**Abstract:**

This project presents the development of an Online Food Ordering System using C++. The system offers a user-friendly interface, enabling customers to browse menus, select dishes and customize their orders. It incorporates key functionality such as user registration. The system also includes error-handling mechanisms to provide reliable and robust performance, ensuring smooth operation even in exceptional scenarios.

**Introduction:**

**Background**

The food industry has witnessed a significant transformation in recent years, driven by advancements in technology and changing consumer preferences. Online food ordering platforms have gained immense popularity, offering convenience and a wide range of options to customers. With the increasing reliance on digital platforms for various services, there is a growing need for a robust and efficient Online Food Ordering System that can cater to the demands of modern consumers, restaurants, and delivery personnel.

**Need for the Project**

Traditional methods of food ordering, such as phone calls or in-person visits, are gradually becoming outdated due to their limitations in terms of time consumption, limited menu options, and the inconvenience of payment handling. Recognizing these challenges, an Online Food Ordering System becomes essential to bridge the gap between customers, restaurants, and delivery personnel, ensuring a seamless and efficient process for all stakeholders involved.

The primary objective of this project is to develop a comprehensive Online Food Ordering System using C++, offering a user-friendly interface, secure payment processing, and efficient data management. This system aims to address the evolving needs of customers who seek convenience, restaurants that aim to expand their customer reach, and delivery personnel who require a streamlined process for managing and delivering orders.

**System Features**

The proposed Online Food Ordering System will incorporate several key features to provide a holistic and satisfying experience for its users. Some of the prominent features include:

**a) User Registration:**

Customers will be able to register on the platform.

**b) Menu Browsing and Customization:**

The system will provide a menu from the restaurant, allowing customers to browse, select dishes, and customize their orders according to their preferences.

**c) Secure Payment Processing:**

The system will integrate secure payment gateways to enable online payments, ensuring the confidentiality and integrity of customers' financial information.

**Modules:**

The program includes the <iostream> header, which is a standard C++ library for input and output operations and <fstream> header, which is a standard C++ library and it provides classes and functions for file handling, allowing to read from and write to files. It includes classes such as ifstream (input file stream), ofstream (output file stream), and fstream (file stream for both input and output).

**Language:**

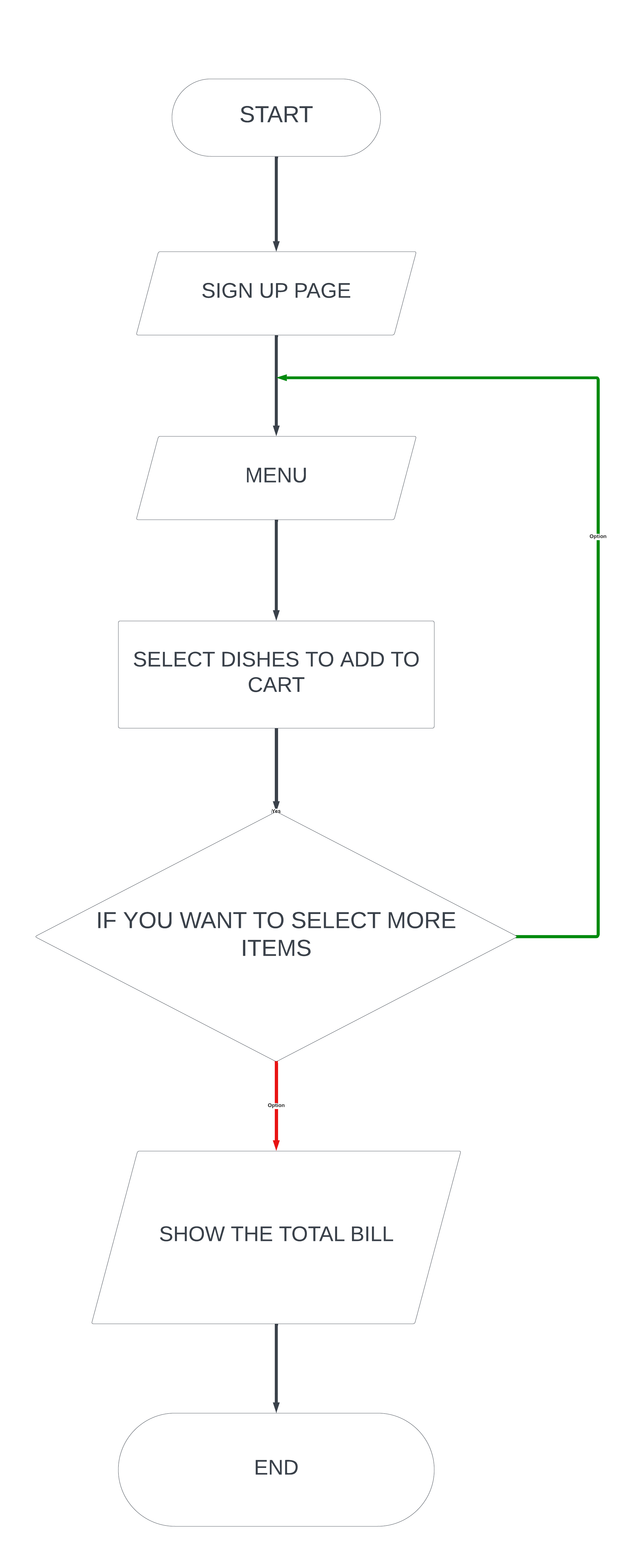
The code is written in C++, as indicated by the inclusion of the <iostream> header and the usage of C++ syntax, such as cout and cin.

