Georgios Vassiou

Project C++
Airport Terminal
Guide through
Intermediate course

Georgios Vassiou/ 2024

Getting Started
IDE & Features explained

IDE that I’m going to use in that project is Xcode Version 15.4 and operating system going to be MacOs Sonoma 14.5.

This is a simple C++ project an Airport Terminal that let the user to 1. Sign up 2. Log in 3. Purchase a ticket 4. Ticket info 5. Log out

Georgios Vassiou/ 2024

Project C++
Files & Class

logo.cpp main.cpp Management.h

I’m using three files for this project. Main.cpp is the main program file that includes my int main().

Management.h is my class header file that includes my class.

And last but not least :P my logo.cpp file that includes all the functions written for my program.

*Keep in mind that at the end of this project we are going to have also one more text file users.txt that is going to include our users that have signedUp into the program.

Georgios Vassiou/ 2024

Project C++
#Include

I’m going to explain here all the #Includes of my program.

#include <iostream>: Provides functionality for input and output operations (e.g., std::cin, std::cout).

#include <vector>: Provides the std::vector container, which is used to store collections of elements. (This helps with tickets)

#include "Management.h": This directive includes the Management.h header file, allowing the source file to use the declarations and definitions contained within it.

#include <fstream>: Provides functionality for file input and output operations (e.g., std::ifstream, std::ofstream). As mentioned before at the end of this guide through a users.txt will appear. This header is for that job.

#include <sstream>: Provides functionality for string stream operations, which can be used to manipulate strings as streams. Is used to convert the cost of a ticket from an integer to a string for display purposes

#include <iomanip>: Provides input and output manipulators (e.g., std::setw, std::setfill) for formatting output.

Georgios Vassiou/ 2024

Project C++
Airport Terminal Logo

```
void Management::logo()
{
    cout << "  __          __   _                                  _ " << endl
              << "  \\ \\        / /__| | ___ ___  _ __ ___    ___  | |" << endl
              << "   \\ \\ /\\ / / _ \\ |/ __/ _ \\| '_ ` _ \\ / _ \\ | |" << endl
              << "    \\ V  V /  __/ | (_| (_) | | | | | |  __/ |_|" << endl
              << "     \\_/\\_/ \\___|_|\\___\\___/|_| |_| |_|\\___| (_)" << endl;

    cout << "\n\t" << setw(45) << setfill('~') << "\n"
              << "\n\t          TO AIRPORT TERMINAL          " << endl;
    cout << "\n\t" << setw(45) << setfill('~') << "\n";
}
```

Void logo() function

Georgios Vassiou/ 2024

First thing we want the user to see is our logo. So lets start with that.

I'm using ASCII art because I think it looks nice inside the terminal when the program is running.

A welcome message and the logo of the company afterwards.

Once my files is created and linked properly into my IDE I'm creating a function called void logo(). Inside this function I simply use std::cout to print my logo.

If you noticed my function is a bit different as I mention above.

This happens because I made a class called Management and linked this function to it. All of my functions are linked into that class.

Also I used a little bit of formatting with setw(45) & setfill('~').

Project C++
userChoice() function

```
int Management::userChoice()
```

userChoice() function

Next up we are going to need a menu for our users to choose an option.

So in that case int userChoice() function is being created.

Here basically is going to be every step that our user will see and interact.

Created bool loggedIn = false; This variable is used when the program starts the loggedIn statement is false by default. This variable works in background user can't see it.

Also int choice; variable this will be what user chooses to proceed inside our program.

```
bool loggedIn = false;
int choice;
do {
    cout << "1. Sign Up\n2. Login\n3. Purchase Ticket\n4. View Ticket Info\n5. Exit\nEnter choice: ";
    cin >> choice;
```

Variables inside of int userChoice()

Georgios Vassiou/ 2024

Project C++
userChoice() function

Inside int userChoice() function added switch case {} statement so our menu to be created. Every case calls a different function (). I will explain them more detailed in the next sliders. At the end of my switch {} default: segment works like a protection for the program if does not enter the right value provided. In that case an integer number.

So my menu is now ready but how can I make it that runs into a loop?

I used do while{} and every time user inputs the right value will run again my menu. As the value isn't going to be 5 number this segment will run forever.

And at the end of every choice userChoice() function will return choice; into int main().

