Project for 1st semester of Bachelor of Information Technology

**Invoice Hub**

****

**Ashbin Rai (310882)**

**Pratik Man Shrestha (310901)**

**Sujal Baidhya (310917)**

**KIST College of Information Technology**

**Faculty of Science and Technology**

**Purbanchal University, Nepal**

**February, 2025**

**Student’s Declaration**

We hereby declare that the project report entitled

**“Invoice Hub”**

Submitted in partial fulfillment of the requirements for the degree of Bachelor in Information Technology to Purbanchal University, Biratnagar, Nepal is our original work and not submitted for the award of any other degree, diploma, fellowship or any other similar title or prizes.

**Submitted by:**

Ashbin Rai (310882)

Pratik Man Shrestha (310901)

Sujal Baidhya (310917)

Date: 2025/02/02

**RECOMMENDATION**

This is to certify that this project entitled **Invoice Hub** **prepared and submitted by Ashbin Rai, Pratik Man Shrestha, and Sujal Baidhya,** in partial fulfillment of the degree of Bachelor of Information Technology awarded by Purbanchal University, and has been completed under my supervision. I recommend the same for acceptance by Purbanchal University.

Deepak Khadka, BIT Coordinator

Kist College of Information Technology

09 February, 2025

**CERTIFICATE**

This project entitled **Invoice Hub** prepared and submitted by **Ashbin Rai, Pratik Man Shrestha and Sujal Baidhya** has been examined by us and is accepted for the award of the marks of First Semester Project in BIT by Purvanchal University.

Name: Manoj Kumar Haluwai Signature Date: 2025/02/02

Associate Professor

Deepak Khadka Signature Date: 2025/02/02

Supervisor

Deepak Khadka Signature Date: 2025/02/02

Head of Department

**ACKNOWLEDGEMENT**

We are pleased to present the “**Invoice Hub**” as our 2nd semester project. We would like to express our sincere gratitude to all those who have supported and guided us throughout the duration of this project.

Firstly, we extend our deepest thanks to our **BIT Coordinator Mr. Deepak Khadka** whose expertise, guidance, and encouragement have been invaluable. His insights and feedback have greatly enriched this project. We are extremely grateful to our college, **KIST College of Information Technology** for giving to this opportunity and providing the resources and peaceful environment necessary for this project.

We are also grateful to our colleagues and team members, Ashbin Rai, Pratik Man Shrestha, Sujal Baidhya, for their active inspiration and cooperation. Our dedication and hard work have been instrumental in the successful completion of this project. We also thank **PURBANCHAL UNIVERSITY** for designing such a course structure, where we get to learn and implement new things

Lastly, we would like to thank our family and friends for their unwavering support and encouragement throughout this project.

**ABSTRACT**

This project, Invoice Hub, aims to streamline restaurant operations by automating the creation and management of customer invoices using C++ programming. The system features user authentication, menu management, and secure data handling, significantly enhancing billing accuracy and efficiency. It supports functionalities like updating prices and removing items, ensuring flexibility for restaurant staff. Future enhancements include cloud integration for data backup and remote access, UI improvements, and exploring IoT applications for further optimization. By reducing manual labor and minimizing billing errors, Invoice Hub demonstrates the practical application of C++ programming in solving real-world challenges in the hospitality industry. This project highlights the potential for further innovation in restaurant management systems.

**Keywords:** Invoice Generation, C++ Programming, Restaurant Operations, Automated Billing, Data Security, Future Enhancements.

Contents

[1. **Background** 8](#_Toc173480410)

[1.1 Introduction 8](#_Toc173480411)

[1.2 Objective of the project 8](#_Toc173480412)

[1.3 Features 9](#_Toc173480413)

[1.4 Future Prospects for Invoice Hub 9](#_Toc173480414)

[1.5 Programming Language Used 9](#_Toc173480415)

[2. **System Requirements** 11](#_Toc173480416)

[2.1 Hardware: 11](#_Toc173480417)

[2.2 Software: 11](#_Toc173480418)

[3. **System design** 12](#_Toc173480419)

[3.1 Algorithm 12](#_Toc173480420)

[3.2 Flowchart 19](#_Toc173480421)

[4. **Source Code** 22](#_Toc173480422)

[5. **System Analysis** 23](#_Toc173480423)

[6. **Problems Faced** 25](#_Toc173480424)

[7. **Conclusion** 26](#_Toc173480425)

[8. **References** 27](#_Toc173480426)

# 1. Background

## Introduction

Invoice Hub is a user-friendly invoicing system made for restaurants. It helps managers create, save, and manage invoices easily. With features like user sign-up, login, password recovery, and detailed invoice creation, Invoice Hub makes the billing process simple and accurate. This tool is designed to improve the efficiency of restaurant operations and enhance customer satisfaction. Additionally, Invoice Hub offers menu management, invoice search, and secure data handling, ensuring a comprehensive solution for all billing needs.

