Project Report: Airline Booking System

Fournier Pierre-Antoine

Contents

1 Requirements Analysis

The objective was to develop an airline booking system to manage flight information, reservations, and ticket refunds. The system includes:

- Input and management of flight data.
- Real-time updates of ticket availability upon booking or cancellation.
- Queries by flight ID, departure time, and terminal station.
- Ticket booking and refund capabilities.

2 Detailed Module Design

The project is modularized into clearly defined components:

2.1 Module: Airline (`airline.h`, `airline.cpp`)

- Stores and manages basic flight data (flight ID, departure and arrival stations, departure time, available seats).
- Functions for loading flight data from files.

2.2 Module: Booking (`booking.h`, `booking.cpp`)

- Manages ticket booking and refund operations.
- Updates available seats after each transaction.

2.3 Module: Query (`query.h`, `query.cpp`)

- Supports queries by flight ID, departure time, or terminal station.
- Provides detailed information about flights matching query parameters.

2.4 Main Application (`main.cpp`)

- Integrates all modules and provides a user-friendly console interface.
- Keeps the application active for multiple operations until explicitly terminated by the user.

2.5 Data Storage

Flight information stored persistently in text files (`flights`).

3 Program Test Analysis and Operation Result Display

The program was thoroughly tested for:

- **Flight information accuracy:** Verified correct loading and displaying of flights.
- **Real-time updates:** Validated seat availability updates after bookings and refunds.
- **Query functionality:** Checked accuracy and robustness of flight searches.
- **Booking and refund process:** Confirmed correct operation and data integrity during multiple bookings/refunds.

All tests passed successfully, demonstrating robustness and reliability.

4 User Manual (Optional)

4.1

Quick Usage Guide:

Run the Program:

```
make
./airline.out
```

Available Commands:

- Enter flight information.
- Browse all flight data.
- Query flights by ID, departure time, or terminal.
- Book tickets.
- Refund tickets.
- Exit program with a specific command (e.g., "exit").

File Structure:

- Flight data stored in the file `flights`.

5 Conclusion

The Airline Booking System meets all specified requirements with clear modular design, robust functionality, and ease of use, making it suitable for real-world applications.

6 Appendix

```
> make
g++ -Wall -Wextra -g -c main.cpp -o main.o
g++ -Wall -Wextra -g -c airlines/airline.cpp -o airlines/airline.o
g++ -Wall -Wextra -g -c airlines/booking.cpp -o airlines/booking.o
g++ -Wall -Wextra -g -c query/query.cpp -o query/query.o
g++ -o airline.out -g main.o airlines/airline.o airlines/booking.o query/query.o
0 | main | ~/Documents/Github/Airport_Reservation
>
```

Figure 1: Compiling the project

```
0 | main | ~/Documents/Github/Airport_Reservation
> ./airline.out
Enter:
0 to exit,
1 to add a flight,
2 to book a flight,
3 to unbook a flight,
4 to look at a flight.
```

