DSA LAB PROJECT

Bus Terminal Management System

(with seats booking)

Submitted To:

Sir Usman Ahmed

Submitted By:

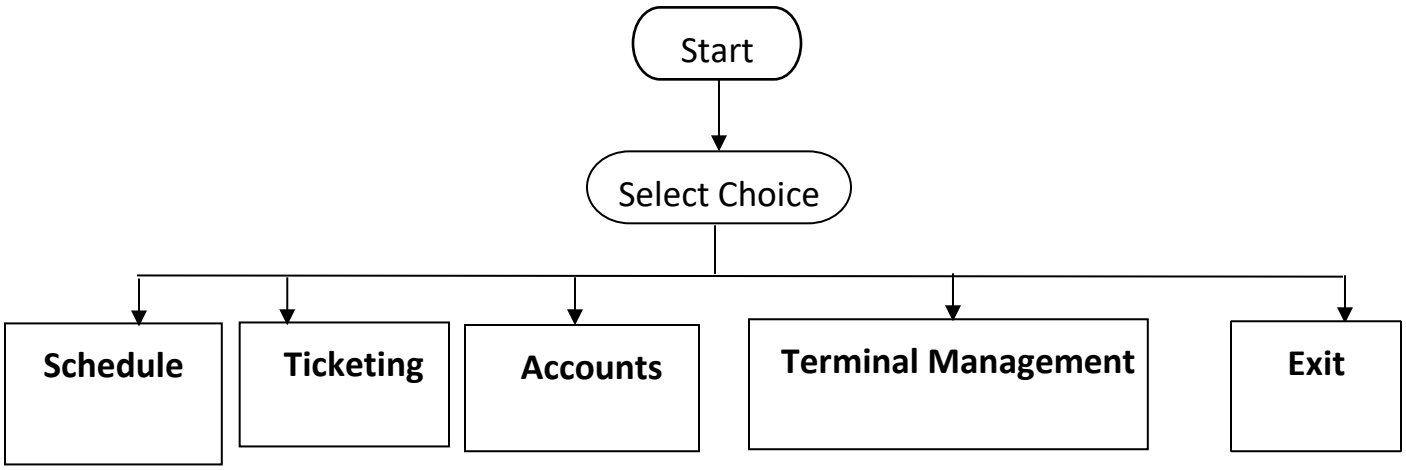
Usama Jalal **2020-CS-665**

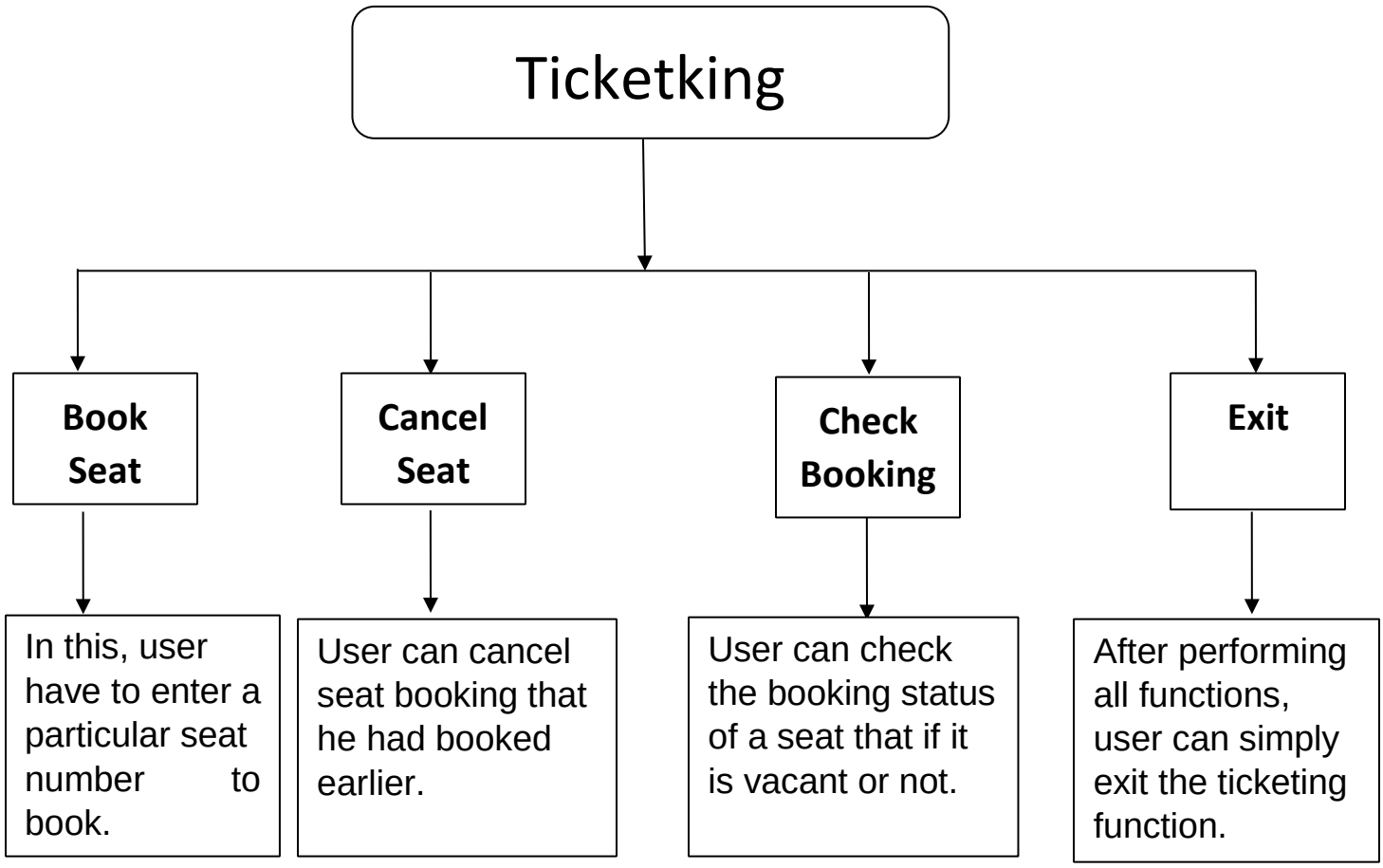
Muhammad Fawad **2020-CS-692**

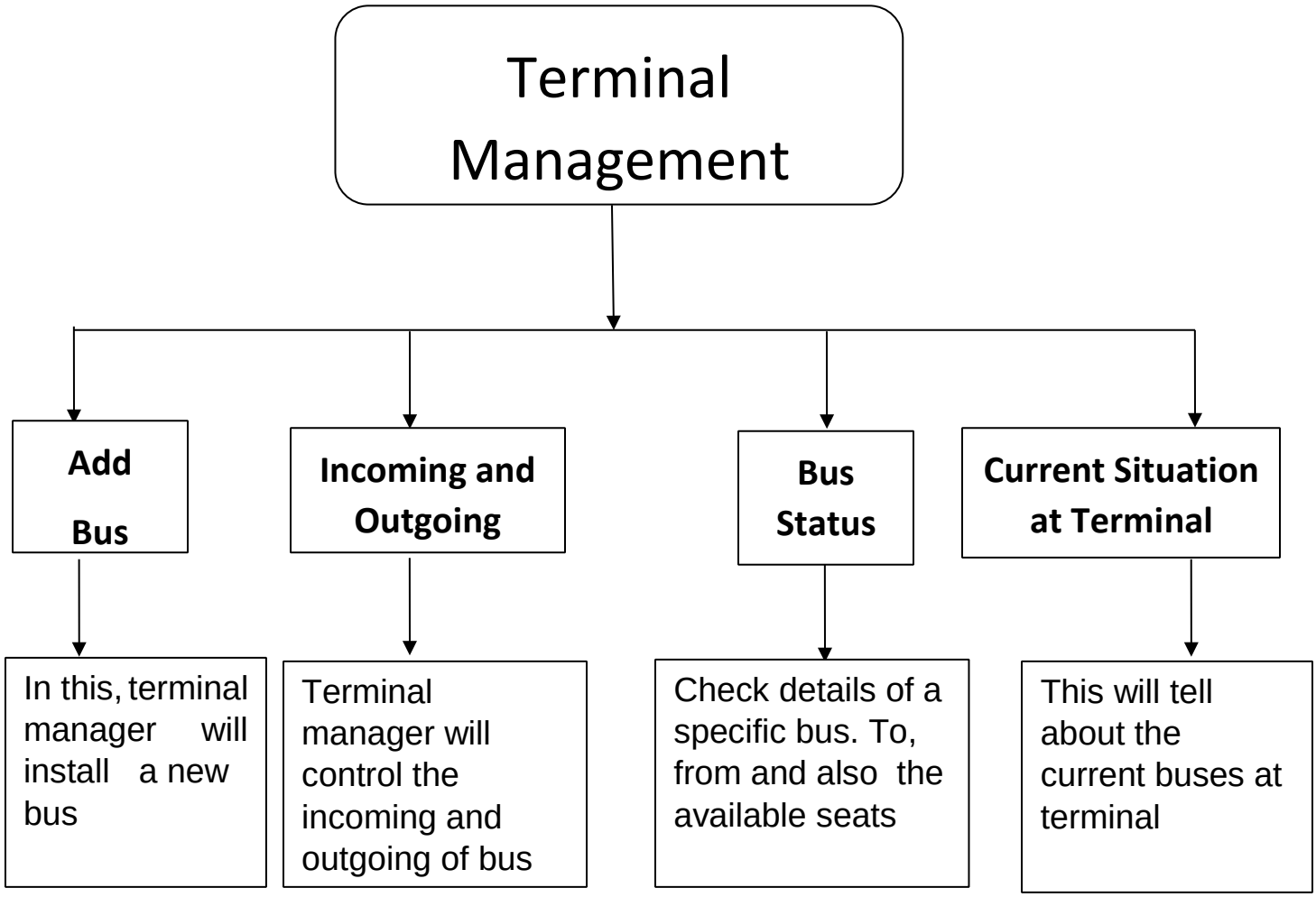
Department of Computer Science

University of Engineering and Technology, Lahore.

Flow Chart:


Start
Select Choice
Schedule
Ticketing
Accounts
Terminal Management
Exit


Ticketking
Book Seat
Cancel Seat
Check Booking
Exit
In this, user have to enter a particular seat number to book.
User can cancel seat booking that he had booked earlier.
User can check the booking status of a seat that if it is vacant or not.
After performing all functions, user can simply exit the ticketing function.


Terminal Management
Add Bus
Incoming and Outgoing
Bus Status
Current Situation at Terminal
In this, terminal manager will install a new bus
Terminal manager will control the incoming and outgoing of bus
Check details of a specific bus. To, from and also the available seats
This will tell about the current buses at terminal

