

Documentation of Final Project

Basics of Programming 2

Project Name: Bus Company Management and Booking System

Submitted by: Das Anjan Kumar

Neptun Code:D42DQA

Supervised by: Mr. AL-Magsoosi Husam Kareem Farhan

Date of Submission: 9th May 2021

Chapter 1: Problem statement and the task description

In this project, I tried to implement a Bus company management and ticket booking system, which can be used both by the owner or manager of the bus company and also the passengers. The managers will decide about the bus information, whereas the passengers can book ticket for those buses.

The program is built using C++ language with object oriented approach. Functionality of the program is to help a Bus company to maintain it's management system and booking system. Only an admin can give input about information of buses and modify bus information. A passenger can see bus information ,book seats in a bus and search previously made booking.

As mentioned before, the program is built with object oriented approach(1).Dynamic memory handling(2) is used in handling an attribute for phone number of a class named passenger. File handling(3) is used in many functions of the projects, because this kind of project is all about storing data. And, lastly exception handling(4) is used in handling the error while giving wrong information while logging in.

I have used inheritance in two of my classes and used multiple inheritance in one class. I have used polymorphism to create a Login function. I have used operator overloading <<, to give output of attributes of a bus object.

Chapter 2: Solution design

Pseudocode:

- 1.Start:
- 2.Choose an option
 - A. Customer entry without login
 - B.Customer entry with login
 - C.Admin login
3. If option "A":
Choose an option:
 - A1.Search a bus
 - A2.Make a booking
 - A3: Search a booking
4. if option "B":
Choose an option:
 - B1: Login using username and password ,then
 - B11.Search a bus
 - B12.Make a booking
 - B13: Search a booking
 - B2: If, don't have a account, Create account and then
 - B21.Search a bus
 - B22.Make a booking

B23: Search a booking

5.If option “C”:

Enter admin user name and password

Now choose an option:

C1:Add new bus information

C2:Show bus information

C3:Remove bus information

6.End.

Description of Coding:

1. Build a class for creating bus objects with needed attributes.
2. Define methods for this class and implement them in a .cpp file,
In our case there are methods for this class, such as, for adding a new bus information, for displaying bus information and for removing a bus information.
3. Define another class for passengers, which will carry attributes and methods related to a passenger.
4. Here, we have used methods for booking a ticket, searching a bus information and searching for a previously made booking.
5. Another class for admin was made, which mainly ensures that the changing and modifying of data is only handled by an admin by the usage of specific user name and password.

Chapter 3: Solution

Description of Classes(with attributes and methods):

Class1: New_Bus {};

It is used to create or adding information about a bus. It has attributes like bus id, driver name, start location,end location,start time, end time,total seats,remaining seats,seat price.

Methods:

a. void add_bus();

This method take input from admin and create a bus information, with file handling the information are stored.

b. void show_businfo();

This method take input of a destination and show the bus information if a bus is available to that destination

c. void remove_businfo();

d. This method removes a bus information from file by taking it's id as input

e. virtual void Login() {};

This method is used with polymorphism with another two class.

Class2: Passenger{};

It has attributes like passenger's name, phone no. , no. of booked seats and also the same attributes as bus class, but for a specific bus.

Methods:

a. void new_booking();

It takes information from the passenger and create a new booking for the passenger according to selected bus of the passenger.

b. void find_booking();

It takes passenger's phone no. and displays a previously made booking

c. void Login();

For a previous passenger, it takes username and password and continues. But for a new passenger, it takes all information from the passenger and create a new account for the passenger.

d.void passenger_menu();

This menu is made for only the option a passenger can see, means the methods previously mentioned.

Class3: admin{};

It only has attributes of username and password .

a.void Login();

It takes admin user name and password and gives access to admin menu.

b. void admin_menu();

It gives access to an admin, to such options, which can be modified

Composition of the solution:

The program has 3 files. One for the main function, name main.cpp, a header file name BUS_PROJECT.h, which has all the class declaration along with their attributes and methods declared. And lastly a cpp file named BUS_PROJECT.cpp , which includes implementation of all the methods.

Chapter 4 : Changes from the "Project Application Form"

There is almost everything in the project, that I mentioned in the form. But in the project I implemented something extra. In the form, I just wrote that this will be booking system, but later in the project I added an admin option for the bus company managers to add information about buses, so now apart from booking, it is also an bus management system.

Chapter 5: Testing

By doing the following steps, the test of the program was done:

1. Firstly, checking the admin login works by inputting the correct username and password for the admin.
2. After admin login, 10 bus with information were added , each of them was searched and also deleted a bus data. So all three functionality for the admin is working.
3. Then, for passengers, firstly without login was tried.
4. It gave all three functionalities for the passengers, firstly searched some destination for the passengers.
5. Then, created 5 bookings for different passengers with information.
6. Then checked all the booking with search booking option, that proves that the passenger file is handled perfectly.
7. Also, the login and creating new account for passengers were checked with new passenger information.
8. Login with existing username and password was done, so file handling for passenger login information were also handled correctly.

So, from the above checking, we can say that the testing of the program was done correctly and the program is doing what it should do.