

Store Operations Management System

WIX1002 Fundamentals of Programming 25/26

Project Overview

You are a computer science student on semester break and currently working part-time in a retail company called GoldenHour that sells watches in Kuala Lumpur City Centre. Knowing your background, your manager has assigned you to design and implement your store's own **operations management system** that facilitates daily operations and improve efficiency, provided with an extra income. Of course you would say yes, I mean, this is a chance to learn, plus another valuable experience to be added into your resume (and extra money!).

Project Prerequisite

You need a strong understanding of the following concepts before starting, they are very crucial to complete this assignment:

- Object-Oriented Programming (OOP) – Apply the concepts of classes, inheritance and polymorphism.
- Input/Output File Handling – Read and write data from files (either CSV file or database).
- Search Mechanisms – Implement a searching algorithm (e.g. linear search, etc.).
- Sort Mechanisms – Implement a sorting algorithm (e.g. bubble sort, etc.).
- Basic Database Operations – Understand basic SQL commands to add, update, or retrieve data.
- Error Handling – Handle invalid inputs and unexpected program errors properly.

Outline

Project Overview -----	1
Project Prerequisite-----	1
Outline-----	2
Basic Features (8 marks) -----	3
Login/Logout and Employee Registration (1 mark) -----	3
Attendance Log (½ mark)-----	4
Stock Management (2 marks) -----	5
Sales System (1 mark) -----	7
Search Information (1 mark) -----	8
Edit Information (1 mark) -----	9
Storage System (1 mark) -----	10
Data Load State (½ mark) -----	10
Extra Features (Max 4 marks) -----	11
Graphical User Interface (2 marks) -----	11
Auto Email to Headquarters (1 mark) -----	11
Data Analytics (1 mark) -----	11
Filter and Sort Sales History (1 mark) -----	12
Employee Performance Metrics (1 mark) -----	12
Advice and Reminders -----	13
Contact Me -----	13

Basic Features (8 marks)

To ease your code, the information on employees, models, and outlets will be given for your use in CSV files.

The CSV files are available to download through the provided link.

<https://github.com/imxnbtrsyia/WIX1002-FOP/tree/main/FOP-Demo-2526>

The examples shown in this document (blue boxes) are provided **for reference purposes only**. You are encouraged to **use your own creativity and design** when implementing your system.

Login/Logout and Employee Registration (1 mark)

Before accessing the system, employees must log in using their assigned Employee ID and their password. If the Employee ID does not exist, or the password entered is incorrect, the system should display an unsuccessful login attempt message. After a user logs out, another employee should be able to log in without restarting or terminating the program.

Only **your manager** is authorized to register new employees. This feature allows the manager to add new employees to the system by assigning them a unique Employee ID (no duplicates), role (Part-time or Full-time), and password.

```
==== Employee Login ====
Enter User ID: C6001
Enter Password: a2b1c0

Login Successful!
Welcome, Tan Guan Han (C60)
```

```
==== Employee Login ====
Enter User ID: C6001
Enter Password: a0b1c2

Login Failed: Invalid User ID or Password.
```

==== Register New Employee ===

Enter Employee Name: Siti Ros binti Rossi

Enter Employee ID: C6013

Set Password: k6l7m8

Set Role: Part-time

Employee successfully registered!

Attendance Log (½ mark)

This feature allows employees to clock in and clock out for attendance tracking. The system should automatically record the **date and time** for both actions. Upon clocking out, the system must calculate and display the **total working hours** of the day for the employee.

==== Attendance Clock In ===

Employee ID: C6001

Name: Tan Guan Han

Outlet: C60 (Kuala Lumpur City Centre)

Clock In Successful!

Date: 2025-10-13

Time: 09:58 a.m.

==== Attendance Clock Out ===

Employee ID: C6001

Name: Tan Guan Han

Outlet: C60 (Kuala Lumpur City Centre)

Clock Out Successful!

Date: 2025-10-13

Time: 06:05 p.m.

Total Hours Worked: 8.1 hours

Stock Management (2 marks)

This module enables employees to manage and update stock-related information in the store.