## 1.2 Objective of the project

The Primary Objective of this Project are as follows:

1. **Create Invoices Easily**: Simplify the process of generating bills for restaurant transactions.
2. **Handle Menu Updates**: Facilitate easy updates to the restaurant menu, including adding, updating, or removing items.
3. **Keep Data Safe**: Implement robust security measures to safeguard user and invoice data.
4. **Find Invoices Quickly**: Enable quick search and retrieval of past invoices for efficient record management.
5. **Easy to Use**: Design a system that's straightforward for restaurant staff to navigate and use.

## 1.3 Features

The major features of Invoice Hub are as following:

1. Create detailed bills for restaurant orders.
2. Easily manage menu items and their prices.
3. Search and display invoices of customer.
4. Securely handle user logins and data.
5. Simple interface with customization options.

## 1.4 Future Prospects for Invoice Hub

1. **Sales Analysis**: Get detailed reports on what sells best and when.
2. **Mobile App**: Make a version that works on phones and tablets
3. **Stock Control**: Help keep track of what's in stock and when to order more.
4. **Language Options**: Add more languages for different customers and staff.

## 1.5 Programming Language Used

C++ programming is an advanced language developed in the 1980s by Bjarne Stroustrup. It is known for its flexibility and support for both low-level and high-level programming. It builds upon C by adding object-oriented features like classes and inheritance, making it more versatile and scalable. C++ is widely used in system programming, game development, and applications requiring high performance due to its efficiency and control over system resources. Its strong emphasis on modularity and reusability has influenced many modern programming languages used today.

Main Concepts of C++ programming used for Invoice Hub are:

1. Functions
2. Array
3. Class & Objects
4. File Handling
5. Branching
6. Looping
7. String

# 2. System Requirements

## 2.1 Hardware:

1. 2GB RAM or more.
2. Color Monitor (LCD, LED)
3. Intel i3 or more
4. 64 GB storage.

## 2.2 Software:

1. Operating System: Windows (Windows XP, Windows 7, Windows 8, Windows 10), MAC, Linux

# 3. System design

## 3.1 Algorithm

**A. Signup Menu**

Step 1: Start

Step 2: Check If one admin already exists or not.

Step 3: Take Name, Phone Number, NEW Username, Password, Double Check Password.

Step 4: Displays entered password as ‘\*’.

Step 5: Re-enter password.

Step 6: Check whether the inputted password matches the re-entered password.

1. If yes, create an account.
2. If not, ask the user to check the password.

Step 7: If the user wants to add a onetime code.

1. If yes, enter the code.
2. If not, goto next step.

Step 8: Encrypt all the data in a file.

Step 9: Goto login page after successful signup.

Step 10: End

**B. Login Menu**

Step 1: Start

Step 2: Take Username and Password.

Step 3. Displays entered password as ‘\*’.

Step 4: Open file and check whether the inputted username and password are correct or not.

Step 5: Enter both username and password

1. Given username and password should match with the username and password given inside the file.
2. If yes, goto next step.
3. If not, give the user three tries to enter correct data.
4. If yes, Enter the one-time code.
5. If not, goto step 2.

Step 6: End

**C. Generate Invoice (Invoice Menu)**

Step 1: Start

Step 2: Initialize a file pointer to store the data.

Step 3: Open Menu file to Display the Menu.

Step 4: Input the customer’s details (Name, Order’s name and Quantity).

Step 5: Display the Bill.

Step 6: Ask If the invoice is to be saved or not.

1. If yes, encrypt and save the data.
2. If not, don't save the data.

Step 7: End

**D. View All Saved Invoices (Invoice Menu)**

Step 1: Start

Step 2: Display all the recorded invoices

Step 3: Retrieve The recorded data from the particular file

Step 4: Display all the Record Until EOF

Step 5: End

**E. Search and View Specific Invoice (Invoice Menu)**

Step 1: Start

Step 2: Input The name of the Invoice that needs to be displayed.

