

Department of Electrical and Computer Engineering

Principles of Software Design

ENSF 614 – Fall 2024

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 and related techniques will be discussed during lectures, then you apply it to your project. In other words, in this pedagogic approach, we show you the theories, techniques and the initial directions, then you will try to apply them to your group project in the best possible way.

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 Movie Theater Ticket Reservation App

For this project, you may assume a movie company called AcmePlex who has several movie theaters, has hired you to analyse, design, and develop a movie theater ticket reservation application that customers can purchase a ticket for a show in one of their theaters.

This system is expected to be used by two groups of users: ordinary users that can search for a movie, select a specific theater, view available movies and showtimes, view graphically available seats for a selected movie, select the desired seat, make payment by credit card, receive a copy of ticket and the receipt, via email. Users also should be able to cancel their ticket only up to 72 hours prior to show and receive a credit with %15 administration fee for future purchase up maximum of one-year expiration date. The other group of users are those who must be registered (let's call them Registered Users, RUs), and their information such as name, address, credit and/or debit card account must be saved on the system's database. RUs must pay a \$20.00 annual account fee, but they don't have to pay 15% admin fee for cancelling their tickets and will receive the movie news before public announcement. There is also one more constraint: Only 10% of the seats can be purchased by RUs on a first come first serve policy prior to public announcements.

Your Design document must include all the required details of the full version of the desired system, but your implementation responsibility can be limited to a single theater development. We will discuss further detail in one of the upcoming lectures.

Deliverables:

Design Phase (50 marks)

In this phase you should submit a Design Document that includes a clear description of system's requirements, and design artefacts as follows:

1. Introduction to the system under the study.
 - a. Brief description of the system
 - b. A system activity diagram that shows the process of browsing and selecting a movie
 - c. A system activity diagram that only shows the process of booking ticket
 - d. A system activity diagram that only shows the process making payment
 - e. You can add more activity diagram(s) if you will (optional)
2. Systems use cases.
 - a. Use case diagram
 - b. A detailed "Scenario" for each use case, **having all candidate objects underlined, and all candidate operations underlined** (use single-underline for objects and double-underline for operations).
 - c. System interaction diagrams for at least four major and important use cases in this system (each team member must take responsibility for one and his/her name must appear on the related page).
3. Domain Layer Class Diagram that includes:
 - a. A class diagram that shows the domain objects and concepts, their attributes, behaviours, cardinalities etc.
 - i. **Note:**
 - b. Your 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.
 - c. In your class diagram you don't need to show Java library classes such as Exceptions, Buttons, String, ArrayList, etc.
 - d. Make sure clearly identify the stereotypes of your classes (such as <<entity>>, <<control>>)
 - e. Make sure the multiplicity/cardinality is properly indicated.
 - f. You don't need to show constructor/destructor, getters/setters.
4. A state transition diagrams for the following objects:
 - a. Ticket object
 - b. Payment object
 - c. And two uses cases in your system
5. Presentation Layer Class Diagram that includes:
 - a. A class diagram that shows the presentation layer objects with their attributes, behaviours, cardinalities etc.
 - i. **Note:**
 - b. Your 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.
 - c. In your class diagram you don't need to show Java library classes such as Exceptions, Buttons, String, ArrayList, etc.

- d. Make sure clearly identify the stereotypes of your classes: <<boundary>>
 - e. Make sure the multiplicity/cardinality is properly indicated.
 - f. You don't need to show constructor/destructor, getters/setters
6. A Package Diagram that shows how the three layers of the system interact with each other. Make sure to show classes in each package and identify those that are public.
 7. A Deployment Diagram that shows how the client side, domain, and database layer are related.

Notes:

- 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.
- Your submitted design document must have all the above-mentioned section, in the order that is explained above (numbers 1 to 7 indicates first level of a table of content numbers). The sub sections must appear as 1.1, 1.2, etc.

Part II (50 marks)- Implementation Stage

In this phase will implement your proposed design, in Java. The details of implementation will be discussed later.

What to submit on the D2L?

1. A **jar** file that contains all **.class** files. File name should be **yourGroup#.jar**
2. A **zip** file that contains all **.java** files. File name should be **yourGroup#java.zip**
3. A **SQL Dump** file (**.sql**) that contains all **SQL** queries need to recreate tables. You can either use **mysql** built-in tool to export database schema and data. File name should be **yourGroup#.sql**
4. A **zip** file that contains web application files as needed such as: **html**, **css**, or **.js**. File name should be **yourGroup#Web.zip**
5. A **zip** file that contains configuration files, as needed. File name should be **yourGroup#Config**.
6. A **readme.txt** explains details of deploying the application and running on a new computer.

In addition to above documents, you need to provide a short video (all members of the group must participate). An instruction in this regard will be published on the D2L .

Due Date for the design document, implementation, and demo video: Tuesday Nov 26, 2024, before midnight (11:59 PM).

Also, you need to demonstrate your working application during the period of Wed Nov 27 to Wed Dec. 4 during the lectures or lab periods. A demo schedule will be posted on the D2L and ALL members of each group should participate in the demo sessions to answer question about design and implementation of the project.