Georgios Vassiou/ 2024

```
switch (choice) {
    case 1:
        //Κλήση της συνάρτησης εγγραφής
        signUp();
        break;
    case 2:
        //Κλήση της συνάρτησης σύνδεσης
        login();
        if (loggedInUser.username != "") {
        //Εάν ο χρήστης συνδεθεί με επιτυχία, ενημερώστε το loggedInUser
            loggedIn = true;
        }
        break;
    case 3:
        //Έλεγχος για την επιλογή 3 πρέπει πρώτα να συνδεθεί ο χρήστης
        if (!loggedIn) {
            cout << "Please login first.\n";
        } else {
            //Κλήση συνάρτησης αγοράς εισιτηρίου
            purchaseTicket();
        }
        break;
    case 4:
        //Έλεγχος για πληροφορίες εισιτηρίου πρέπει πρώτα να συνδεθεί ο χρήστης
        if (!loggedIn) {
            cout << "Please login first.\n";
        } else {
            cout << "Enter ticket number: ";
            string ticketNumber;
            cin >> ticketNumber;
            //Κλήση συνάρτησης προβολής πληροφοριών εισιτηρίου
            viewTicketInfo(ticketNumber);
        }
        break;
    case 5:
        if (loggedIn) {
            cout << "Logging out...\n";
            //Αποσύνδεση χρήστη
            loggedInUser.username = "";
            //Ορισμός μεταβλητής σε ψευδή
            loggedIn = false;
        }
        cout << "Exiting...\n";
        break;
    default:
        cout << "Invalid choice. Please try again.\n";
        break;
}
}while (choice != 5);
return choice;
```

Switch {} statement & do while {}

Project C++
userChoice() function

```
if (cin.fail()) {

//Εκκαθάριση της κατάστασης σφάλματος
                cin.clear();

//Αγνόηση του υπόλοιπου buffer εισόδου
                cin.ignore(numeric_limits<streamsize>::max(), '\n')
                cout << "Invalid input. Please enter a number.\n";
                continue;
            }
```

cin.fail()

Inside int userChoice() after our menu was up and running a major problem appeared.

When user wasn't entering the right value to make a choice program was entering the infinite loop this function overloading ram and creating input lag.

Searching through the internet reading from Stack overflow forum posts I came out with one solution that seems to be working. Inside of a logical statement to check cin.fail() returns true in situations such as:

Type Mismatch: When the user input cannot be converted to the required type. For example, entering a non-numeric string when an integer is expected.

End of File (EOF): When the end of the input stream is reached unexpectedly.

Stream Errors: Other input-related errors that may occur.

cin >> choice;: Attempts to read an integer from the input and store it in the variable choice.

cin.fail(): Checks if the input operation succeeded.

If the user enters a non-numeric input, such as a string, cin.fail() will return true.

If the input is valid, cin.fail() will return false.

Georgios Vassiou/ 2024

Project C++
signUp() function

void Management::signUp()

signUp() function

The signUp() function is used to register a new user. It collects user information such as username, password, email, and phone number, checks if the username already exists, and if not, saves the user's information to a file.

- The usernameExists function is called with newUser.username as an argument to check if the username is already taken.
- If the username exists, a message is displayed, and the function returns early, preventing further execution. (I will explain this function detailed later.

A. The file users.txt is opened in append mode (ios::app). This means that new data will be added to the end of the file without overwriting existing data.

B. If the file cannot be opened, an error message is displayed, and the function returns early.

C. At the end file.close(); closes the file to ensure that all data is properly saved.

Georgios Vassiou/ 2024

```
    //Συνάρτηση εγγραφής νέου χρήστη
void Management::signUp() {

    //Δημιουργία ενό νέου χρήστη
    User newUser;
    cout << "\nEnter username: ";

    //Εισαγωγή ονόματος χρήστη
    cin >> newUser.username;

    //Έλεγχος αν το όνομα χρήστη υπάρχει ήδη
    if (usernameExists(newUser.username)) {
        cout << "\nUsername already exists. Please choose a different one.\n" << endl;
        return;
    }

    //Εισαγωγή κωδικού πρόσβασης
    cout << "\nEnter password: ";
    cin >> newUser.password;

    //Εισαγωγή email
    cout << "\nEnter email: ";
    cin >> newUser.email;

    //Εισαγωγή αριθμού τηλεφώνου
    cout << "\nEnter phone nubmer:";
    cin >> newUser.phone;

    //Άνοιγμα το αρχείο για εγγραφή πληροφοριών
    ofstream file("users.txt", ios::app);

    //Αν το αρχείο δέν είναι ανοιχτό τότε εκτύπωνει σφάλμα
    if (!file.is_open()) {
        cout << "\nError: Unable to open file.\n";
        return;
    }

    //Εγγραφή πληροφοριών
    file << "Username: " << newUser.username << "\n";
    file << "Password: " << newUser.password << "\n";
    file << "Email: " << newUser.email << "\n\n";
    file << "Phone: " << newUser.phone << "\n\n";

    //Κλείσιμο αρχείου για αποθήκευση και αποφυγή σφάλματος
    file.close();

    //Εφόσον κλείσει το αρχείο εκτύπωση μηνύματος Εγγραφή επιτυχής
    cout << "Sign up successful.\n";
}
```