Step 3: Check whether that record is on the file or not.

1. If yes, goto step 4.
2. If not, username not found.

Step 4: Display the recorded invoices.

Step 5: Retrieve The particular recorded data from the particular file.

Step 6: Display the record Until EOF.

Step 7: End

**F. Add Item (Change Menu Section)**

Step 1: Start

Step 2: Open the menu file using a pointer.

Step 3: Input the Name and Price of the Item.

Step 4: Write the Given input into the menu file.

Step 5: Ask If the user wants to add another file or not.

1. If yes, goto step 2.
2. If not, goto next step.

Step 7: End

**G. Remove Item (Change Menu Section)**

Step 1: Start

Step 2: Open menu file and temp menu file using pointer.

Step 3: Input the Name of the Item.

Step 4: Check If the item is on the file or not.

Step 5: If an item is on the file, move every other item to the new temp menu file.

Step 6: Remove the original menu file and then rename the temp menu file to the original name.

Step 7: Ask if the user wants to remove more items.

1. If yes, goto step 2.
2. If not, goto next step.

Step 8: End

**H. Change Price of The Item (Change Menu Section)**

Step 1: Start

Step 2: Input the name of the Item.

Step 3: Open menu file and temp menu file.

Step 4: Check if the item is on the file or not.

Step 5: If an item is on the file, move every other item to the temp file.

Step 6: Input new price of item and move it to temp menu file.

Step 7: Ask if the user wants to change the price of more items.

1. If yes, goto step 2.
2. If not, goto next step.

Step 8: End

**I. View Current Menu (Change Menu Section)**

Step 1: Start

Step 2: Display all the recorded items.

Step 3: Retrieve The recorded items from the particular file.

Step 4: Display all the Records Until EOF.

Step 3: End

**J. Remove Admin**

Step 1: Start

Step 2: Input the username of the admin.

Step 3: Input The given String.

Step 4: Ask If Admin wants to delete all the data present in the current files or not.

Step 5: If yes remove all the files along with User, else remove user only.

Step 6: End

**K. Change Admin’s Password**

Step 1: Start

Step 2: Input Verifier Code.

Step 3: Check whether the code and The Username is Same or not.

1. If yes, enter the new password.
2. If not, Display “you can't change the password”.

Step 4:  Save all the modified and unmodified encrypted data into the file of the user.