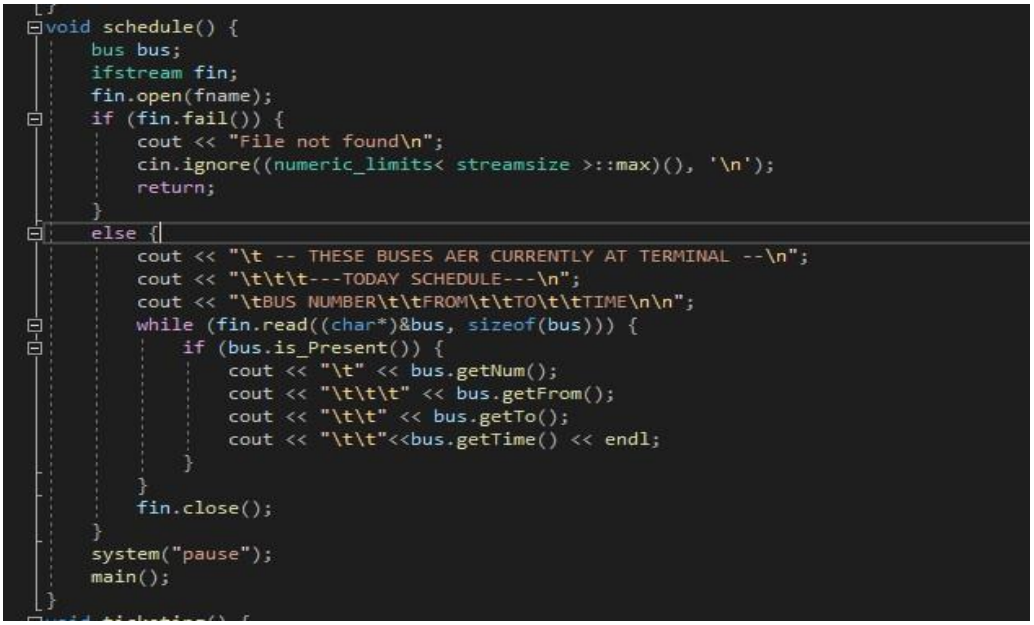
USER MANAUAL

Project is based on bus terminal management, contains schedule for a day of buses, tickets, bus seat allocation. Project is processed by header files of manager, ticket, bus, bus seat. In which functions are performed according to program functionality. These header files are added in a main .cpp file to program execution. At the start of program, user receives a menu displayed on screen such as:

1. Check Schedule
2. Ticketing
3. Accounts
4. Terminal Management
0. Exit

1. Check Schedule

When user enters in schedule menu, a list of buses is displayed on the screen indicating the buses that are available at the terminal.



```
void schedule() {
    bus bus;
    ifstream fin;
    fin.open(fname);
    if (fin.fail()) {
        cout << "File not found\n";
        cin.ignore((numeric_limits< streamsize >::max)(), '\n');
        return;
    }
    else {
        cout << "\t -- THESE BUSES AER CURRENTLY AT TERMINAL --\n";
        cout << "\t\t\t---TODAY SCHEDULE---\n";
        cout << "\tBUS NUMBER\t\tFROM\t\tTO\t\tTIME\n\n";
        while (fin.read((char*)&bus, sizeof(bus))) {
            if (bus.is_Present()) {
                cout << "\t" << bus.getNum();
                cout << "\t\t\t" << bus.getFrom();
                cout << "\t\t" << bus.getTo();
                cout << "\t\t"<<bus.getTime() << endl;
            }
        }
        fin.close();
    }
    system("pause");
    main();
}
```

2. Ticketing

In the ticketing management, there pop up a menu to book a ticket for a particular seat, cancel a ticket at any time, checking the status for a particular seat.

