Question 1 (10pts): Suppose that the FBI wants to hire you to build a system for keeping track of cases that they are trying to solve. For each case, they will want to track several hundred (or maybe even several thousand) pieces of information about the case's evidence and suspects. At present, however, the FBI only has a vague idea about exactly which of these pieces of information really need to be part of the system.

It appears that the system will have three parts:

1. a web application where detectives can log in, then edit and search the data
2. a program that periodically downloads all the evidence data and makes a backup of it
3. a program that periodically downloads all the suspect data and publishes it via an XML feed that other law enforcement agencies can access

There probably are several alternatives for designing each of these parts of the system. The FBI is extremely eager to have the first part of the system working soon. They can wait for the other two parts to be implemented later.

1a) Would you use an *iterative* or an *incremental* process for building this system, and *why*?

Incremental, since the customer clearly wants just one part to be delivered very quickly, with other parts added later. (Iterative would have been preferable if the customer wanted a fast, basic implementation of the entire system, with improvements throughout the system delivered later.)

- 3 points for responding "incremental"
- 2 points for giving a valid reason

1b) Would you use a *waterfall*, *spiral*, or *agile* process for building this system, and *why*?

Spiral is often a good choice for larger systems with vague requirements and many alternatives for designing and coding.

- 3 points for responding "spiral"
- 2 points for giving a valid reason

Question 2 (10pts): Your FBI customer tells you a lot about the system, including these statements:

1. "We might need to add some additional features in the future. At the FBI, we always need new features to be added to old systems." maintainability
2. "We initially will run the web application on Windows, but are hoping to switch to Linux some day." portability
3. "The XML feeds need to have formats that are compatible with the XML feeds required by the CIA and other agencies." interoperability
4. "It should never be possible for people to get into the system unless if they have authorization." integrity
5. "The system should be working all of the time. We can't afford to have it unavailable for even a single day, even on holidays." reliability

Each of those statements above is related to a quality attribute. Based on the five statements above, which *five of the ten* quality attributes shown below are probably most important for this system?

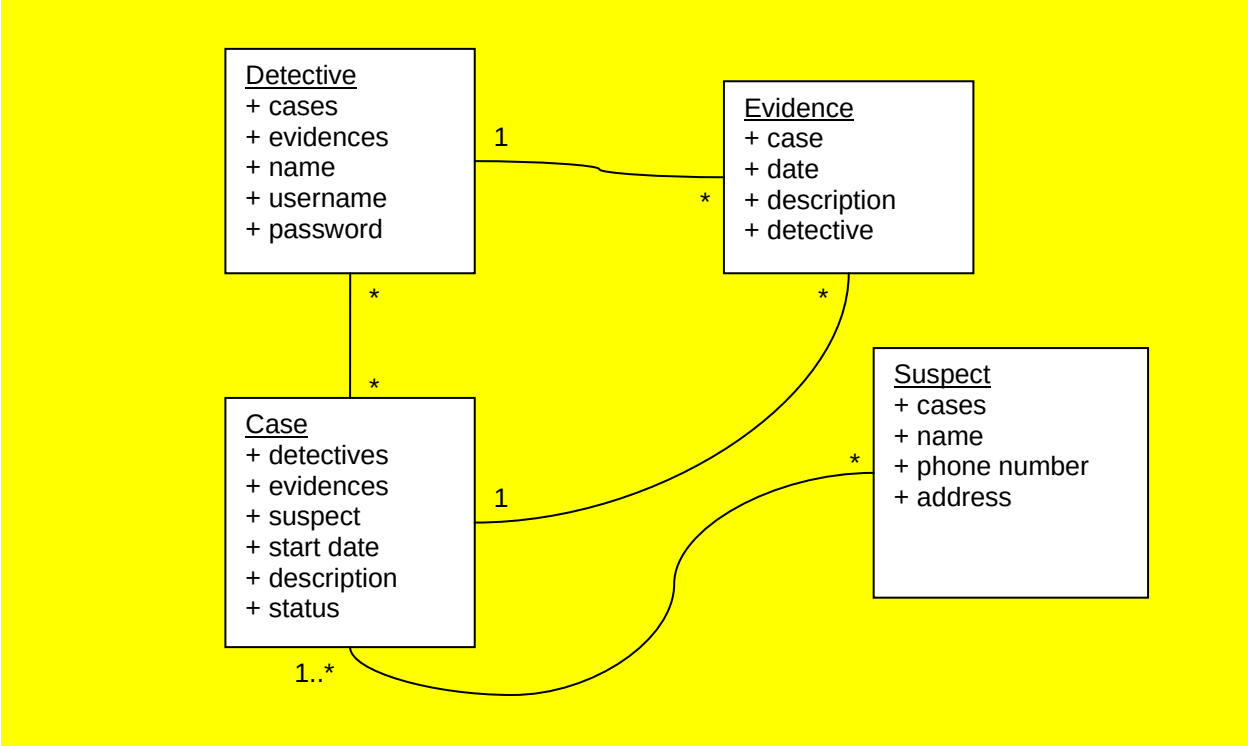
	Yes	No
Reliability	X	
Efficiency		X
Integrity	X	
Usability		X
Maintainability	X	
Testability		X
Flexibility		X
Portability	X	
Reusability		X
Interoperability	X	