Step 5: End

**L. Logout**

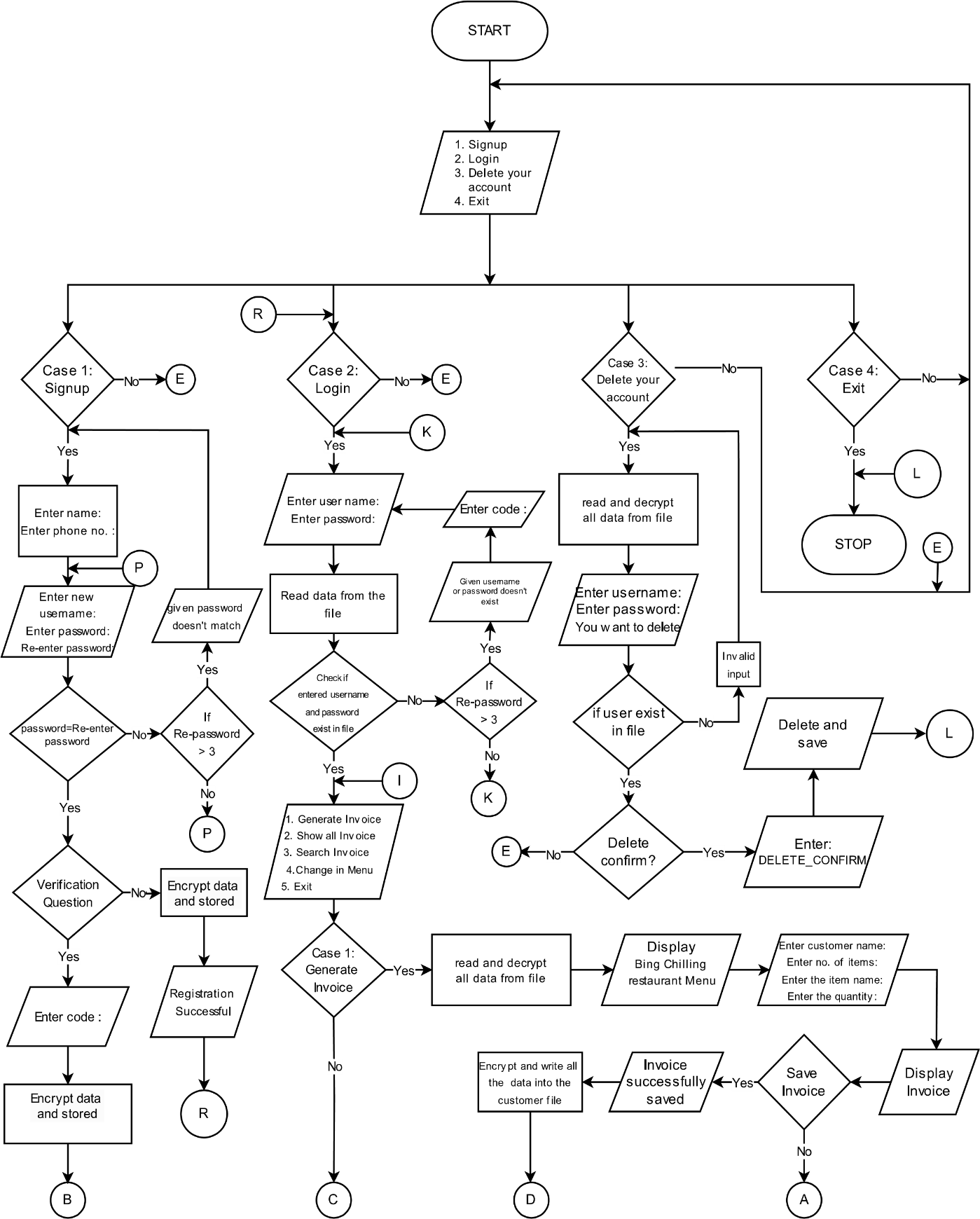
Step 1: Start

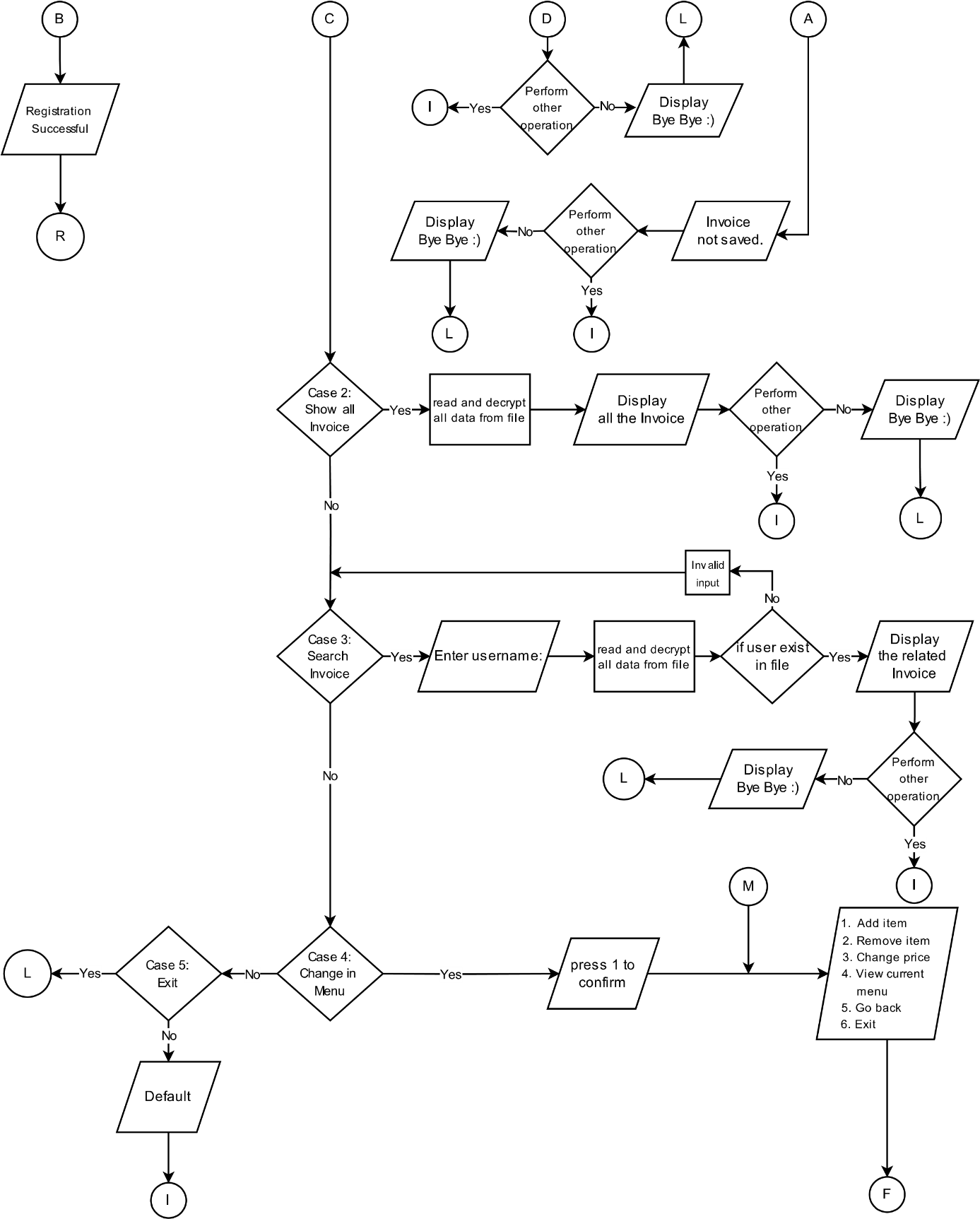
Step 2: If the user presses 4(Exit).

