Department of Electrical and Computer Engineering
Principles of Software Design
ENSF 480 – Fall 2023
Term Project

M. Moussavi, PhD, P.Eng

Total Mark: 100

Introduction:

In this assignment you are supposed to practice a complete process of design and development of a software project, using a systematic design methodology.

The last part of the course is focused on system level design and architecture and will be achieved in an "**Active Learning"** pedagogic approach. It means while theoretical concepts will be discussed in lectures, the corresponding practical techniques to build some design components will be developed in the classroom, or during the lab periods. This method shows you the initial directions and helps you to have a better vision on how to continue the details later.

When you are working as a member of group you should assume a full responsibility and your commitments must be achieved at your best capacity. All group members should be available during the lectures and labs, to participate in class/group discussions, and to achieve their own portion of work.

A Flight Reservation Web Application

In this project your task is to analyze, design, and develop a web-based system that can be used by different type of users, tourism agents, airline agents, and system admin(s). Some of the major requirements of the system for users and airline agents is to be able to:

1. Browse the available flights to a specific destination.
2. Select their desired flight.
3. Browse the seat map graphically.
4. Select their desired seat (regular, or business-class)
5. Select the desired ticket cancellation insurance, if interested.
6. Make payment, using a credit card.
7. Receive ticket via email.
8. Receive receipt for their payments via email.
9. Cancel their flight.
10. Etc.

Airline agents and flight attendants should be able to:

1. Browse the list passengers in a flight.

System admins should be able to manage flight information and other information on the database. For example,

1. Browse the list fights, their origin and destination in a specific date.
2. Browse the list crews in a specific flight (for example flight number AB123 to New York).
3. Brows the list Aircrafts that company owns
4. Add/remove a crew.
5. Add/remove an aircraft.
6. Add/remove flight destinations.
7. Add/remove/modify flights information.
8. Print list of users who have registered with the airline company. For more details see the additional information, below.
9. Etc.

Some Additional Information:

1. The application should be considered a single airline company. Aircrafts in this company, have options of different type of seats: Ordinary, Comfort, and Business-Class. The price of different seats can be managed by the company, to give you a rough idea, comfort seat is almost 40% more than ordinary seats, and business class seat is more than double.
2. Some users called "registered users", can register for the membership, and apply for company's credit card, and take advantages such as:
   - Receiving monthly promotion news (first day of each month)
   - Use airport lounges with a discount price.
   - Receive a free companion ticket once a year.
   - Etc.

   Note: the registered users information: name, address, etc. will be saved on the company's database.

Further information will be discussed during the Active Learning session. Same as real-world the project's functional and non-functional requirements can be subject to change during the development by your customers (course instructors).

Deliverables:

Design Phase (50 marks)

In this phase you should submit a Design Document that includes:

Part A – System Analysis:

1. System's description
2. System's Use-case diagram:
   This is the first draft of your use-case diagram. Mainly focused on business processes. In this use-case, you don't need to worry about administrative processes, such as: login, logout, etc.
3. System's Scenarios for each Use-Case, having all candidate objects underlined.
4. System's Conceptual Model:
   This isa very simple and very first draft of you class diagram that only shows class of domain objects and their simple association relationship with labels such as: uses, has, is-a, or other appropriate labels. In this diagram you don't need to worry about advanced software engineering terms such as aggregation/composition, inheritance, realization, etc.)

Part B – Domain Diagrams:

1. Highlights of the system's architecture (for example: client-server, web-based, etc.). In addition to a textual description, you should use a graphical presentation of the system.
2. Updated use case diagram.
   This is a use case diagram that adds additional (non-business-related) use cases. For example the processes that needs to maintain, deploy or administrate the system.
3. Systems activity diagram
4. Sequence diagram for at least four or five use cases (one per each member of the group)
5. State transition diagram for at least four or five GUI objects or domain objects such as ticket, payment, (one per each member of the group)
6. System's Domain class diagram: without attributes and functionalities (only relationships and multiplicities). In one page
7. System's Domain class diagram: with attributes, functionalities, and relationships/multiplicities. Can be in multiple pages. Please try to keep them well organized, clear, and easy to read. Marks will be deducted for disorganized of difficult to read diagrams of texts.

Important Notes about Domain class diagram:

- You class diagram should be traceable in your use-case scenarios. Mark will be deducted for class that are not traceable in the use-case diagram.
- In your class diagram you don't need to show Java library classes such as Exceptions, Buttons, String, ArrayList, etc.
- Make sure the multiplicity/cardinality is properly indicated.
- You don't need to show constructor/destructor, getters/setters.
- In this stage you need to apply all possible design strategies and techniques to make the architecture of the system more: reusable, scalable, maintainable, reliable, and using necessary concepts such as modular design, inheritance, realization, aggregation, composition, polymorphism, and appropriate design patterns as needed. <u>A minimum of 2 design patterns must be implemented.</u>

Part C – System's Detailed Design-Class Diagram:

In this section you need to provide the final detailed design. This diagram is like diagram in Part B-6, that includes:

1. Domain classes: identical to classes in B-6 with the stereotype << entity>>
2. Boundary classes: new classes
3. Controller classes: new classes

Part D – High-Level System's Architecture:

1. A Package Diagram
2. A Deployment Diagram

<u>Implementation Phase (50 marks)</u>

In this phase will implement your proposed design.
Development Options and Frameworks

- For the front end you have the option of using JavaScript or react.
- For backend, possible options are: Node.js, and MySQL

The details of implementation will be discussed later.

What to submit on the D2L?

A **zip** file that contains:

1. A **jar** file that contains all `.class` files.
2. A zip file that contains all `.java` files.

Due Date for both parts (Design and implementation): Wed Nov 29th, before 11:59 PM. You need to demonstrate your working application during lab periods on Dec 1, 4 and 6, based on a scheduled that will be posted on the D2L. During the demonstration, all team members MUST be present to assess their part of work. If a member is absent without a legitimate reason his/her mark will be zero. Otherwise, in case of a legitimate reason (sickness), the group's mark will be held back, until the absentee's work assessment completes.