(Check "Yes" for 5 quality attributes and "No" for 5 quality attributes)

- 1 point for each of the ten checkboxes in the table above
  - Ok to omit the "No" checkbox, as long as the corresponding "Yes" option is not checked

Question 3 (20pts): Suppose that the key entities in the FBI system are *detectives*, *cases*, pieces of *evidence*, and *suspects*. Each detective has zero or more cases, and each case has zero or more detectives. Each case also has zero or more pieces of evidence, as well as zero or more suspects. Each piece of evidence belongs to exactly one case, but each suspect is attached to one or more cases. Each piece of evidence has a date, a description, and one detective who collected the evidence. Each detective can have zero or more pieces of evidence. Each detective has a name, username, and password. Each case has a start date, a description, and a status. Each suspect has a name, phone number, and address.

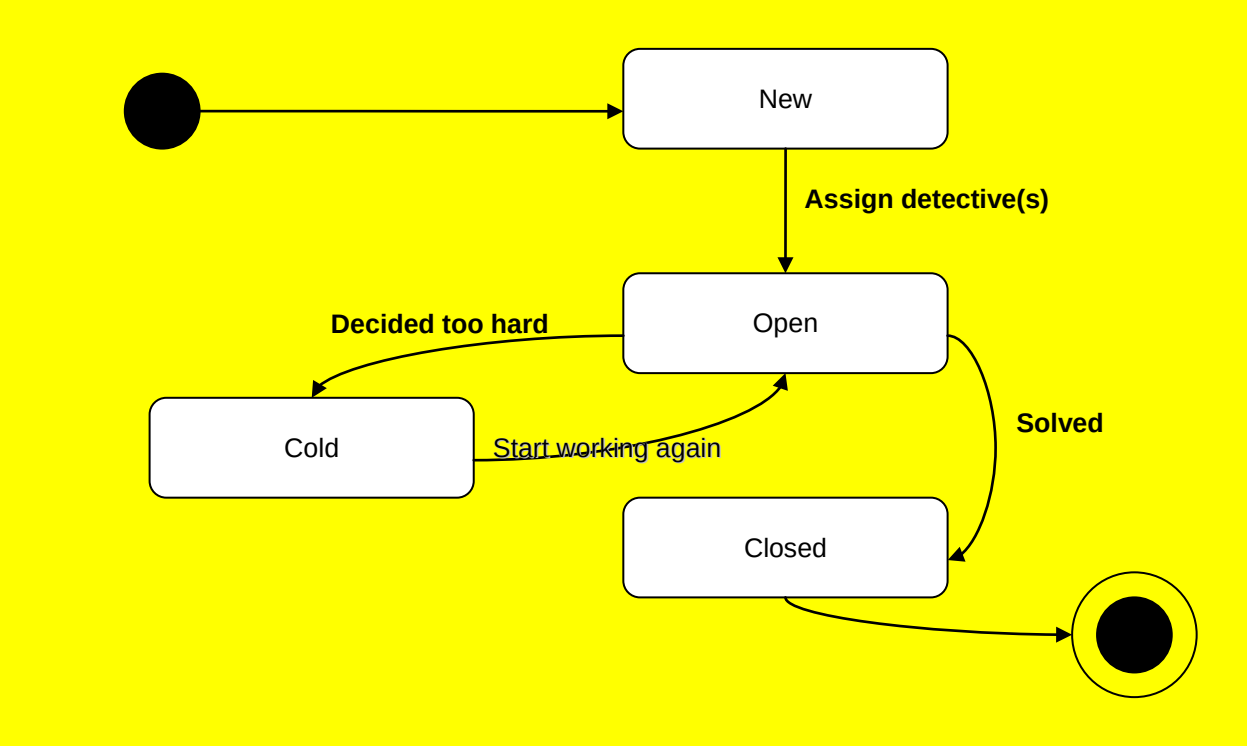
Draw a UML class diagram showing these four entities, their relationships, and their attributes. Remember to show cardinality on relationships. You do not need to show the types for attributes.