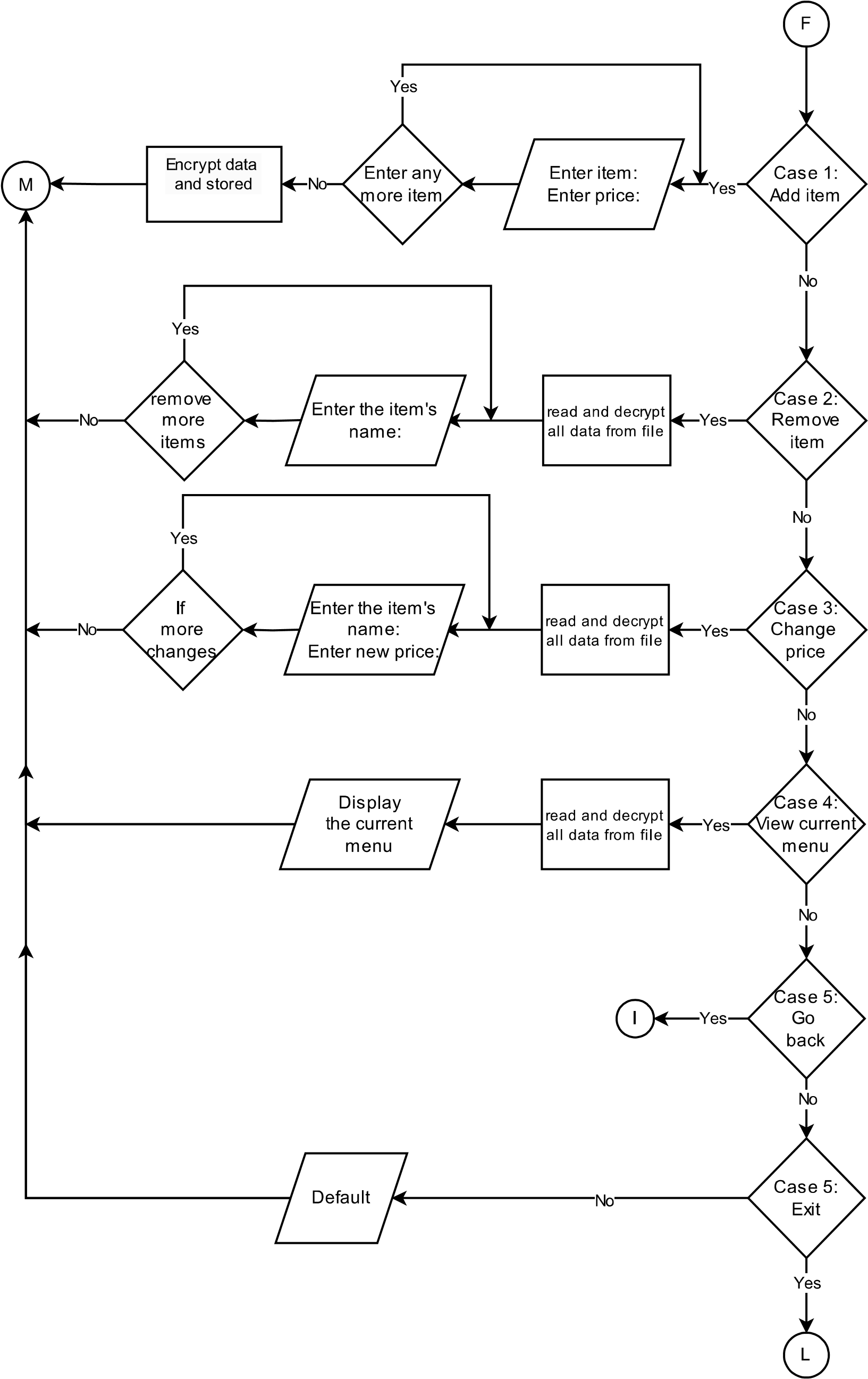
1. If yes, stop the program.
2. If not, ignore and continue with the user page.