1. Morning and Night Stock Count

Employees must perform stock counting twice a day, once during opening (morning count) and another before closing (night count). For **each model**, the employee will key in the total number of items counted in-store (assuming all models are displayed). The system should display a confirmation message if the numbers match. However, if the numbers do not tally, a warning message should appear.

```
==== Morning Stock Count ====
Date: 2025-10-13
Time: 10:00 a.m.

Model: DW2300-1 - Counted: 2
Store Record: 2
Stock tally correct.

Model: DW2300-2 - Counted: 2
Store Record: 1
! Mismatch detected (1 unit difference)

// (Repeat for each model)

Total Models Checked: 15
Tally Correct: 14
Matches: 1
Morning stock count completed.
Warning: Please verify stock.
```

2. Stock In and Stock Out

This feature records stock movements between outlets or from the service center.

- Stock In - When new models are received from the service center or other outlets.
- Stock Out - When models are transferred out to another outlet.

Each stock movement must generate a **text-based receipt** containing:

- Transaction Type (Stock In/Out)
- Date and Time (automatic)
- From (Outlet Code)
- To (Outlet Code)
- Model Name(s) with Quantity
- Total Quantity
- Name of Employee in Charge (automatic, based on the currently logged-in account)

Receipts should be saved by **date**, ensuring that records from previous days are not overwritten. All stock movements for the **same day** should be **appended** to the same file.

==== Stock In ===

Date: 2025-10-13

Time: 02:50 p.m.

From: HQ (Service Center)

To: C60 (Kuala Lumpur City Centre)

Models Received:

- DW2400-2 (Quantity: 2)
- DW2300-3 (Quantity: 1)

Total Quantity: 3

Model quantities updated **successfully**.

Stock In **recorded**.

Receipt generated: receipts_2025-10-13.txt

Sales System (1 mark)

This module records all sales transactions between the employee and customer. When a sale is recorded, the system must automatically capture the **transaction date and time**, as well as the **employee in charge** (based on the currently logged-in account) in each transaction record.

The employees should be able to input:

- Customer Name
- Item(s) Purchased
- Transaction Method (cash, debit/credit card, e-wallet, etc.)
- Total Price

Each sale must automatically update the product stock and generate a **text-based receipt** summarizing the transaction details. Receipts should be saved by date, ensuring that records from previous days are not overwritten. All transactions for the **same day** should be **appended** to the same file.

```
==== Record New Sale ===

Date: 2025-10-13
Time: 02:50 p.m.

Customer Name: Zikri bin Abdullah
Item(s) Purchased:
Enter Model: SW2500-1
Enter Quantity: 1
Unit Price: RM845
Are there more items purchased? (Y/N): N

Enter transaction method: Credit card
Subtotal: RM845

Transaction successful.
Sale recorded successfully.
Model quantities updated successfully.
Receipt generated: sales_2025-10-13.txt
```

Search Information (1 mark)

This feature allows employees to quickly retrieve information related to stocks and sales.

1. Stock Information

Employees can search by model name to view current stock availability in-store and across other outlets.

2. Sales Information

To verify the authenticity of transactions, employees can search sales records by date, customer name, or model name.

```
==== Search Stock Information ====
```

Search Model Name: DW2300-4

Searching...

Model: DW2300-4

Unit Price: RM349

Stock by Outlet:

KLCC: 1 MidValley: 1 Lalaport: 3 KL East: 1

Nu Sentral: 2 Pavillion KL: 2 MyTown: 1

```
==== Search Sales Information ====
```

Search keyword: Zikri

Searching...

Sales Record Found:

Date: 2025-10-13 Time: 02:50 p.m.

Customer: Zikri bin Abdullah

Item(s): SW2500-1 Quantity: 1

Total: RM845

Transaction Method: Credit Card

Employee: Tan Guan Han

Status: Transaction verified.