Project C++
Quick note inside Management.h

I’m going to make a quick explanation inside of my class header file here.

The Management.h file is a header file that contains declarations of structures, classes and functions. This is included on all of my source files that end up with “.cpp” to allow them to access these declarations.

Struct User{} helps us with signUp() function defines a structure to hold user-related data. Username, password etc.

Is a simple data structure that groups together related variables. This makes it easier to handle user data throughout my program. I’m using the same almost method for tickets. But I will explain it later after the purchaseTicket() function. So lets return() to usernameExists() function :P

```
struct User {
    std::string username;
    std::string password;
    std::string email;
    std::string phone;
};
```

Struct user{} declaration inside Management.h file

Georgios Vassiou/ 2024

Project C++ usernameExists() function

```
    //Συνάρτηση για τον έλεγχο αν το όνομα χρήστη υπάρχει ήδη
bool Management::usernameExists(const std::string& username) {

    //Άνοιγμα του αρχείου χρηστών
    ifstream file("users.txt");
    string line;

    //Διάβασμα του αρχείου γραμμή προς γραμμή
    while (getline(file, line)) {

    //Έλεγχος αν η γραμμή περιέχει το όνομα χρήστη
        if (line.find("Username: " + username) != string::npos)

    //Κλείσιμο αρχείου
            file.close();

    //Το όνομα αρχείου αν το βρεί επιστρέφει αληθής
            return true;
        }
    }
    //Κλείσιμο αρχείου
    file.close();
    //Αν δεν υπάρχει το όνομα χρήστη επιστρέφει το Ψευδής
    return false;
```

Georgios Vassiou/ 2024

If stream file(“users.txt”); opens this file for reading purposes. String line; Declares a std::string variable to hold each line read from the file.

While { } getline(file, line) Reads the file line by line until the end of the file is reached.

If { } line.find(“Username: “ + username) != string::npos checks if the current line contains the specified username. If found closes the file and returns “True”.

If { } not found then proceeds to file.close(); closes the file after the loop ends and return “False”.

Project C++
login() function

Let’s get a little bit complicated :D

The fun part begins now! A new function declaration begins as usual with two string variables username and password. Here I’m going to store what user going to input and save it in users.txt file.

If { } !file.is_open() attempt to open a file with a name users.txt if the file can’t be opened it std::cout a message to the user “Error Unable to open file.”

Once our program opened the file for *reading it reads line by line in a loop it checks if the line contains the entered username using find(“Username: +username) if found continues if to find the password.

If the password matches it sets the Authenticated to true and exits the loop. After that closes the file *Remember always to close the file because you will run into unexpected problems… :/

```
void Management::login() {

    //Δημιουργία μεταβλητών string
    string username, password;

    //Εισαγωγή Ονόματος Χρήστη
    cout << "Enter username: ";
    cin >> username;

    //Εισαγωγή Κωδικού χρήστη
    cout << "Enter password: ";
    cin >> password;

    //Άνοιγμα αρχείου χρηστών
    ifstream file("users.txt");

    //Αν το αρχείο δέν είναι ανοιχτό τότε εκτύπωνει σφάλμα
    if (!file.is_open()) {
        cout << "Error: Unable to open file.\n";
        return;
    }

    string line;

    //Μεταβλητή για να βρεί αν ο χρήστης ταυτοποιήθηκε
    bool isAuthenticated = false;


    while (getline(file, line)) {

    //Εύρεση του ονόματος χρήστη στο αρχείο χρηστών
        if (line.find("Username: " + username) != string::npos) {
            getline(file, line);

    //Διαβάζει γραμμή προς γραμμή για να βρεί τον κωδικό πρόσβασης
            if (line.find("Password: " + password) != string::npos) {

    //Εφόσον τον βρεί ο χρήστης παίρνει τιμή αληθής
                isAuthenticated = true;
                break;
            }
        }
    }
    //Κλείσιμο αρχείου χρηστών
    file.close();

    // Άν αληθής εκτύπωση μηνύματος επιτυχή εισόδου
        if (isAuthenticated) {
            cout << "Login successful.\n";
            loggedInUser.username = username; // Ανανέωση του loggedInUser
            cout << "Logged in as: " << loggedInUser.username << endl; // Τυπώνουμε το username για έλεγχο
        }
    // Αλλιώς εκτύπωση μηνύματος σφάλματος
        else {
            cout << "Invalid username or password.\n";
```