- 1 point for each of the boxes in the diagram, correctly matched to the entities (total: 4 points)
- 1 point for each of the lines in the diagram, including correct cardinality (total: 4 points)
- 1 point if zero lines have arrowheads (ie: there should be no arrowheads)
- 1 point for each of the following attributes in boxes of the diagram (total: 11 points)
    - Evidence date, description
    - Detective name, username, password
    - Case start date, description, status
    - Suspect name, phone number, address
- Slight variations in naming are acceptable, eg “Detectives”, “Pieces of Evidence”, “start_date”
- Attribute types are optional
- The + “public” indicator on attributes is optional

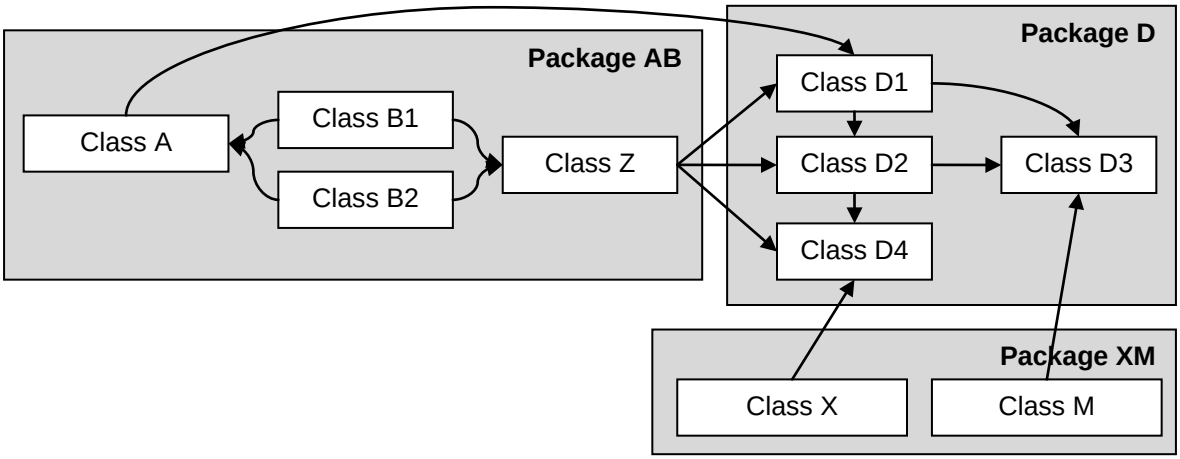
Question 4 (15pts): Suppose that a case initially starts out as “new”. When one or more detectives are assigned to a new case, the case becomes “open”. At some point, the detectives might decide that an open case is too hard to solve right now, at which point they might decide to mark the open case as “cold” (which means that they’re not going to work on the case for a while). Eventually, the detectives might decide to start working again on a cold case, at which point the cold case again becomes open. A case could possibly switch back and forth between open and cold for many years. Usually, the detectives eventually figure out the case while it is open, and they mark it as “closed”, where it stays closed forever.

