on the products:

- Product Code – string
- Name – string

- Short Description – string
- Long Description – string
- Items in Stock – integer
- Unit price - float

Check Out

The cashiers will be using the check-out module to assist customers in making payment. The cashiers will use this module to create an invoice and accept payment from the customer. Upon finalizing payment, the items in the inventory should be updated.

Store the following information on invoices:

- Invoice Num – integer
- Billing Date – date
- Item – string
- Quantity of item – int
- Cashier – string
- Customer – string

NB. Pay close attention to, many to many relationships and normalization

Sales Reports

The accounting personnel need to print reports on various aspects of the system. They should be able to pull the data on a particular product between a particular period. This should aid them in analyzing the customer demand for the product to determine if it should be restocked, or the rate at which that product should be restocked.

Requirements

Design a JAVA application that will meet the requirements discussed in the meeting. Your application should be built using the ***Client/Server*** architecture. ***Your Database should be located on the server machine*** where the client will make requests to the

server over the network. **Clients cannot communicate with the database**. The server will process the corresponding request from the client. Your application should be developed with a **graphical user interface** that will aid the cashier in the check-out process.

The management has decided to buy the adjacent store to the supermarket to expand. The expansion requires that the management add more cashiers to check out customers. The server thus needs to be threaded to accommodate multiple client cashiers that will connect to it to interact with the database.

Instructions:

1. System Name: Jan's Wholesale & Retail Management System (Jans W&RMS)
2. Please stick to programming conventions such as **proper indentation**. Include **comments** (one to briefly describe the function and any other special/important lines).
3. Classes begin with **capital** letters (ex: **Person**), methods begin with **common** letters followed by **initial** caps (**ex: getName( )**)
4. Variables should have **meaningful** names. Refrain from using 'x' or any other single letter variables. Variables also begin with **common** letters followed by **initial** caps (**ex: regStatus**)
5. Logging should be on the server side
6. Use appropriate exception handling
7. Develop your group solution with the use of a GitHub Repository or its equivalent to host the project's source code files, to which each member of the group is expected to contribute. Proof of members' contribution will be verified via the project's history logs.
8. Your lab tutor is to receive an invitation from each group, to collaborate on their project, within seven (7) days of this document being distributed.

You are not limited to the database fields given above for the database tables. Feel free to add other fields or develop your own database schema.

Development Tools:

- Eclipse IDE or NetBeans IDE (JDK 1.8 or higher)
- MySQL 8 current version (Administered with MySQL Workbench or PhpMyAdmin)
- MySQL Connector/J (JDBC)
- Hibernate 5 (Object Relational Mapping)
- Apache Log4J2 (Logging and Audit Trails)
- Spring Framework (Inversion of Control, Already Packaged in Netbeans)
- Git or its equivalent (Version Control)

Deliverables:

- Group Report
    - Details of each group member's contribution
    - Signed Declarations of Authorship Forms - one for each group member
- Application User Manuals
    - Staff User Guide (Document or Screen Recording)
- Server-side application
- Client-side application

Assessment:

- Application Documentation
    - User Manuals, UML Class and Database ER Diagrams **5%**
    - Comments, Programming Conventions and documentation **5%**
- Functioning Software solution
    - Graphical User Interface **10%**
    - Client/Server Networking Model **15%**
    - Database Connectivity **15%**
    - Exception Handling & Logging, User & data input validation **10%**
    - Classes, Interfaces, Inheritance and Polymorphism **15%**
    - Core Functionality described above **25%**
        - ♠ Customer Membership/Records Management
            - C.R.U.D. [8 marks]
        - ♠ Member discount correctly applied [5 marks]

- ♠ Staff data/record management
  - C.R.U.D. [8 marks]
- ♠ Inventory management
  - Management dashboard [5 marks]
  - C.R.U.D. operations [8 marks]
  - Real-time inventory update [8 marks]
- ♠ Checkout
  - Invoice generation (Printable) [10 marks]
  - Real-time inventory update [8 marks]
- ♠ Sales report
  - Pull product data for specific period (Printable) [10 marks]

- 45 minutes grading interview – all group members must be present and able to fully present the project. Absence from the grading interview means no grade.

Project Submission

Your lab tutor will communicate to you, what their preferred medium of project submission is. Projects must be submitted on or before the due date.

Late Submission

As per the university's policy, late submission of projects will attract a penalty as follows:

Time of Submission after Deadline	Penalty (%)
One Day (or any part thereof)	10
Two Days (or any period > 1 day, but <= 48 hrs.)	20
Three Days (or any period > 2 days, but <= 72 hrs.)	50
Beyond Three Days	100