Edit Information (1 mark)

This feature allows employees to correct mistakes or update records when errors occur. Editable fields include:

- Stock-related data (daily stock count and model stock levels)
- Sales information (customer details, transaction method, total price, etc.)

```
==== Edit Stock Information ====
```

```
Enter Model Name: DW2300-4
```

```
Current Stock: 1
```

```
Enter New Stock Value: 3
```

```
Stock information updated successfully.
```

```
==== Edit Sales Information ====
```

```
Enter Transaction Date: 2025-10-13
```

```
Enter Customer Name: Zikri bin Abdullah
```

```
Sales Record Found:
```

```
Model: SW2500-1 Quantity: 1
```

```
Total: RM845
```

```
Transaction Method: Credit Card
```

```
Select number to edit:
```

```
1. Name    2. Model    3. Quantity    4. Total
```

```
5. Transaction Method
```

```
> 4
```

```
Enter New Transaction Method: E-wallet
```

```
Confirm Update? (Y/N): Y
```

```
Sales information updated successfully.
```

Storage System (1 mark)

All records (employee, models, attendance, stock, sales, etc.) should be stored and retrieved from a database (preferably SQL-based), or alternatively from a CSV file. Any changes made in the system (e.g. new sales, attendance logs, or stock adjustments) the stored data must be modified accordingly.

The system should maintain separate data records for:

- Employees
- Models
- Attendance
- Sales

Students may decide what specific attributes to include for each dataset, as long as the stored data supports the system's functions.

Data Load State (½ mark)

When the system starts, existing data must be loaded from storage and converted into objects (based on Object-Oriented Programming principles). Each dataset (e.g., Employee, Model, Sales) should be represented as a corresponding class, allowing structured and object-oriented data manipulation throughout the program.

Extra Features (Max 4 marks)

Graphical User Interface (2 marks)

Design a user-friendly and simple Graphical User Interface (GUI) that allows users to perform the given key tasks:

- User log in and log out
- User clock in and clock out
- Current stock viewing (read-only inventory)
- Sales recording

The GUI must be created using your programming language (e.g. JavaFX or Java Swing), and not using a web page built (e.g. HTML, CSS, or JavaScript).

Auto Email to Headquarters (1 mark)

Automatically send a summary of the daily sales report to the headquarters via email.

- The daily sales receipt file generated must be attached to the email.
- The email should include a short sales summary in the body, stating the total sales amount and the report date.
- The email is expected to be sent before 10:00 p.m. daily (after completing all sales entries for the day and before the store closes).
- The email must be sent to the student's own email address as proof that the automated email function works correctly.

Data Analytics (1 mark)

Provide a simple analytical summary of the sales or stock performance.

- Calculate total sales per day/week/month, most sold product model, or average daily revenue.
- Display the results either numerically or using simple charts (if GUI is implemented).
- Analytics should use data retrieved from the stored records (not manually input).

Filter and Sort Sales History (1 mark)

Allow the user to filter and sort the sales records for easier analysis.

- View transactions between two specific dates.
- Display the total cumulative sales within the selected date range.
- Sorting can be by date (ascending or descending), amount (lowest to highest and vice versa), or customer name (alphabetically).
- The filtered and sorted transactions must be displayed in a tabular format for clear insight.

Employee Performance Metrics (1 mark)

This feature provides an overview of employee performance based on their total sales amount within a given period.

- The system should calculate and display the total sales value achieved by each employee.
- Employees are listed in descending order of performance (highest to lowest total sales).
- Each record should include the employee's name, total sales amount, and the number of transactions handled.
- This feature is restricted to the manager account, as it contains confidential performance data (only the manager can view this report; regular employees are not allowed to access it).

Advice and Reminders

This assignment is meant to help you learn as a team, so **please** contribute actively and avoid being a '*free rider*'. Make sure to update your group's GitHub repository weekly to show consistent progress. Remember, it's okay to take things one step at a time. If a part of the task feels challenging to the point it overwhelms you, pause, rest all you need, and reach out for help when needed. Always prioritize your well-being. And please avoid procrastinating! Start early and pace yourselves so you can submit something you're proud of. It's better to start early than regretting it later!

Good luck 😊

Contact Me

If you have any inquiries or in need of further clarification for this assignment, do not hesitate to contact me, Nur Iman Batrisyia binti Husni Rizal, through

- WhatsApp (+6019-208 9702)
- Email (23001812@siswa.um.edu.my)

Feel free to connect with me in LinkedIn too:

<https://www.linkedin.com/in/nur-iman-batrisyia-husni-rizal/>