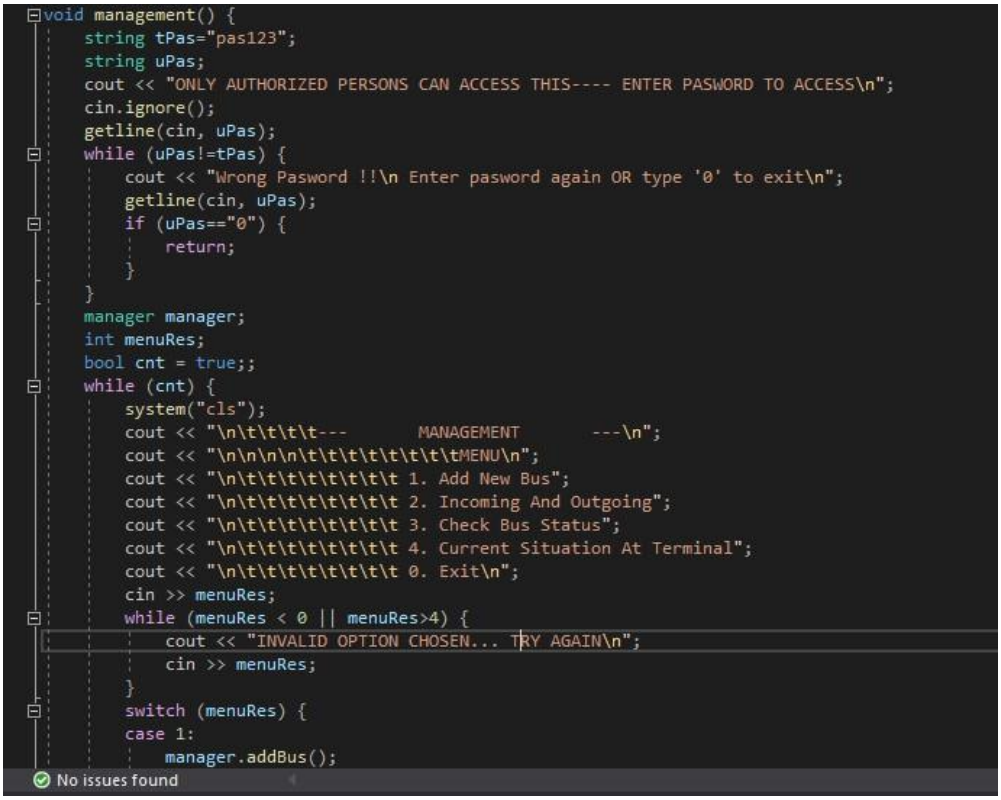
```
void ticketing() {
    ticket t;
    int menuRes;
    bool cnt = true;;
    while (cnt) {
        system("cls");
        cout << "\n\t\t\t\t---       MANAGEMENT       ---\n";
        cout << "\n\n\n\n\t\t\t\t\t\t\t\tMENU\n";
        cout << "\n\t\t\t\t\t\t\t\t 1. Book Seat";
        cout << "\n\t\t\t\t\t\t\t\t 2. Cancel Seat";
        cout << "\n\t\t\t\t\t\t\t\t 3. Check Booking";
        cout << "\n\t\t\t\t\t\t\t\t 0. Exit\n";
        cin >> menuRes;
        while (menuRes < 0 || menuRes>3) {
            cout << "INVALID OPTION CHOSEN... TRY AGAIN\n";
            cin >> menuRes;
        }
        switch (menuRes) {
        case 1:
            t.bookSeat();
            break;
        case 2:
            t.cancelSeat();
            break;
        case 3:
            t.showAll();
            break;
        case 0:
            cnt = false;
            main();
        }
    }
    system("pause");
}
```

3. Accounts

In accounts section, it is maintained that an account that is being used by customer to book a ticket having a valid balance in it and display balance of account.

4. Terminal Management

Terminal management is basically only for management people i.e., Manager. First, we must check that person trying to enter in terminal management have the access to enter by asking him the special password. When a particular person enters in terminal management than we a menu is shown in front of him, that contains options for adding a new bus in terminal, incoming and outgoing buses from the terminal, check the status of any bus in terminal, and status of any bus at terminal.



```
void management() {
    string tPas="pas123";
    string uPas;
    cout << "ONLY AUTHORIZED PERSONS CAN ACCESS THIS---- ENTER PASWORD TO ACCESS\n";
    cin.ignore();
    getline(cin, uPas);
    while (uPas!=tPas) {
        cout << "Wrong Pasword !!\n Enter pasword again OR type '0' to exit\n";
        getline(cin, uPas);
        if (uPas=="0") {
            return;
        }
    }
    manager manager;
    int menuRes;
    bool cnt = true;;
    while (cnt) {
        system("cls");
        cout << "\n\t\t\t\t---       MANAGEMENT       ---\n";
        cout << "\n\n\n\n\t\t\t\t\t\t\t\t\tMENU\n";
        cout << "\n\t\t\t\t\t\t\t\t 1. Add New Bus";
        cout << "\n\t\t\t\t\t\t\t\t 2. Incoming And Outgoing";
        cout << "\n\t\t\t\t\t\t\t\t 3. Check Bus Status";
        cout << "\n\t\t\t\t\t\t\t\t 4. Current Situation At Terminal";
        cout << "\n\t\t\t\t\t\t\t\t 0. Exit\n";
        cin >> menuRes;
        while (menuRes < 0 || menuRes>4) {
            cout << "INVALID OPTION CHOSEN... TRY AGAIN\n";
            cin >> menuRes;
        }
        switch (menuRes) {
        case 1:
            manager.addBus();
```

No issues found

0. Exit

After performing a particular function in bus terminal, a user exits the program by exit function.