Step 3: Stop

## 3.2 Flowchart







# 4. Source Code

https://github.com/Pratik167/Project\_2/blob/main/Second%20Sem%20Project/inv.cpp

# 5. System Analysis

 **User Authentication:**

* **Sign-Up:** New users can easily create an account to get started.
* **Login:** Existing users can securely log in to access their accounts.

 **Menu Management:**

* **Add Items:** Staff can quickly add new items to the menu, specifying names and prices.
* **Update Items:** Prices and details of menu items can be updated effortlessly.
* **Remove Items:** Outdated or unavailable items can be removed from the menu with ease.

 **Invoice Generation:**

* **Record Orders:** Customer orders are recorded and invoices are generated automatically.
* **Store Invoices:** Invoices are saved for future reference and easy retrieval.

 **Data Security:**

* **Password Encryption:** User passwords are securely encrypted for safety.
* **Secure Data Handling:** All customer and menu data is handled securely to prevent unauthorized access.

 **User Interface:**

* **Ease of Use:** The system is designed to be intuitive and user-friendly, making it easy for restaurant staff to navigate and use.
* **Flexibility:** Staff can efficiently manage menu items and process invoices without hassle.

 **Error Handling:**

* **Accurate Billing:** Automating invoice generation reduces billing errors.
* **Data Validation:** Ensures that all input data is accurate and reliable.

 **Future Enhancements:**

* **Cloud Integration:** Plans to add cloud features for data backup and remote access.
* **UI Improvements:** Ongoing efforts to refine the user interface for an even better user experience.
* **IoT Integration:** Exploring ways to incorporate IoT technology for further optimization of restaurant operations.

# 6. Problems Faced

**1. Compatibility Issues:**

* Different versions of our development environment caused some features to work perfectly in one version but fail in another, making it hard to get everything running smoothly.

**2. Debugging Challenges:**

* Tracking down and fixing bugs was tougher than expected. Each feature, from generating invoices to managing the menu, had its own set of issues that required significant time and effort to resolve.

**3. User Authentication:**

* Setting up a secure system for user login and password encryption was challenging. Ensuring only authorized users could access the system required careful planning and implementation.

**4. User Interface Design:**

* Creating a user-friendly interface for restaurant staff took several iterations. Balancing functionality with simplicity required multiple redesigns based on feedback.

**5.** **Testing and Validation:**

* We had to test the system thoroughly to ensure all features worked correctly in different scenarios. This involved a lot of trial and error to make sure we hadn’t missed anything.

.

# 7. Conclusion

Our Invoice system program in C successfully creates a simple and efficient way to handle billing, reducing mistakes and making the process faster. Key features include easy bill generation, a login system, and a user-friendly interface. We've tested the system thoroughly and added encryption to keep data safe. In the future, we plan to add more features, improve the design, make it scalable, and compatible with variety of businesses. This project has taught us a lot and shown that our billing system can greatly improve how billing is done.

# 8. References

ADV. LEARNING, YouTube, (Jul. 05, 2021) Password encryption

Retrieved from https://www.youtube.com/watch?v=xtk14TcWaN4&amp;t=1s (accessed Jun 13, 2024).

S.FOX( Apr. 27, 2014  ) delete a structure from a binary file

Retrieve from https://stackoverflow.com/questions/23323723/how-do-i-delete-a-structure-from-a-binary-file (accessed Jun. 18, 2024).

R.VERMA(Nov. 30,2017) Nested structure

Retrieve from https://www.quora.com/What-is-a-nested-structure-and-explain-with-examples-with-syntax (accessed Jun. 15, 2024).