Figure 2: Launching the project

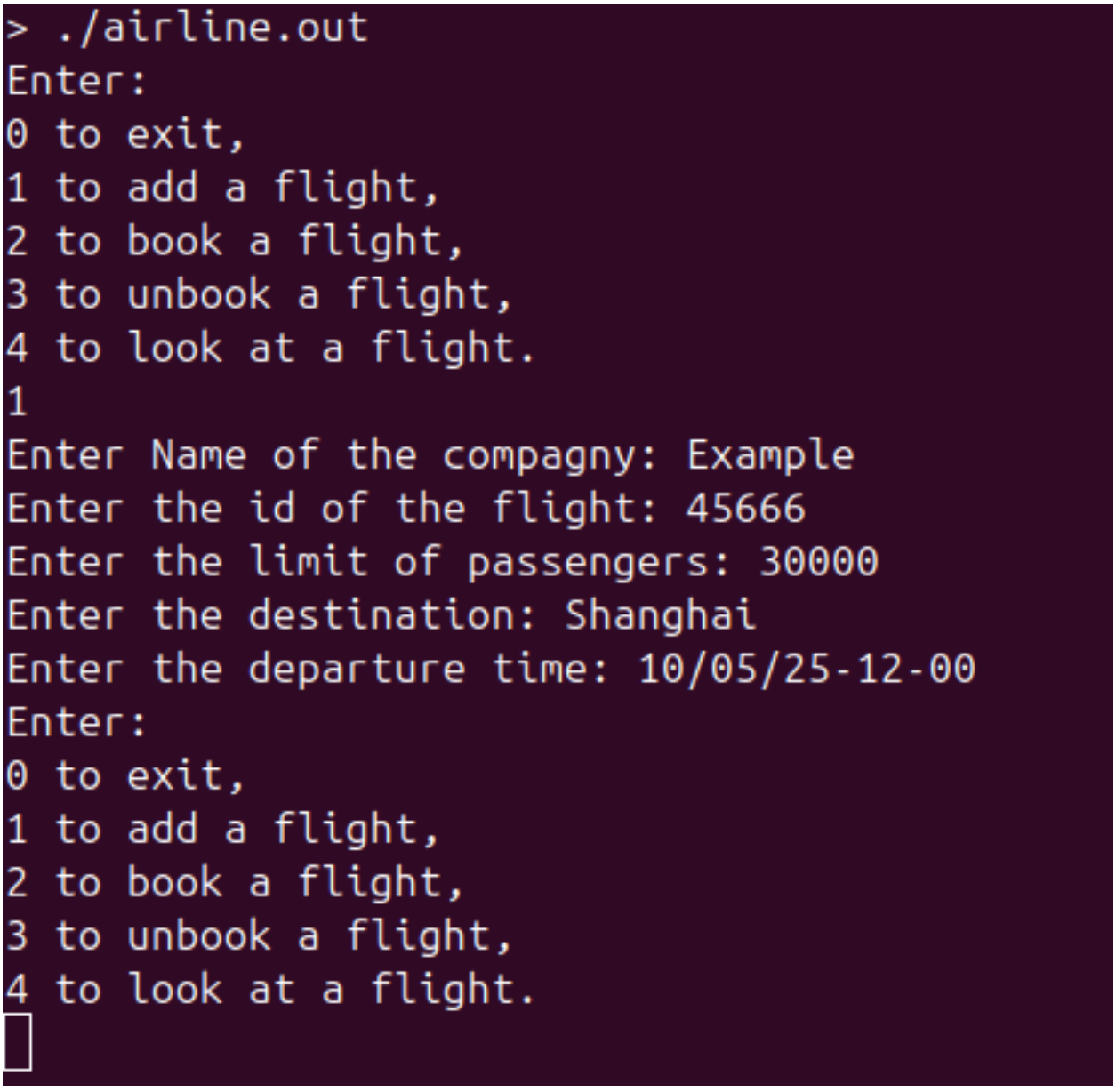

Figure 3: Adding a flight to the database