Georgios Vassiou/ 2024

Project C++
login() function

```
    if (isAuthenticated) {
        cout << "Login successful.\n";
        loggedInUser.username = username; // Ανανέωση του loggedInUser
        cout << "Logged in as: " << loggedInUser.username << endl; // Τυπώνουμε το username για έλεγχο
    }
// Αλλιώς εκτύπωση μηνύματος σφάλματος
    else {
```

Last but not least after everything goes as planned xD Authenticated is set to true and user can proceed further to the program and purchase an air-ticket.

This function is crucial to the program handles file operations, string searching user input/output and ensuring that only valid users can login in by verifying their credentials against the stored data in users.txt file.

Georgios Vassiou/ 2024

Project C++
purchaseTicket() function

This is another switch case {} statement and the last in that program. This function displays a list of destination with their corresponding costs. Option for the user to choose among (1, 2 & 3) with the variable destinationChoice. If the choice isn't valid it prints "Invalid choice".

```
    //Συνάρτηση εισιτηρίων
void Management::purchaseTicket() {

    Ticket newTicket;
    //Εμφάνιση επιλογών προορισμού
    std::cout << "Select destination:\n";
    std::cout << "1. Athens (70 euros)\n";
    std::cout << "2. Thessaloniki (50 euros)\n";
    std::cout << "3. Patra (40 euros)\n";
    std::cout << "\nChoose here:";

    int destinationChoice;
    std::cin >> destinationChoice;

    switch (destinationChoice) {
        case 1:
            newTicket.destination = "Athens";
            newTicket.cost = 70;
            break;
        case 2:
            newTicket.destination = "Thessaloniki"
            newTicket.cost = 50;
            break;
        case 3:
            newTicket.destination = "Patra";
            newTicket.cost = 40;
            break;
        default:
            std::cout << "Invalid choice.\n";
            return;
    }
```

purchaseTicket() function

Georgios Vassiou/ 2024

Project C++ purchaseTicket()function

Program asks the user if wants a round trip. Option for the user (1 for true or 0 for false). This part here is crucial for this program as it stores that into newTicket variable.

Quick note here to explain what I mean “crucial”. All this information we want to store it somewhere and then pass it to the main.cpp program.

This is going to Management.h header file I will explain later more detailed. As I noticed at the beginning #include section <vector> template needed here to create dynamic array storage and manipulate data.

In that case what does this really mean for my program? Generate a new Ticket because we may have multiple users inside users.txt file.

```
//Ερώτηση για μετ επιστροφής;
std::cout << "Round trip? (1 for yes, 0 for no): ";
std::cin >> newTicket.roundTrip;

//Εισαγωγή βάρους έξτρα αποσκευών
std::cout << "Luggage weight (kg): ";
std::cin >> newTicket.luggageWeight;

//Υπολογισμό κόστους
if (newTicket.roundTrip) {
//Διπλασιασμός κόστους για μετ επιστροφής εισιτηρίου
    newTicket.cost *= 2;
}

//Δημιουργία αριθμού εισιτηρίου
newTicket.ticketNumber = "T" + std::to_string(tickets.size() + 1);

//Ανάθεση ονόματος χρήστη
newTicket.username = loggedInUser.username;

//Καταχώρηση εισιτηρίου στο vector
tickets.push_back(newTicket);

//Εκτύπωση απόδειξης
printReceipt(newTicket);
```

Part inside purchaseTicket() function

Georgios Vassiou/ 2024

Project C++ purchaseTicket()function

Here is more detailed as mentioned before the structures variables for Ticket in my Class file.

