Assignment: Implementing a Clinical Decision Support System for a Pain Assessment Medical Guideline
Due dates: Demos—In-class on November 17; other artifacts—November 17, 11:59 PM
This is a *team* assignment.

Part I: Implement a clinical decision support (CDS) system for assessment of pain in older people.

- High-level Requirements:
  - The CDS system must have a graphical user interface.
  - The CDS system must guide the user (a clinician) through the same sequences of questions as shown in the flowchart in Appendix 3 of the *The Assessment of Pain in Older People* by the British Pain Society and the British Geriatrics Society (document posted on Blackboard). In particular, the system should allow the user to input an answer to each question and should automatically display the next question(s) (or the final recommendation(s)) based on the user's answers.
  - The CDS system must capture *all* the information from the flowchart, i.e., all questions and the correct decision logic.
  - The step "take detailed pain history" from the above flowchart should be elaborated using the flowchart produced by the Quinnipiac nursing students (also posted on Blackboard).
  - If you need to make any assumptions because of ambiguity in the flowcharts, run those assumptions by the course instructor before you implement the system and make sure they are clearly stated in the reflection write-up (see below) and during the demo of your CDS system.
- You can use any technology/programming language as long as your application satisfies the above requirements and the instructor can install and run your system on his machine.
  - Sample code for GUI applications in Java is posted on the class web site. If you decide to use Java, you can refer to this code as an example of the general structure of Java GUI-based applications.
- Before implementing the CDS system, create the following design documents:
  - GUI mock-ups for each screen of your application.
  - A story board showing the transitions between the different screens and the events that trigger these transitions.
  - If your implementation is object-oriented, create a UML class diagram (https://en.wikipedia.org/wiki/Class_diagram) showing your classes and their relationships. Each class in the diagram should also show the methods and the fields of that class.
  - If your implementation is not object-oriented, use some other kind of diagram that represents your design. For example, you might want to use a kind of behavioral diagram, e.g., UML sequence diagrams (https://en.wikipedia.org/wiki/Sequence_diagram) or a state diagram (https://en.wikipedia.org/wiki/State_diagram_%28UML%29). Other behavioral diagrams can be found at https://en.wikipedia.org/wiki/Unified_Modeling_Language#Diagrams
- As you implement the CDS system, regularly update the design documents.
  - E.g., if you introduce a new class or add a new method to a class, update the class diagram accordingly.
  - For your final submission, the implementation and the design documents need to be in sync with each other.
- Carefully document your code.
  - Include comments that explain the purpose of each class, method/function, and instance or class/static variable.

- Create a README.txt file with detailed and precise instructions for installing and running your system.
- Write a brief reflection on your experience with implementing the CDS system. What were the major difficulties? What were the easy parts? What would you change in your development approach if you had the opportunity to start from scratch? Is there anything else related to the development of this CDS system that you thought was interesting?

Part II: Demo your implementation of the CDS system for pain assessment. Plan for a 10-15 minute presentation.

- Show the flowcharts that you used and explain any assumptions you made.
- Show how your CDS system works to the rest of the class.
- Go over the class diagram (or whatever other design diagram you decide to use) for your system; describe the internals of the system.
- Summarize the reflection mentioned above.

Artifacts to submit:

Make sure you attach all necessary files before you submit the assignment, because Blackboard will not let you resubmit. Please, make a single submission per team.

- The source code of your CDS system
- The README.txt file
- The design documents described above
- The reflection-write-up