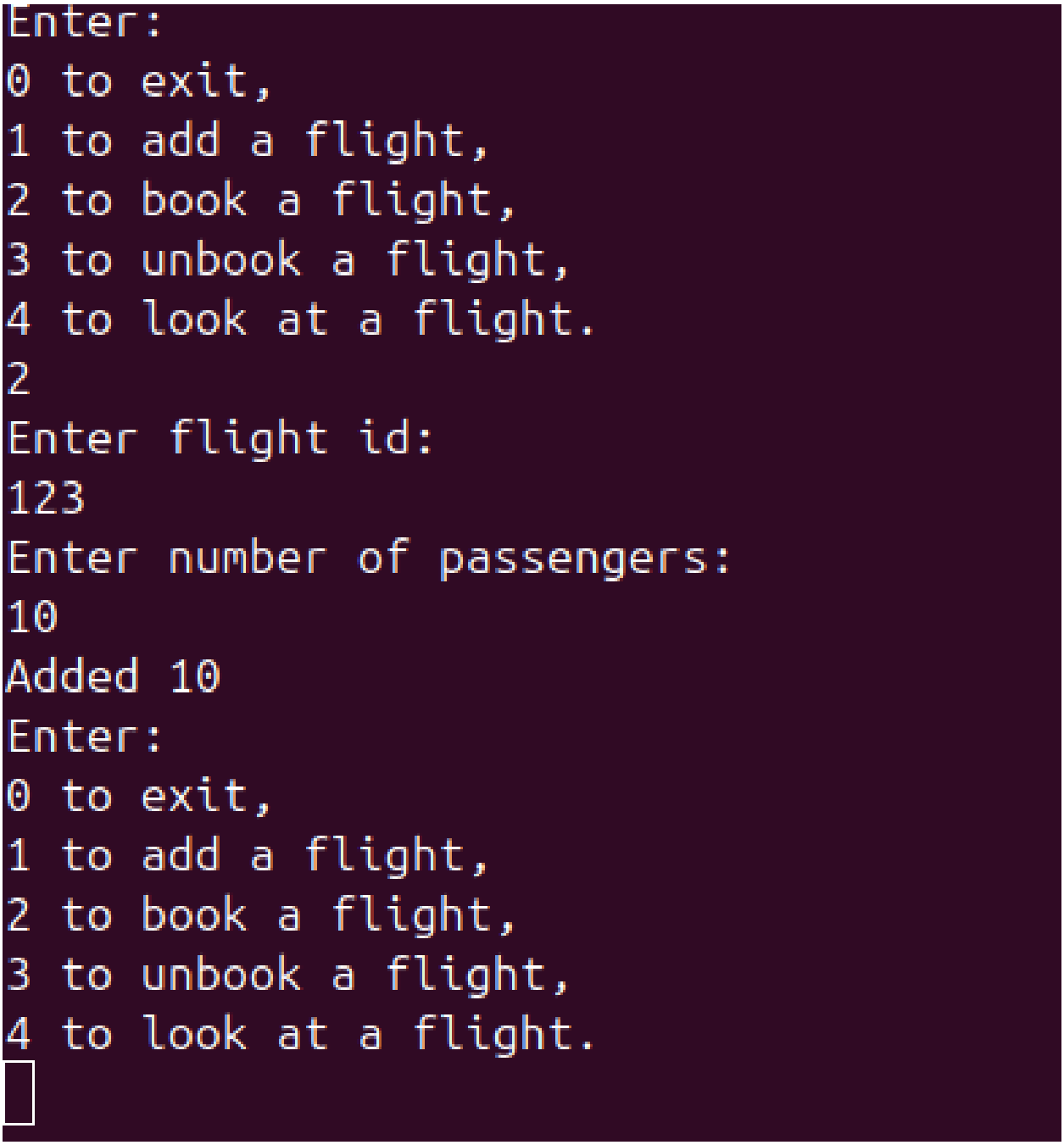

Figure 4: Booking a flight

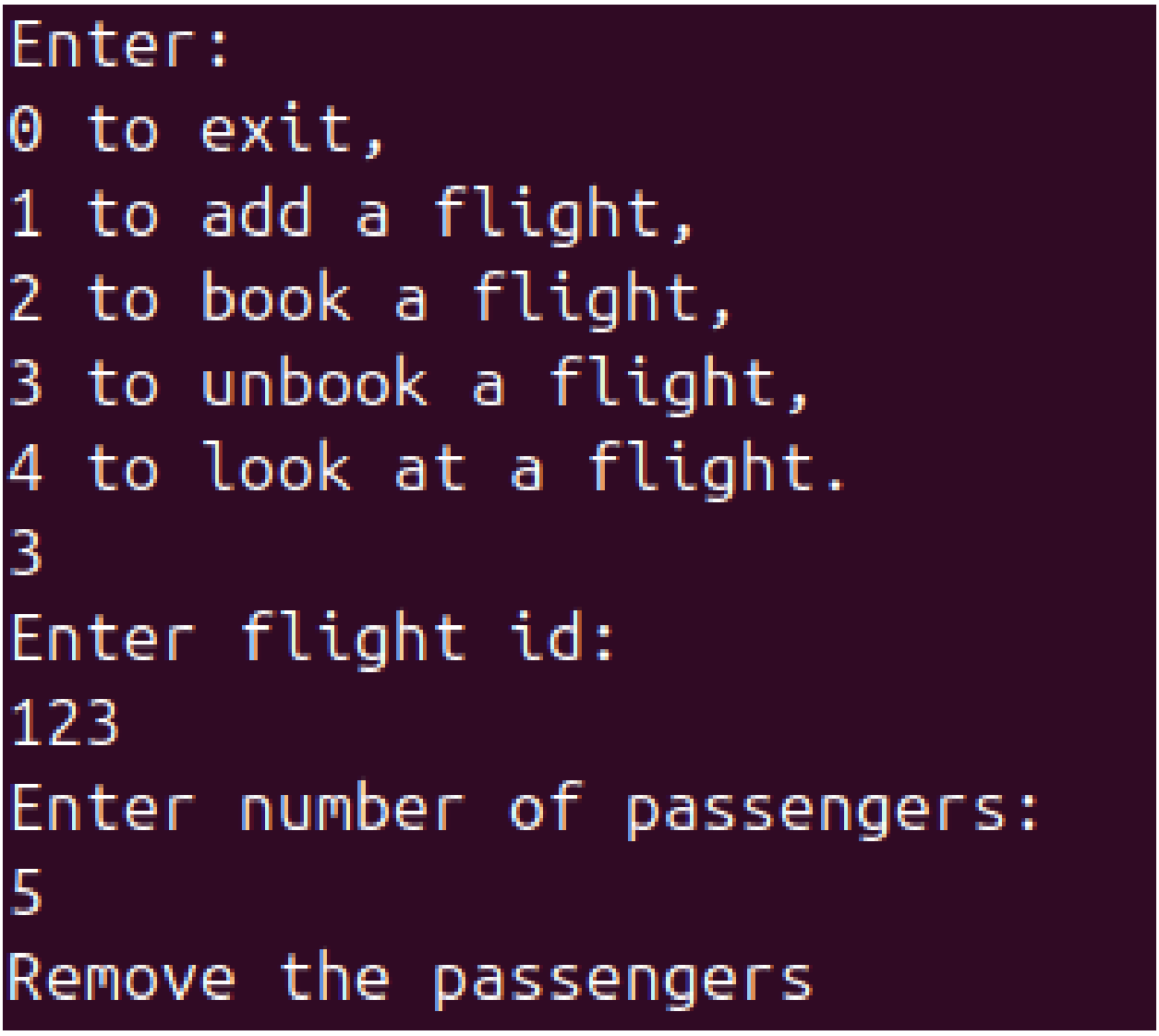

