

Mawlana Bhashani Science & Technology University

Department of Information & Communication Technology

Santosh, Tangail-1902, Dhaka, Bangladesh

A Project On

Shopify Manager

A Console Platform for Managing Orders, Inventory,
and Employee Records

By,

Name : Anirban Rahi

ID : IT-23043

Session: 2022-2023

Supervisor,

Dr. Ziaur Rahman

Associate Professor

Dept. of ICT, MBSTU

Declaration

This is to certify that the work presented in this project is carried out by the candidate under the supervision of **Dr. Ziaur Rahman** in the Department of Information and Communication Technology, MBSTU, Tangail, Bangladesh. It is also declared that neither of this project has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given

Supervisor

Signature of

DR. ZIAUR RAHMAN

Associate Professor

Dept. of ICT, MBSTU

ACKNOWLEDGEMENT

We would like to express my gratitude towards guide **Dr. Ziaur Rahman** for the useful instructions, remarks and for giving his valuable guidance and inspiration throughout the learning process of this report. Furthermore, we would like to thank him for making available all the facilities for the successful completion of this work and other staff members of the ICT Department for their valuable help. It is with humble gratitude & sense of indebtedness, we thank our respected and esteemed Dean for his valuable guidance, suggestion and constant support which lead towards successful completion of this work. We take this opportunity to express our sincere gratitude to all those who helped us in various capacities in undertaking this project and devising the report. We are privileged to express our sense of gratitude to our respected teacher **Dr. Ziaur Rahman** whose unparalleled knowledge moral fiber and judgment along with her immense support in completing the project. We take this opportunity also to thank our friends and contemporaries for their cooperation and compliance.

CONTENT PAGE

1	Introduction-----	5
2	Project Overview-----	6
3	Class Descriptions and Methods-----	7
	3.1 Authentication-----	7
	3.2 Employee-----	7
	3.3 Inventory-----	8
	3.4 ProcessOrder-----	8
	3.5 Transaction Records-----	9
	3.6 Update Inventory-----	9
4	Main Method-----	10
5	File System-----	11
6	Implementation Details-----	11
7	System Requirement-----	12
8	Future Work-----	12
9	Conclusion-----	13
10	References-----	13

Introduction

This report provides a detailed overview of the Shop Management System developed during the course. The objective of this project is to create a software application that simplifies the management of shop operation, including employee management, inventory control, and order processing. The system aims to enhance operation efficiency and improve overall user experience.

The development of this system was motivated by a strong need to address common challenges faced by shop owner, particularly in small to medium-sized enterprises. Inventory mismanagement, lack of transparency in employee roles, and the absence of a structured ordering process can hind a shop's growth and affect customer satisfaction.

The kay aspect of the project was to ensure the system's usability and flexibility, making it accessible to users with minimal technical expertise. Through an intuitive command line interface, users can easily interact with the software and perform required tasks with minimal training. Moreover, they system's modular structure allows for future expansion, so additional functionalities can be integrated as the business grows.

Project Overview

The Shop Management System is a command-line application designed to facilitate various shop management tasks. Key functionalities include:

- **Authentication:** Ensures secure access for employees and manager
- **Employee Management:** Allows tracking of employee information and performance
- **Inventory Management:** Provides tools for monitoring stock levels and managing product details.
- **Order Processing:** Facilitates the handling of customer orders.
- **Transaction Records:** Maintains comprehensive logs of all transactions.

Class Descriptions and Methods

This section details the main classes and their methods, describing their roles in the system.

1.Authentication

- Methods:
 - clearcurrentline(): Replace the line with empty string.
 - userInput(int low, int high, string s): Here all the inputs are taken from the user. It ensures handling edge cases by using low and high number with a suitable message “s”.
 - loadData(): It loads data from csv file and saves it in a private variable.
 - LoginMenu(): Takes email and password from user and verifies it using the private loaded data;
 - changeCredentials(): It allows to change the user and admin email and password. By default, the admin email and password are assigned to “admin”. One can set these suitable credentials by using this admin credentials. If someone wants to change the admin credentials then he needs to provide the previous admin credentials to change.
 - Verification(): It verifies the given admin email and password to change any credentials.

2.Employee

- Method:
 - clearcurrentline(): Replace the line with empty string
 - userInput(int low, int high, string s): Here all the inputs are taken from the user. It ensures handling edge cases by using low and high number with a suitable message “s”.
 - loademployerdata(): This loads the all associated employee data from csv file and saves it in a map.

- `updateFile()`: This can update the file according to how the user wants to change the information.
- `printdata()`: It prints the staff of a specific category.
- `showemployeedata()`: It show the employee data in a alphabetical order. Here everything about the staffs can be updated. From creating a new category or deleting existing category to update name or salary, everything can be done here.

