

**Department of Electrical and Computer Engineering**  
**Principles of Software Design**  
**ENSF 480 – Fall 2021**  
**Term Project**

*M. Moussavi*

**Total Mark: 100**

**Introduction:**

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

As I mentioned before 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. 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.

**Online Rental Property Management System:**

Let's assume you have been hired by a company to analyse, design, and develop an online **Property Rental Management System (PRMS)** that its minimal requirements include:

- RQ1: Landlords should be able to register their properties using this online application. Once their property is registered they should be able to pay certain fee and make their property posted online and be available for renters to view it. The fee is certain amount in dollars for a fixed period of time (example 60 days).
- RQ2: Regular renters don't need to login, and should be able to enter their search criteria such as:
  - Apartment, attached/detached house, townhouse, etc.
  - Number of bedrooms,
  - Number of bathrooms
  - Furnished/unfurnished
  - City quadrant: SW, NW, NE, SE
- RQ3: Registered renters, must first login. This group of renters, in addition previous functionalities for regular renters, will have the privilege of being notified, when new listings is posted that matches their search criteria. Also, they should be able to unsubscribe at any time.
- RQ4: Renters who need further details should be able to send email to the landlord without seeing the landlord's name, or landlord's email address.
- RQ5: Renters who are interested in property should be able to send email to landlord to arrange a meeting to view the property, and possibly sign a contract. Note: Your system will only provide email communication between renters and landlord, and doesn't do anything with signing the contract and rest of the details.
- RQ6: Managers should be able to login in, set or change the amount and period of fees, and should have a full access to the renters, landlords, and properties information via company's database system, which can be some sort of light database. Please notice your application need to access this database engine to save or retrieve properties, and registered customers, and landlord.
- RQ7: Managers should be able to ask for a periodical summary report that shows:

- Total number of houses listed in the period. Notice that some houses that are listed may not be active anymore. It means some houses their posting period can be expired or landlords have cancelled their posting, therefore the renters cannot view them anymore.
- Total number of houses rented in the period
- Total number of active listing.
- List of houses rented in the period. Which displays, the landlord's name, the house's id number, followed by its address.
- RQ8: Managers and Landlords should be able to change the state of a listing, from active, to rented, cancelled, or suspended.
- RQ9: The system needs to have flexibility for future changes
- RQ10: Other requirements, as might be discovered during development process.

## **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 in the following order:

- Systems use case diagram.
- System interaction diagrams for each use case. You only should draw a new use-case diagram if a use case introduces a new object. In a real world situation you may need to draw many sequence diagrams, but for the purpose of this course and because of our time limitations, each member of the team in a group should draw a sequence diagram. Means a group of 4 must draw at least 4 sequence diagrams and a group of 5 must draw at least 5 sequence diagrams.
- A design level detailed class specification that includes:
  - A class diagram with only major attributes and behavior that only shows the class name and the relationships among them (no getter, no setter, no constructor, no clone method). The purpose of this diagram is to show the entire system in one page. Please make sure lines do not cross each other and the diagram is clear and easy to read.
  - A detail class specification, which is class diagram with no relationships (no lines), only showing the class details: attributes and behaviours. This diagram can be organized in multiple pages. Please try to keep them well organized, clear, and easy to read. You must keep classes that are in the same package on the same page.
  - A state transition diagram for major and core classes in the system or a use case diagram (one diagram per team member). Examples
    - Property Class
    - A use case that landlord posts his/her property
    - A use case that of landlord making fee-payments
    - Etc.
- Systems activity diagram
- A package diagram
- A deployment diagram

**Note:** 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.

**Due Date for Design Document: Monday November 29 11:59 PM.**

## Part II (50 marks)- Implementation Stage

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

**Due Date:** Wed Dec 8, at 10 AM.

### What to submit on the D2L:

1. A **jar** file that contains all `.class` files.
2. A tar file that contains all `.java` files.
3. A one-page document that summarizes each member activities for the term project. You should provide a table to identify how each team member contributed to this project. Here the format of the table and an example:

Members Names	% of contribution	Code developed by each member	Modelling contribution
	25%	Entity classes: Renter, Landlord, Observer, Login, Logout	Sequence Diagram for “Browse Property Use Case”, State transition diagram for Property, Class Diagram, ...
	25%		
	25%		
	25%		

4. A video recording that demonstrates the functionalities and features of your working project. Your demo will be marked based on the following scoring rubric:
  - a. Graphical User Interface (GUI): 5 marks
  - b. Database connectivity and database access 5 marks
  - c. Implementing all requirement 30 marks
  - d. Reasonable match between design and implementation 5 marks
  - e. Use of design patterns 5 marks

Note:

- Marks will be deducted for lack of documents and reasonable error checking in your source code