**Techniques:**

* **File Handling:** The code utilizes the ofstream class to handle file output operations. It writes data to a file named "items.txt" using the fout object.
* **Control Flow**: The code employs control flow constructs such as if statements and while loops to control the execution flow based on certain conditions.
* **Functions:** The code uses function declarations to define several functions (burger(), fries(), etc.), although their implementations are missing. Functions are used to modularize the code and perform specific actions.
* **User Input:** The code accepts user input through the cin statement, allowing users to select food items and quantities.

**Procedure/Methodology:**

* The program includes necessary header files, defines functions, and declares variables.
* The main() function is defined.
* Inside the main() function, a file named "items.txt" is opened in append mode for storing the order details.
* Various variables and pointers are declared to store user input and calculate the total bill.
* A do-while loop is used to display the menu and accept user choices until the user chooses to exit.
* Inside the loop there is a Signup option that prompts the user to enter their desired Gmail and password. Check if the entered Gmail already exists in the file. If yes, prompt the user to enter a new Gmail. Write the entered Gmail and password to the file for registration.
* Similarly, there is a login option that prompts the user to enter their Gmail. Search for the entered Gmail in the file. If found, prompt the user to enter the corresponding password. Compare the entered password with the stored password. If they match, log in successfully; otherwise, display an incorrect password message.
* Then, the user is prompted to select an option from the menu.
* The option\_checker() function is called to validate the user's choice.
* Based on the user's choice, a corresponding function is called to handle the specific food item.
* Each food item function prompts the user for additional choices (e.g., quantity) and calculates the cost of the selected items.
* The total cost for each food item is accumulated in variables (e.g., b\_sum, f\_sum) and stored in an array (array[0] to array[7]).
* After processing the food item, the user is prompted to select another item or view the total bill.
* If the user chooses to view the total bill (by entering 't' or 'T'), the total bill is calculated by summing up the accumulated costs.
* The total bill is displayed, and the order details (individual food item costs and total) are stored in the "items.txt" file.
* If the user chooses to continue ordering (by entering 'y' or 'Y'), the loop continues.
* If the user chooses to exit (by entering 'e' or 'E'), the loop breaks, and the program ends.
* Dynamic memory allocated using the new operator is deallocated using the delete operator.
* The program ends with a return statement.



**FALSE**

**TRUE**

**FLOWCHART:**

**Conclusion:**

In conclusion, the development of the Online Food Ordering System using C++ has provided valuable insights into the modernization of the food industry and the significance of technology in meeting the evolving demands of customers, restaurants, and delivery personnel. Through this project, several key learnings have been derived. This project aims to fulfill the growing need for a modern Online Food Ordering System that offers convenience, accessibility, and efficiency to customers, restaurants, and delivery personnel.

**Future Enhancements for the Online Food Ordering System:**

Future work on this Online Food Ordering System can focus on several areas of improvement. Firstly, integrating artificial intelligence and machine learning algorithms can enhance the system's recommendation capabilities, suggesting personalized food options to customers based on their preferences and past orders. This can further improve customer engagement and satisfaction.

Secondly, expanding the system to include additional features such as customer reviews and ratings, social media integration, and promotional offers can enhance customer engagement and boost restaurant participation. These additions can provide valuable feedback to both customers and restaurants, promoting transparency and building trust within the system.

Furthermore, incorporating a delivery optimization algorithm can improve the efficiency of order deliveries, reducing delivery time and optimizing routes for the delivery personnel. This can lead to increased customer satisfaction and cost savings for the delivery operations.

**REFERENCES:**

* Dynamic memory allocation example (from lab course material)
* https://itsourcecode.com/free-projects/cplusplus-projects/food-ordering-system-project-in-c-with-source-code/?expand\_article=1