Draw a state diagram for a case.



- 1 point for each of the boxes in the diagram, correctly matched to the states (total: 4 points)
- 1 point for each line that connects boxes (total: 4 points)
- 1 point for having a sensible label on each line (total: 4 points)
- 3 points for having appropriate start and end markers
    - You can do this as shown above
    - Or you can do this as in formal language theory, ie: the “start” line points to the start state (New), omitting the filled black start circle; and the final state (Closed) is circled, omitting the circled filled black end circle; for an example of this notation, refer to Wikipedia http://en.wikipedia.org/wiki/Deterministic_finite-state_machine

Question 5 (10pts): Consider the following approach of implementing the FBI system.



The concern of Package AB is to provide a user interface.
The concern of Package D is to load and store data.
The concern of Package XM is to retrieve data for specialized purposes.

5a) When Classes B1 and B2 call Class Z, Class Z then modifies instances of Classes D1, D2, and D4. Now *suppose that Class Z was moved into Package D*. Would moving Z into D increase or decrease the coupling of the implementation? Why?

It would decrease coupling. There are two valid reasons. One reason is that fewer lines would cross package boundaries (going from 4 AB-D lines down to 3 AB-D lines). The other reason is that since Z modifies D1/D2/D4, Z has content coupling to those classes; in contrast, B1/B2 only call Z, which is control coupling. Since control coupling is a weaker coupling than content coupling, it is preferable to have control couplings cross package boundaries, rather than content couplings. Therefore, moving Z into D would decrease overall coupling by reducing the strength of the coupling from AB to D.

- 2 points for "decrease"
- 2 points for valid reason

5b) The classes in Package D together form a Composite pattern. (Specifically, each instance of Class D1 owns some instances of Classes D2 and D3, and each instance of Class D2 owns some instances of Classes D3 and D4.) What principle is violated by the implementation shown in the diagram? (Select one of the four options below.)

A. Packages AB, D, and XM form a cycle
B. Classes D1, D2, D3, and D4 should be related via inheritance, not composition
C. Classes Z, X, and M violate the Law of Demeter
D. Classes D2 and D3 fail to implement an interface

- 3 points for answering C

5c) Which of the following should be done to improve the quality of the implementation? (Select one of the four options below.)