Also the template std::vector<Ticket> tickets;

As mentioned previously.

```
struct Ticket {
    std::string ticketNumber;
    std::string username;
    std::string destination;
    int cost;
    bool roundTrip;
    int luggageWeight;
};

class Management {
private:

    //Λίστα για τα εισιτήρια
    std::vector<Ticket> tickets;

    //Χρήστης που έχει συνδεθεί
    User loggedInUser;

    //Συναρτήσεις εισιτηρίων private
    void printReceipt(const Ticket& ticket);
    void purchaseTicket();
    void viewTicketInfo(const std::string& ticketNumber);
    bool usernameExists(const std::string& username);
```

Photo of Management.h header file

Georgios Vassiou/ 2024

Project C++
Void printReceipt()

This function prints a receipt for my program. Some key information about the ticket. Ticket Number, Destination, Round trip or not and Luggage Weight. This function takes the previous values that user chose on purchaseTicket() function.

```
void Management::printReceipt(const Ticket& ticket) {

    //Εκτύπωση απόδειξης
    cout << "\n-------- Legal Receipt --------\n";
    cout << "Ticket Number: " << ticket.ticketNumber << endl;
    cout << "Destination: " << ticket.destination << endl;
    cout << "Round Trip: " << (ticket.roundTrip ? "Yes" : "No") << endl;
    cout << "Luggage Weight: " << ticket.luggageWeight << " kg" << endl;

    // Μετατροπή του κόστους από int σε string
    std::ostringstream costStream;
    costStream << ticket.cost;
    std::string costString = costStream.str();

    // Εκτύπωση του κόστους ως συμβολοσειρά
    cout << "Cost: " << costString << " euros" << endl;

    cout << "\nThank you for your purchase!\n";
    cout << "\n-------- Legal Receipt Ends ! --------\n";
}
```

Georgios Vassiou/ 2024

Project C++
Void printReceipt()

Std::ostringstream is provided by the C++ library. Used by <sstream> header. Declared a variable costStream to allow it to insert data.

Ticket.cost is assumed to be an integer variable. With the operator << I'm allowing it to store the value of ticket.cost on costStream object. Internally to store this value as a sequence of characters.

The str() method extracts this string representation from the ostringstream. Finally I save it into another variable called costString.

```
cout << "Round Trip: " << (ticket.roundTrip ? "Yes" : "No") << endl;
cout << "Luggage Weight: " << ticket.luggageWeight << " kg" << endl;

// Μετατροπή του κόστους από int σε string
std::ostringstream costStream;
costStream << ticket.cost;
std::string costString = costStream.str();
```

Georgios Vassiou/ 2024

Project C++
Void viewTicketInfo()

This function iterates through the tickets collection to find a ticket with the ticketNumber. If a matching ticket is found it prints the ticket's information Ticket Number, Destination, Round Trip etc and exits the function.

If not matching ticket is found after checking all tickets prints a message Ticket not found.