Figure 5: Withdraw from a flight

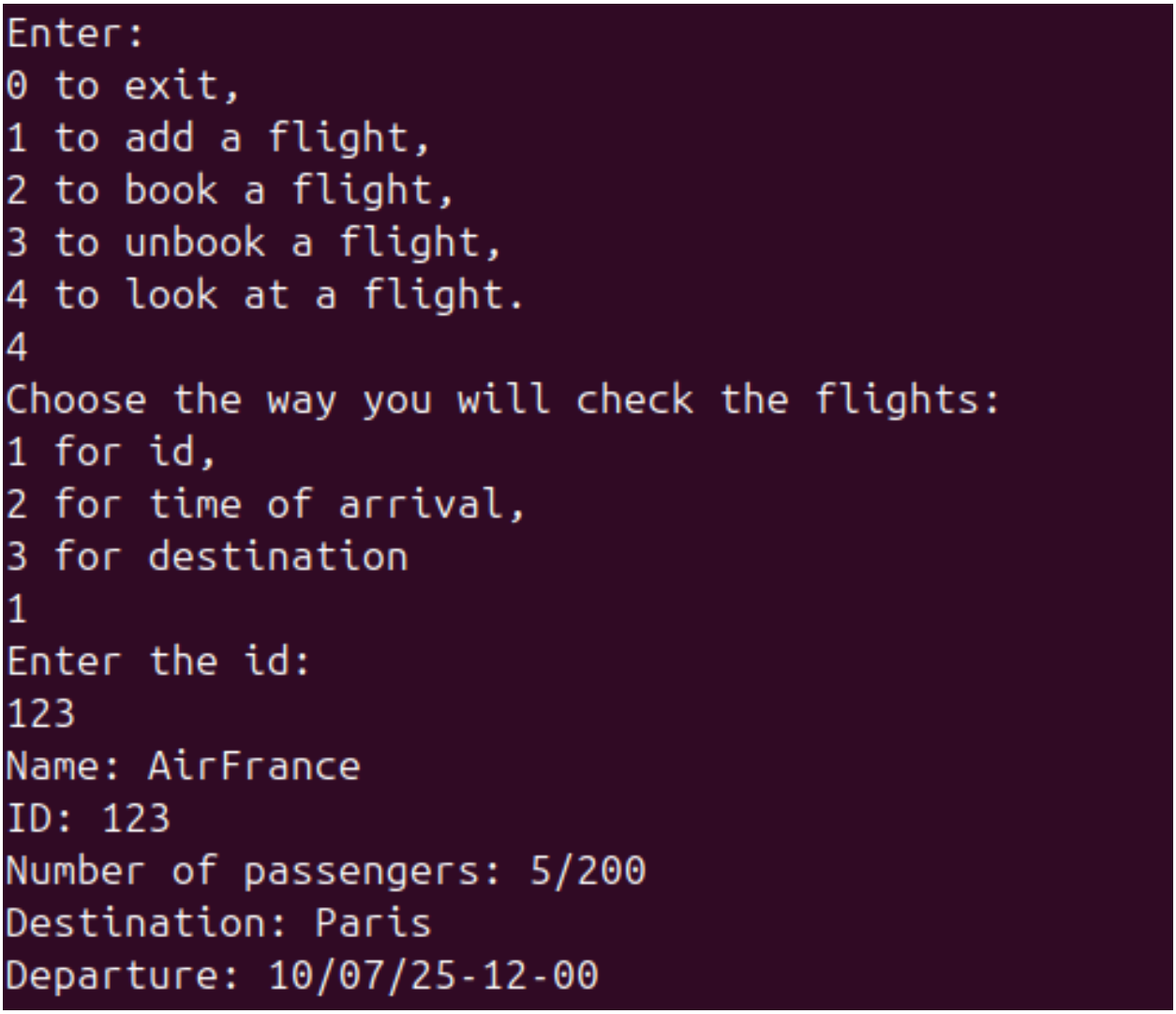

