LAB211 Assignment

Type:	**Long Assignment**
Code:	**J1.L.P0026**
LOC:	**400+**
Slot(s):	**N/A**

Title

Flight Management System – Read and Write File.

Background

An airline company aims to streamline its operations by implementing a **Flight Management System**. This system **should assist airline staff** in **managing flight schedules**, **booking** reservations, handling passenger **check-ins**, and providing necessary information to both passengers and crew members.
Flight information, passenger reservations, and crew assignments is **stored in a text file (product.dat).**

Program Specifications

Develop a Flight Management System with the **following key features**:

1. Flight **schedule management**
2. Passenger **reservation and booking**
3. Passenger **check-in and seat allocation**
4. Crew **management and assignments**
5. Administrator access for **system management**
6. **Data storage** for flight details, reservations, and assignments

Others- Quit.

Features:

This system contains the following functions:

- **Function 1: Flight schedule management – 50 LOC**
  - Airlines staff can **add new flights** to the system, including details such as **flight number, departure and arrival locations, departure time, and duration.**
  - **Create** a **Flight class** to represent flights. This class should include **attributes such as**:
    - Flight number (must be follow as: **F**xyzt, with xyzt is a number and no spaces, ex: F0001)
    - Departure city
    - Destination city
    - Departure time (check valid time)
    - Arrival time (check valid time)
    - Available seats
  - **The application asks** to **continuous** create new flight **or go back** to the main menu
- **Function 2: Passenger Reservation and Booking – 50 LOC**
  - Passengers can **search** for **available flights** based on **departure and arrival locations and date**.
  - Passengers can **make reservations** by providing **their personal information** (name, contact details) and **selecting a flight**.
  - **A unique reservation ID** is **generated** for each reservation

- The application should **go back** to the main menu and **wait for asking user to choose one option**.

- **Function 3: Passenger Check-In and Seat Allocation and Availability – 50 LOC**
  - **Passengers with reservations** can **check in** for their flights by **providing their reservation ID**.
  - The application will **generate boarding passes** with passenger and flight information.
  - The application **allocates seats to passengers during check-in** based on **availability**.
  - It will **display the availability of seats** for each flight, then it **allows passengers to choose available** seats.
  - The application should **go back** to the main menu and **wait for asking user to choose one option**.
- **Function 4: Crew Management and Administrator Access – 50 LOC**
  - **Manage crew assignments** for each flight, including **pilots, flight attendants, and ground staff**.
  - Administrators have **additional privileges to manage flight schedules, crew assignments, and system settings.**
  - The application should **go back** to the main menu and **wait for asking user to choose one option**.
- **Function 5: Save to file – 50 LOC**
  - Student should implement **a data storage solution to save** flight information, passenger reservations, and crew assignments.
  - The application should **go back** to the main menu and **wait for asking user to choose one option**.
- **Function 6: Print all lists from file – 50 LOC**
  - The system should **load all data from** the **file into the flight information list**
  - The application will **display flight information** list **that is sorted by date descending**.
  - The application should **go back** to the main menu and **wait for asking user to choose one option**.
- **Function 7: Create a layout – 50 LOC**
  - The application should be organized in the form of **a function menu**.
  - The application should **go back** to the main menu and **wait for asking user to choose one option to or quit** the application

- **Bonus 50 LOC (not over maximum 500 LOC in total of this project) if the student applies one of the Design Patterns (such as DAO pattern, Factory pattern, Repository pattern, and so on) in this project. More references for the design pattern:** https://www.tutorialspoint.com/design_pattern/index.htm
- The above specifications are **only basic information**. You must perform a **requirements analysis step and build the application according to real requirements.**
- The **lecturer** will **explain the requirement only once on the first slot of the assignment**.