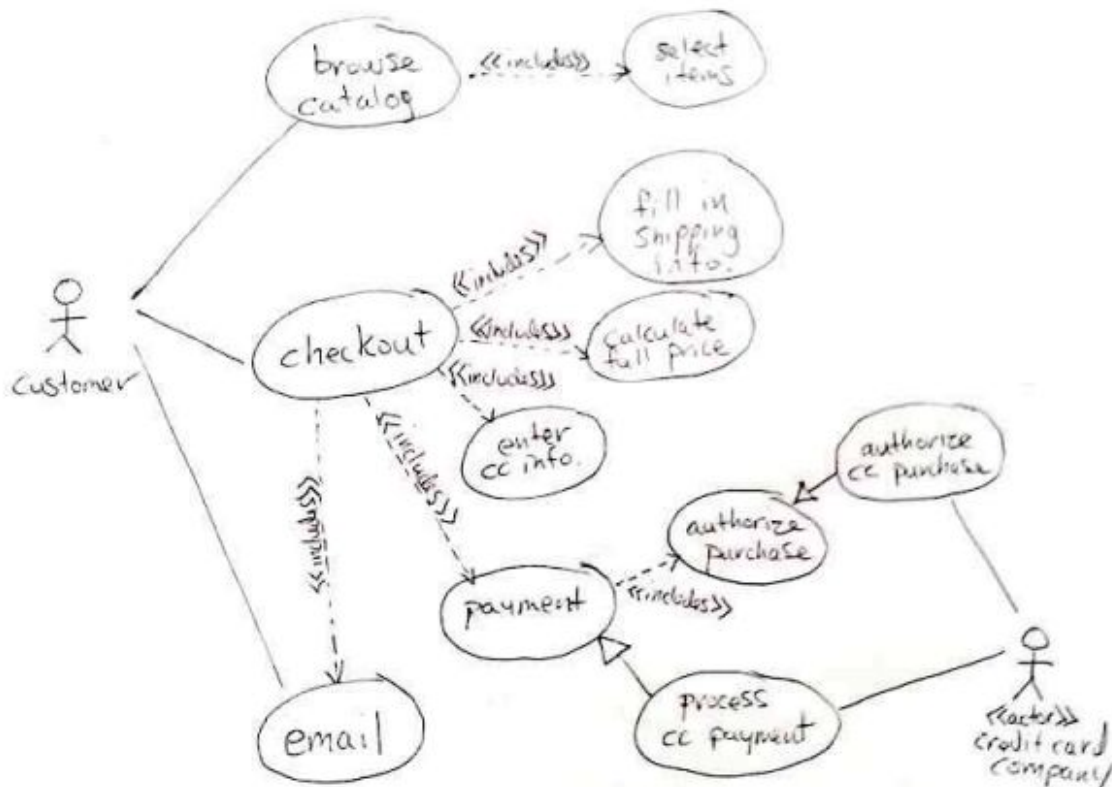


(b) [Use Case Diagrams - 10 marks] Draw a UML Use Case diagram for the detailed use case discussed in lecture below. Use page 9 (and the scratch paper at the back of this package, if necessary) to draw the Use Case Diagram.

Use Case - Buy a Product

1. Customer browses catalog & selects items to buy
2. Customer goes to check out
3. Customer fills in shipping information (address, next-day or 3-day delivery)
4. System presents full pricing information
5. Customer fills in credit card information
6. System authorizes purchase
7. System confirms sale immediately
8. System sends confirming email to customer

