Important:

1. **Upload the design document as word or PDF.**
2. **The required code must be uploaded in JAVA (as NetBeans or Eclipse project)**
3. **Each group must be at most two students**
4. **You are required to defend your work. If not, you get automatically ZERO.**

The design document MUST provide the following:

1) Use cases Analysis: Provide the use case diagram. You should provide the following information about each of the provided use cases
   - *A detailed textual description according to the template discussed in the class lectures. (see slide #5)*
   - *An Activity Diagram for each use case description*
2) The System Context Model (see slide #6)
3) The Boundary Objects Class Diagram (see slide #9)
4) The Entity Class Diagram: show only classes names with relationship between them (see slide #7).
5) Provide the attributes for each entity class (see slide #8).
6) Develop the communication diagram for each use case. Each use case diagram must contain the following information (see slides 13 &14):
   - *Your diagrams should show control objects (see slide #10)*
   - *Identify all the active (thread) objects in your system*
   - *Identify all the message types: synchronous or asynchronous*
7) Expand all the entities in your previous entity class diagram (step #4 & step #5) to provide access methods to the information part. Note: this step depends on the passed messages on the communication diagram (see slide #15)

Note: You may combine steps 4, 5, and 7 above (if it is easier for you).

Required Code:

You need to provide the code for ONLY the entity classes (step #4 & step #5). This means only entity classes with their attributes (No classes methods), and the code should also show the relationship between classes (*Hint:* static relationships are shown in attributes and classes). The code **must match** the diagram. *The code is only required to compile at this time*.

Problem Description: Dentist Office System

A dentist office is currently using paper files to manage all office activities. To make their work easier, the office decided to build a software system to manage those activities instead of using their old paper-way methods.

The office patients can take their appointments either by phone or by walking in person to the office lobby. In either way, the receptionist schedules an appointment for the patient with the desired date and time. The system shall not allow scheduling appointments with the same doctor for a duration of less than 30 minutes. Of course, the system shall accept appointments for a specific doctor if the doctor is working on that time. This means that the system shall keep track of each doctor working schedule. The office receptionist enters doctors working schedule in the form of working days for each week. For example, a doctor work schedule can be on Monday, Wednesday, and Thursday from 9:00 am until 4:00 pm for all weeks until June 01, 2021. Now, the office receptionist can cancel a doctor specific day schedule, or group of days, or the entire schedule. If this happens, the system should generate a list of the patients affected by this cancellation, allowing the office receptionist to call them by phone to tell them about this cancellation.

Further, the system needs to create a file for each patient upon scheduling first appointment. The system keeps the following information about each patient: name, unique patient number, phone number, owed balance, total of payed balance, email (optional), and details about any previous visits (which each visit should contain description and the cost of that visit).

Furthermore, the system shall send reminders by email (if exists) to patients when their appointment is getting closer. Of course, the system should also allow printing a specific visit details.

Assume the following:

1. The system is intended to be used as a single system on one computer
2. Ignore Login operation into the system