3.Inventory

- Method:
 - `loadData()`: This method loads inventory data from csv files and saves them in a map.
 - `displayInventory()`: This method display all the available items with price under specific category. Here $n*3$, which means in every row 3 category is displayed and under each category the available items are displayed in an intuitive way.

4.ProcessOrder

- Method:
 - `clearcurrentline()`: Replace the line with empty string.
 - `DateOrTime()`: It generates the present time and date.
 - `userinput()`: Here all inputs are taken from the user. It ensures handling edge cases by using low and high number with a suitable message "s".
 - `takeorder()`: In this method, first the availbe items are show by using `displayInventory()` method from `Inventory()` class. Then, three input are taken until 0 pressed, the category no. the serial no. and the quantity (which is by default 1-20). After taking orders, these are saved in a map for future use.
 - `SaveOrder()`: Here all the sold products with quantity and total amount, given and change money is saved. It saves in a

created folder which names is given by the date and the file name is given according to the time.

- Showorder(): Here loadData() and displayInventory() methods are called for inventory usage. Here necessary info are generated for printing the order product and money transactions in the terminal and in the end if any product is ordered then it calls the saveOrder() function by passing the necessary parameters.

5.TransactionRecords

- Method:
 - clearcurrentline(): This Replace the line with empty string.
 - userInput(): Here all the inputs are taken from the user. It ensures handling edge cases by using low and high number with a suitable message “s”.
 - loadDates(): It loads the folder name of Dates and saved in a vector.
 - loadTimeCSV(): This loads the csv file inside the Date folder.
 - showDate(): It shows all the information of the given date and time given in the parameter.
 - saleMenu(): A small login menu that initially show the date folders then can go to inside to choose which time csv file he wants to check.

6.UpdateInventory

- Method:
 - clearcurrentline(): This replaces the current line with empty string.
 - userInput(): Here all the inputs are taken from the user. It ensures handling edge cases by using low and high number With a suitable message “s”;

- `updateFile()`: This updates the csv file according to how the user changes the data.
- `printCategoryitems()`: This displays the item names and prices of a specific category.
- `updateMenu()`: This is the menu where every single information about the inventory can be changed. From deleting the category or creating new category to updating the prices or name.

Main Method

The main method is the entry point of the program where execution begins. It is typically found in `main.cpp` and serves to coordinate the behavior of the entire program. In main method of Shopify Manager, it has two standalone function `clearcurrentline()` and `userinput()` for clearing the current line and handling edge cases of the user inputs. Inside the main method all necessary objects created and instantiated for the final outcome the project. After that a while loop is created, inside it has two while loop one for login credentials checking and the second one is the main menu where the user will do operations according to his needs.

File System

The root folder named Shop Management System. Inside it has 4 major folders.

- Files: It stores all the necessary files for inventory and employees.
- src: It stores all the source files.
- Include: It stores all the header files.
- Sale History: It stores the transaction records

Alongside these folders, cmake-build-debug folder and a CMakeLists.txt file and a .exe file is located here.

Implementation Details

- Technology Used
 - Programming Language: C++, Cmake.
 - Development Enviroment: CLion IDE by JETBRAINS.
- Data structures
 - Vector
 - Map
 - String
- Key Algorithm
 - Authentication: Validations for secure logins.
 - UserInput: Handles edge cases while giving input by user.

System Requirement

- Software Requirement
 - Operating System : DOS
 - Developing software : CLion IDE by JETBRAINS
 - Language Used : C++
- Hardware Requirements
 - Processor : PC with an Pentium II-Class
 - Memory : 256 MB Ram
 - Hard disk space : 2.5 GB installation drive
 - Monitor : 15.6 inch Color Monitor
 - Keyboard : 105 keys
 - Display Type : Super VGA(800X600)
 - Mouse : Serial Mouse

Future Work

Future enhancements for the Shopify Manager could include:

- Graphical User Interface (GUI): Developing a user-friendly GUI using frameworks like Qt or GTK.
- Database Integration: Implementing a database (e.g., SQLite) for persistent storage of employee, inventory, and transaction data.
- Mobile Application: Creating a mobile version of the application for on-the-go access.

Conclusion

The Shopify Manager successfully meets its objectives of improving shop management efficiency. The project demonstrates a solid understanding of C++ programming, data structures, and software design principle. Continuous improvements and potentials enhancements can significantly increase its utility and user satisfaction.

References

- Programming in ANSIC by E. Balagurusamy
- Object Oriented Programming with C++ by E. Balagurusamy