Figure 6: Checking the info of a flight

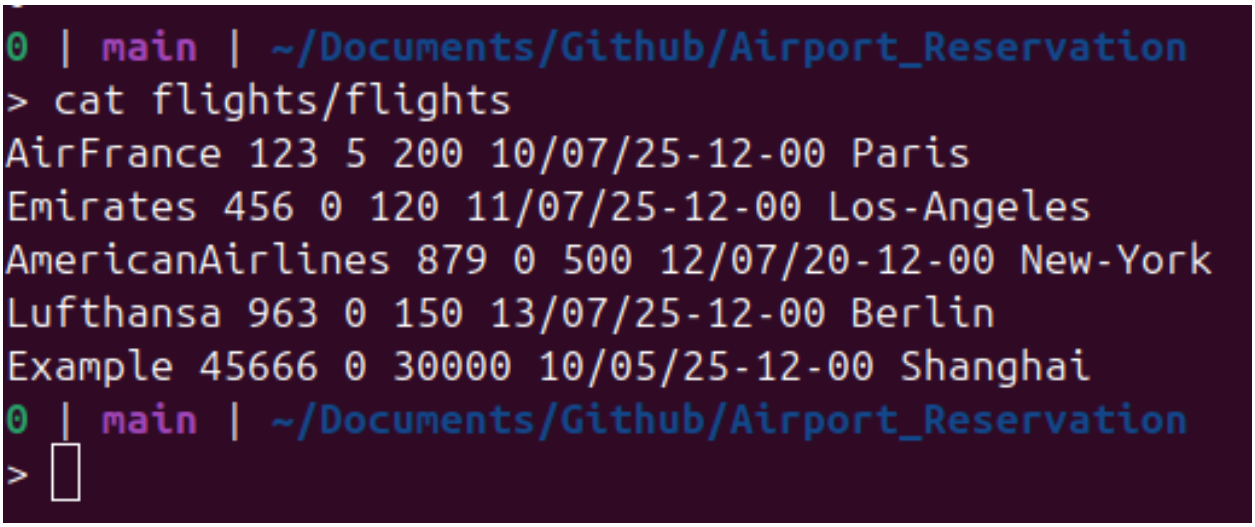

Figure 7: The file containg the data