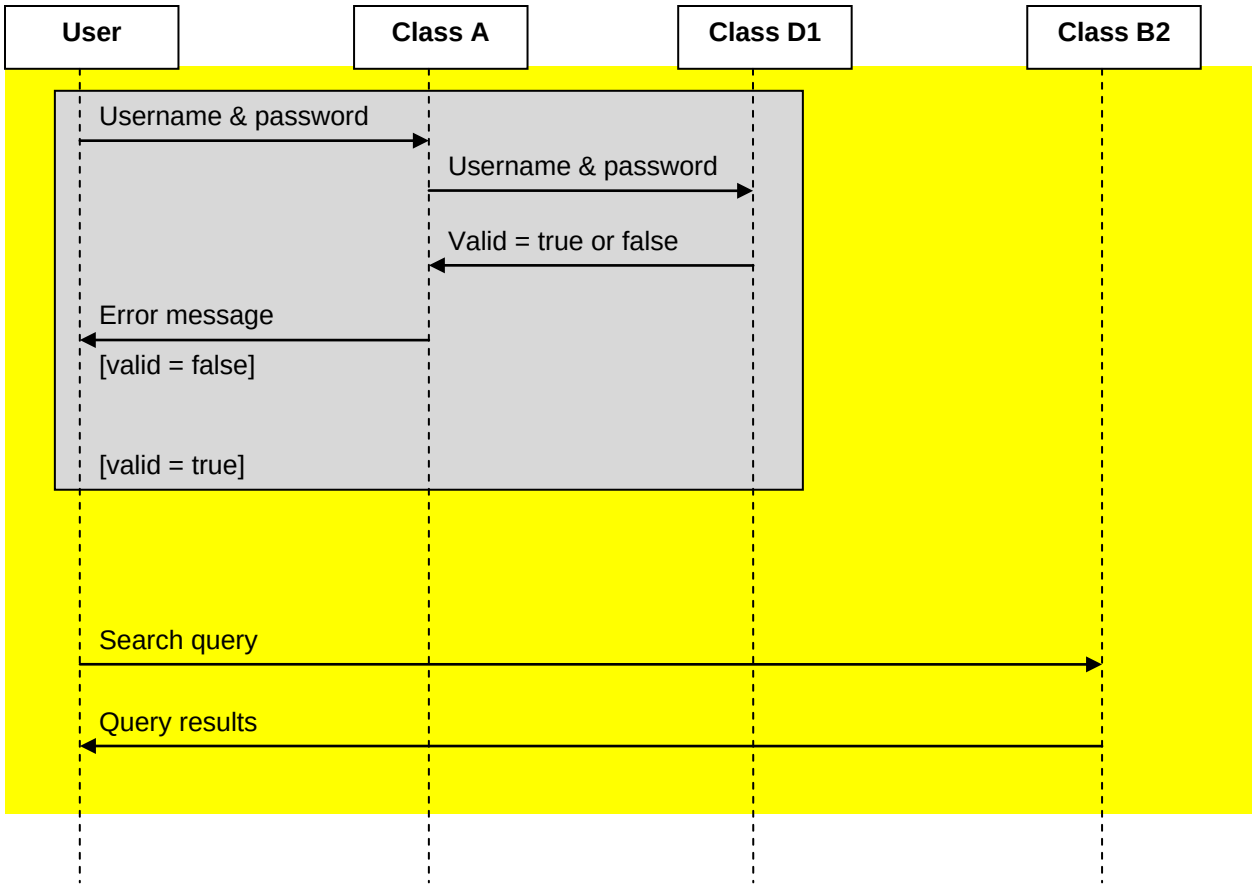
A. Use sandwiching to split Package D in half
B. Only allow Classes Z, X, and M to access the data via Class D1
C. Eliminate Class Z and make B1 and B2 access classes D1, D2, and D4
D. Just move Class D1 from Package D to Package XM

- 3 points for answering B

Question 6 (15pts): Here is how the system (diagrammed in the previous question) implements a use case required by the FBI:

Step 1. The user sends username and password to an instance of Class A
Step 2. An instance of Class A asks an instance of D1 whether the username and password are right
Step 3. If D1 tells A that the username and password are invalid, then A shows an error message to the user, who returns to step 1, above; steps 1 and 2 are repeated until the username and password are valid.
Step 4. Once the username and password are right, the user can send a search query to an instance of Class B2
Step 5. Finally, the instance of Class B2 returns search results to the user

Complete the following message sequence diagram showing these events.



- 1 point for each of the lines (6 points total)
- 2 points for having a reasonable label on each right-ward line (6 points total)
    - Left-ward lines may omit labels, as each is just a return value
- 1 point for having a condition on the "Error message" line shown above, or a suitable wording on the label so that it's clear that the error message is only shown when the username/password were invalid (1 point)
- 2 points for having a "repeat" loop box around the authentication messages, with a suitable guard so that the loop is only exited when a valid username/password have been provided

Question 7 (10pts): The table below describes the complexity of each item in this implementation. (Note that Classes D1, D2, D3, and D4 are *together* just one single 3GL component, so the system has a total of four 3GL components.)

Class	This is a…
A	3GL component
B1	Screen, simple complexity
B2	Screen, simple complexity
Z	3GL component
D1-D4	3GL component
X	3GL component
M	Report, difficult complexity

7a) How many application points does this system have, in total?