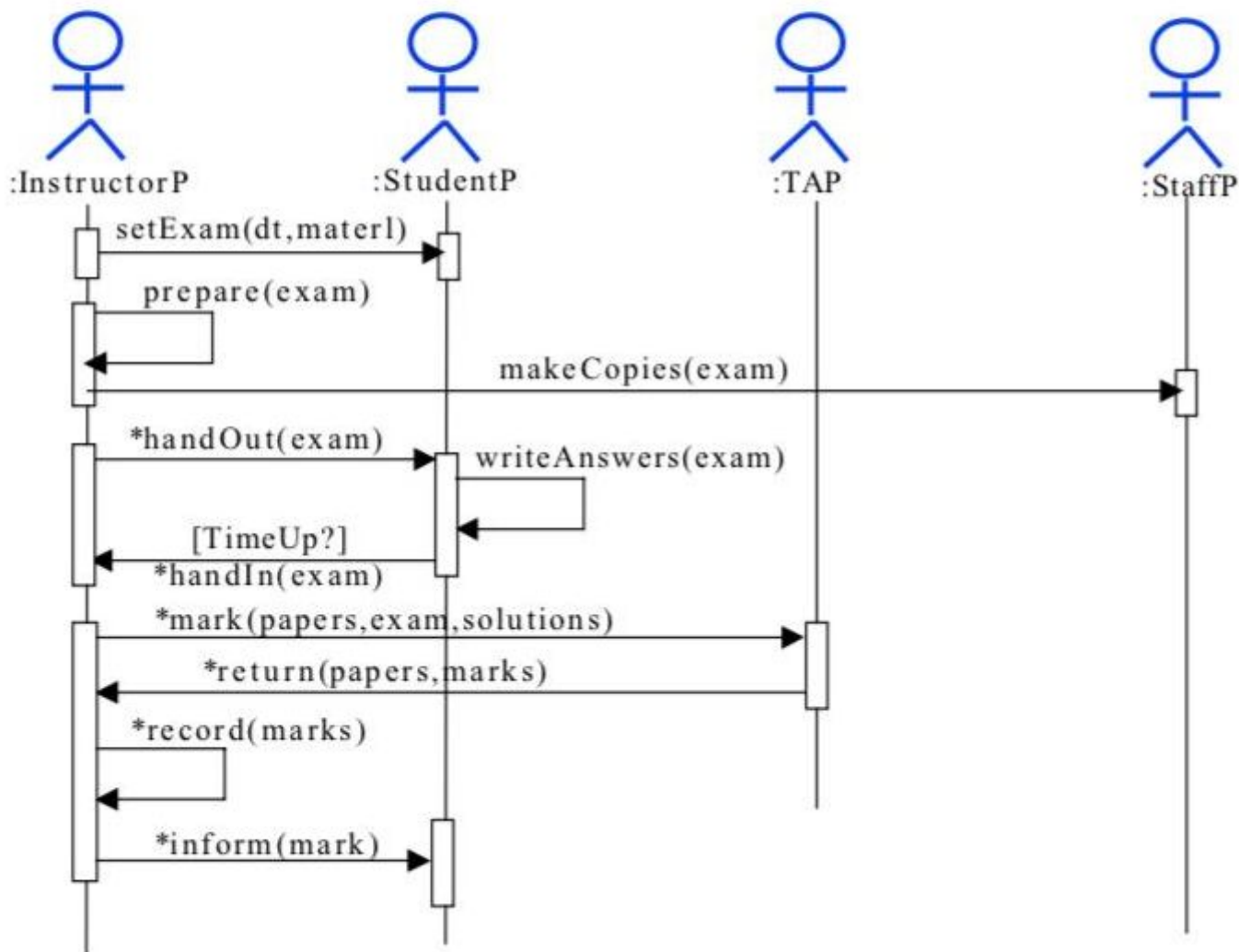
One possible solution below. There are many others.



Marks for: (1) completeness: all use cases listed (2) show all the <<includes>> for checkout (3) acknowledge the external <<actor>> representing the credit card company (or equivalent), and (4) optionally show some <<extends>> relationships, and generally (5) it makes some sense.

3. [Sequence Diagrams; 20 marks] To give an exam, an instructor first notifies the students of the exam date and the material to be covered. She then prepares the exam paper (with sample solutions), gets it copied to produce enough copies for the class, and hands it out to students on the designated time and location. The students write their answers to exam questions and hand in their papers to the instructor. The instructor then gives the exam papers to the TAs, along with sample solutions to each question, and gets them to mark it. She then records all marks and returns the papers to the students.