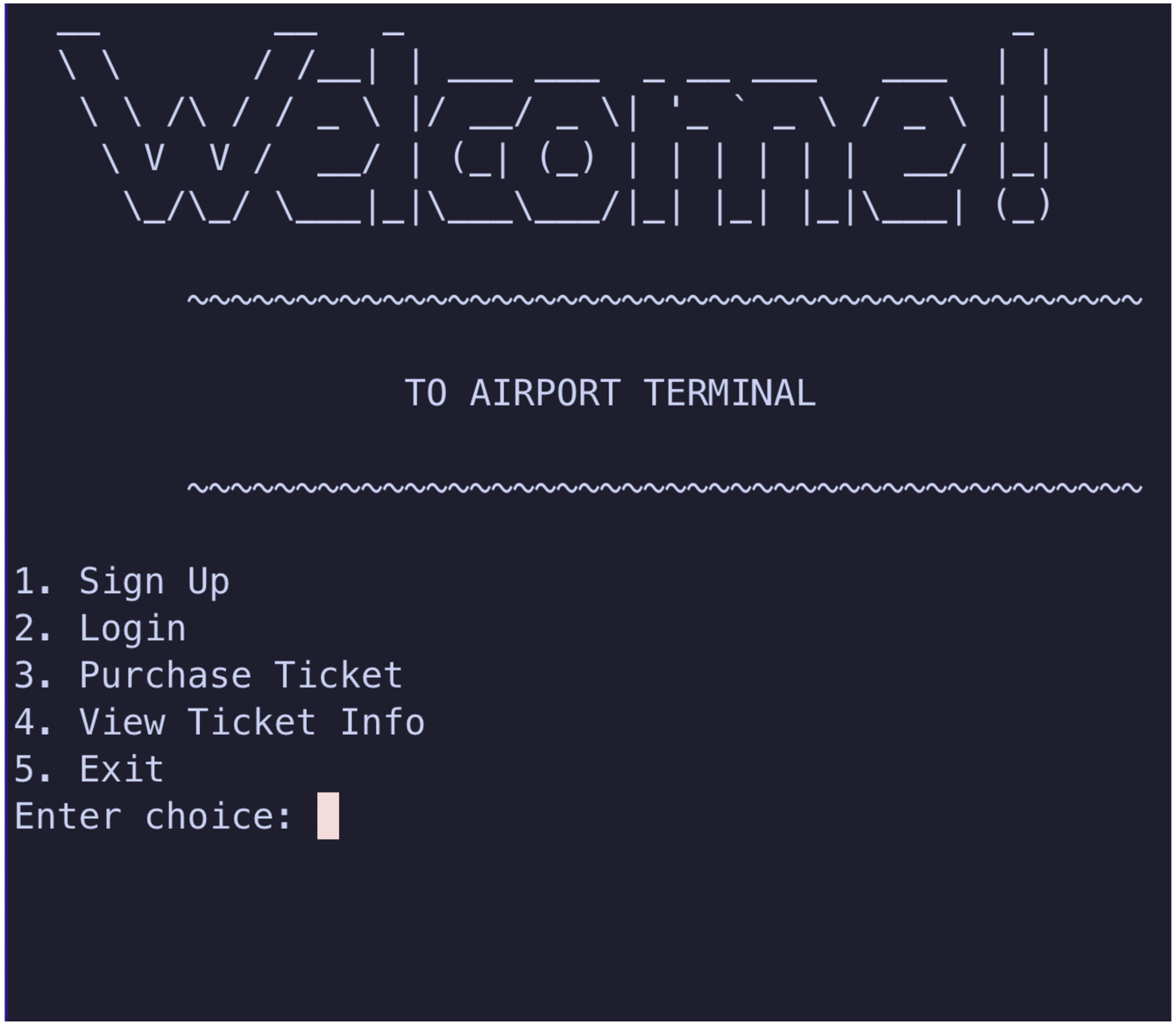
*Quick not about for() loop. Const& auto ticket declares ticket as a constant reference to avoid copying each ticket object.

```
void Management::viewTicketInfo(const std::string& ticketNumber) {

    //Επαλήθευση του αριθμού εισητηρίου στη λίστα των εισιτηρίων
    for (const auto& ticket : tickets) {
            if (ticket.ticketNumber == ticketNumber) {
    //Άν βρεί εισιτήριο εκτυπώνει τις πληροφορίες παρακάτω
                cout << "\nTicket Information\n";
                cout << "Ticket Number: " << ticket.ticketNumber << endl;
                cout << "Destination: " << ticket.destination << endl;
                cout << "Round Trip: " << (ticket.roundTrip ? "Yes" : "No") << endl;
                cout << "Luggage Weight: " << ticket.luggageWeight << " kg" << endl;
                return;
            }
        }
    //Άν ο αριθμός του εισιτηρίου δεν βρεθεί εκτύπωση δεν βρέθηκε εισιτήριο
        cout << "Ticket not found.\n";
}
```