Each simple screen is 1 a.p. Each 3GL component is 10 a.p. The difficult report is 8 a.p. Adding these up, 1*2 + 4*10 + 8 = 50 a.p.

- 4 points for getting the right answer of 50
  - 1 point partial credit for showing work.

7b) Suppose that your team has *nominal experience and capability* with creating this kind of system, and your team has very *low CASE maturity and capability*. What is the expected productivity of each team member, in application points per month?

Nominal experience and capability supports 13 a.p./person-month. Very low CASE maturity and capability supports only 4 a.p./person-month. The average of these is (13+4)/2 = 8.5 a.p./person-month

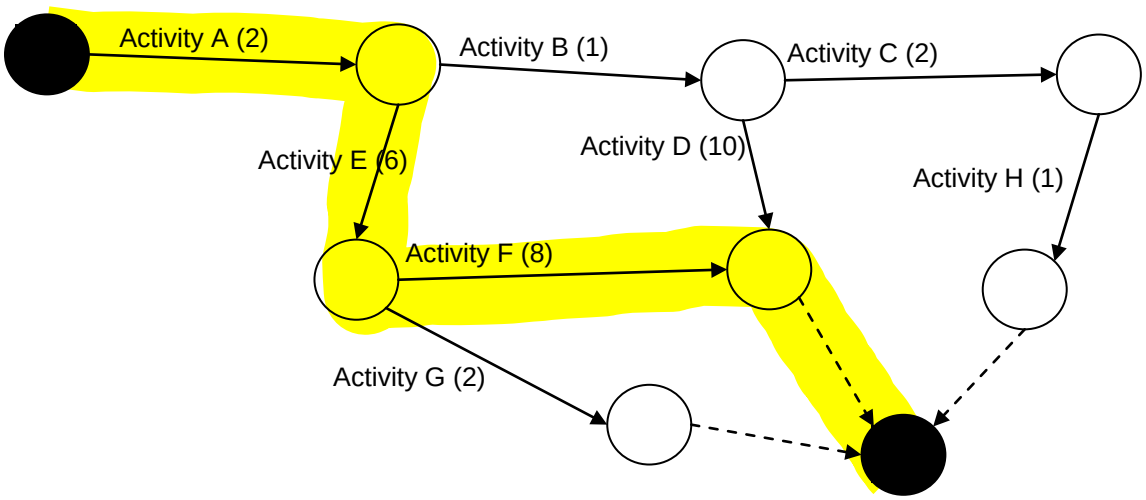
- 4 points for getting the right answer of 8.5
  - 1 point partial credit for showing work.

7c) How many person-months would this system take to implement?

50 / 8.5 = approximately 5.89 person-months

- 2 points for getting the right answer of 5.89
  - Can be rounded to 5.9
  - Can be rounded to 6 only if it is clear that the student used the right numerator and denominator in the calculation (since there are certain incorrect numerators and denominators that would also lead to an answer of 6)
  - 1 point partial credit for showing work

Question 8 (10pts): Suppose that your team builds the system by performing the activities shown in the graph below. All estimates of effort are shown in person-weeks.



8a) What activities are on the critical path?

Shown in highlight above, A-E-F (followed by null activity).

- 1 point for each of the three non-null activities in this path (3 points total)

8b) What is the slack time for Activity G?

The earliest that G can start is A + E = 2 + 6 = 8. But it doesn't need to finish until simultaneously with the end of F, so the latest start is A + E + F – G = 2 + 6 + 8 – 2 = 14. The difference is the slack, 14 – 8 = 6 person-weeks.

- 3 points for getting the right answer of 6
  - 1 point partial credit for showing work.

8c) What is the length of the critical path, in weeks?

The length of the critical path is the sum of its edges, A + E + F = 2 + 6 + 8 = 16

- 4 points for getting the right answer of 16
  - 1 point partial credit for showing work.