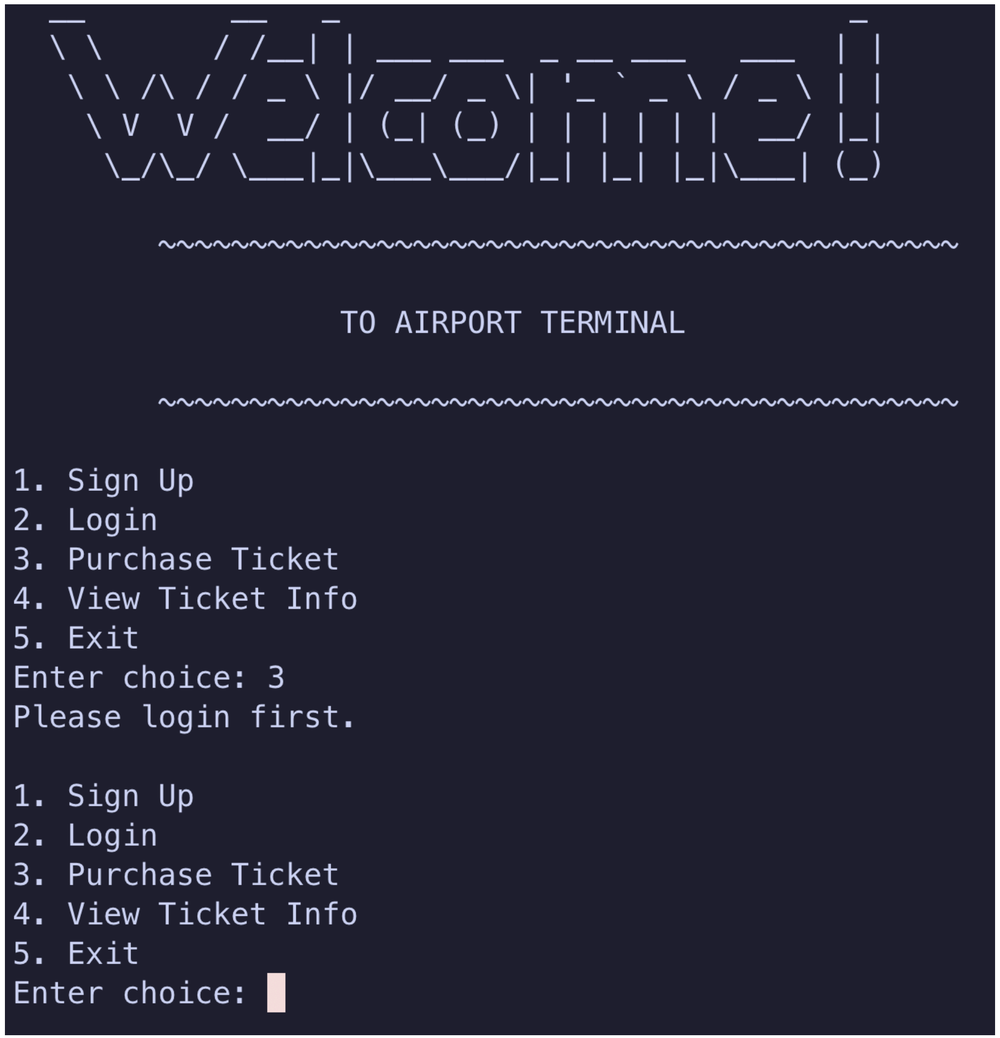
Georgios Vassiou/ 2024

Project C++
Program
Running



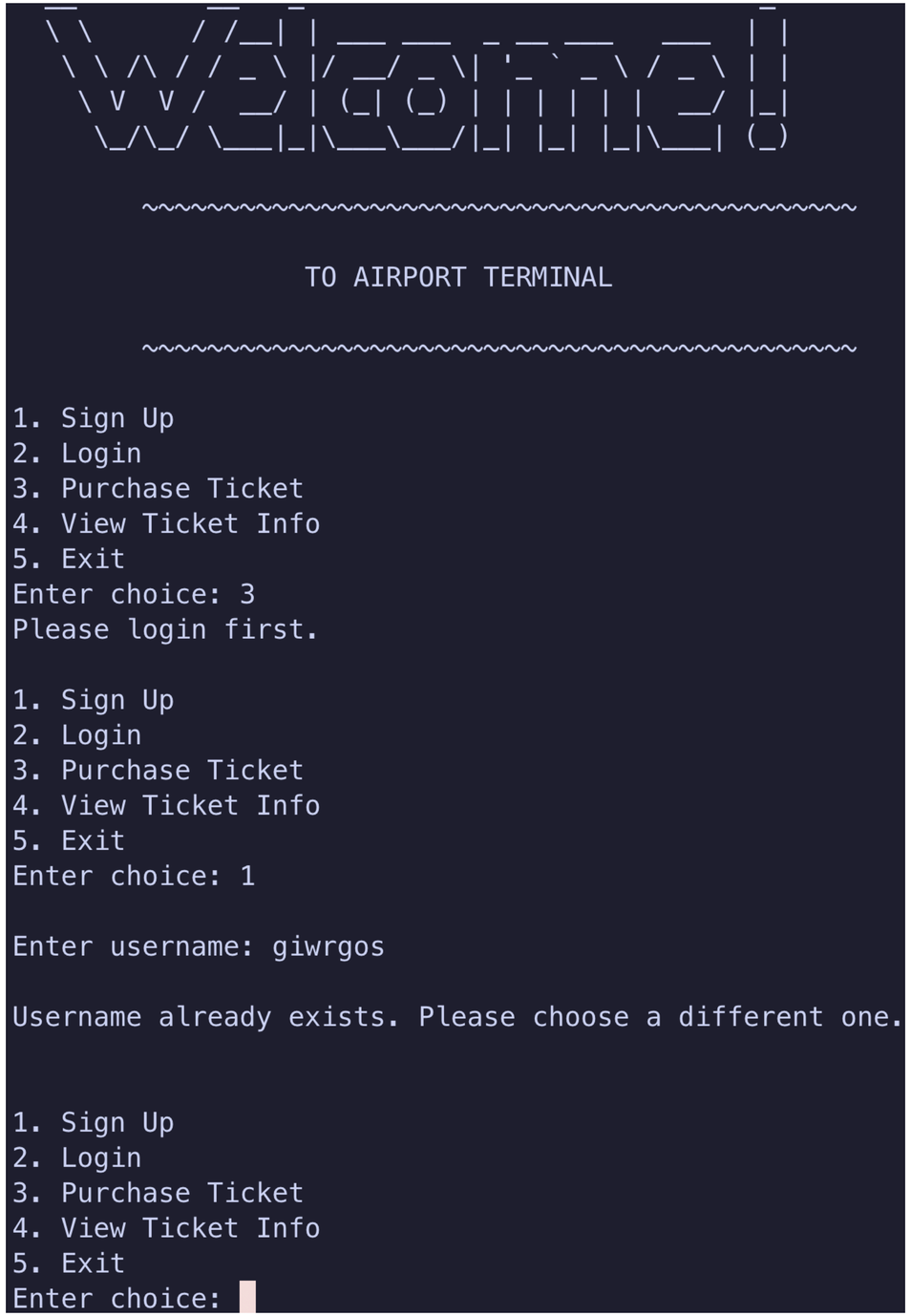
Georgios Vassiou/ 2024

Project C++
Program Running



Georgios Vassiou/ 2024

Project C++
Airport Terminal



Georgios Vassiou/ 2024

Project C++
Airport Terminal

```
1. Sign Up
2. Login
3. Purchase Ticket
4. View Ticket Info
5. Exit
Enter choice: g
Invalid input. Please enter a number.

1. Sign Up
2. Login
3. Purchase Ticket
4. View Ticket Info
5. Exit
Enter choice: 1

Enter username: Georgios

Enter password: 12345

Enter email: example1@gmail.com

Enter phone nubmer:6932471289
Sign up successful.

1. Sign Up
2. Login
3. Purchase Ticket
4. View Ticket Info
5. Exit
Enter choice:
```

Georgios Vassiou/ 2024

Project C++
Airport Terminal

```
Enter username: Georgios
Enter password: 12345

==================================

Login successful.

Logged in as: Georgios

==================================

1. Sign Up
2. Login
3. Purchase Ticket
4. View Ticket Info
5. Exit
Enter choice: 3
Select destination:
1. Athens (70 euros)
2. Thessaloniki (50 euros)
3. Patra (40 euros)

Choose here:2
Round trip? (1 for yes, 0 for no): 1
Luggage weight (kg): 28

-------- Legal Receipt --------
Ticket Number: T1
Destination: Thessaloniki
Round Trip: Yes
Luggage Weight: 28 kg
Cost: 100 euros

Thank you for your purchase!

-------- Legal Receipt Ends ! --------

1. Sign Up
2. Login
3. Purchase Ticket
4. View Ticket Info
5. Exit
Enter choice:
```

Georgios Vassiou/ 2024

Project C++
Airport Terminal

```
1. Sign Up
2. Login
3. Purchase Ticket
4. View Ticket Info
5. Exit
Enter choice: 4
Enter ticket number: T1

Ticket Information
Ticket Number: T1
Destination: Thessaloniki
Round Trip: Yes
Luggage Weight: 28 kg
```

Georgios Vassiou/ 2024

Project C++
Airport Terminal

This is the end. If you made it till the end a big thank you for your patience :) This is my first project. I hope that I did a decent work stay tuned for more. :D

THANK YOU !

Georgios Vassiou/ 2024