Draw a sequence diagram that represents this process. Make sure to show when is each actor participating in the process. Also, show the operation that is carried out during each interaction, and what its arguments are.

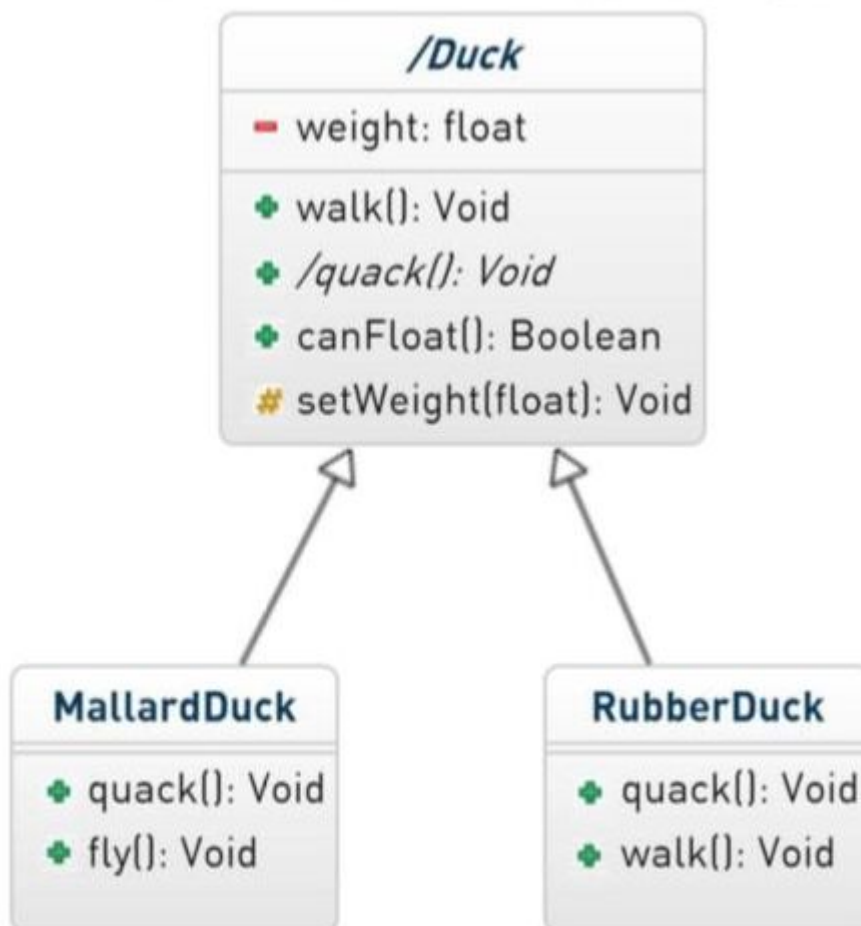


5. [10 points] Consider an abstract class Duck, representing all available ducks. Every duck can quack and walk, which are publically accessible functionalities of every duck. While every duck walks the same way, they quack differently which makes quacking an abstract feature of a duck. Every duck has a weight (float) which determines their ability to float (no pun intended), which ability can be checked by everybody through canFloat method. The weight of a duck can be set through a method available only for all concrete ducks implementations. A rubber duck, a kind of a duck, has all the features of a duck (yes, it quacks), but when asked to walk – it acts differently, because a rubber duck can't walk, so it overrides a duck walk feature and does nothing. An example of a duck is also a mallard duck that apart of quacking and walking, can also fly.

Draw a class diagram depicting classes, with names, attributes and methods along with proper relationship between classes. Skip constructors and destructors, but don't forget to annotate elements visibility using UML notation. For the actual attributes types or method signatures you can use Java or C++ notation, if you wish. Precede a name with a '/' symbol to indicate slanted (italicized) text.

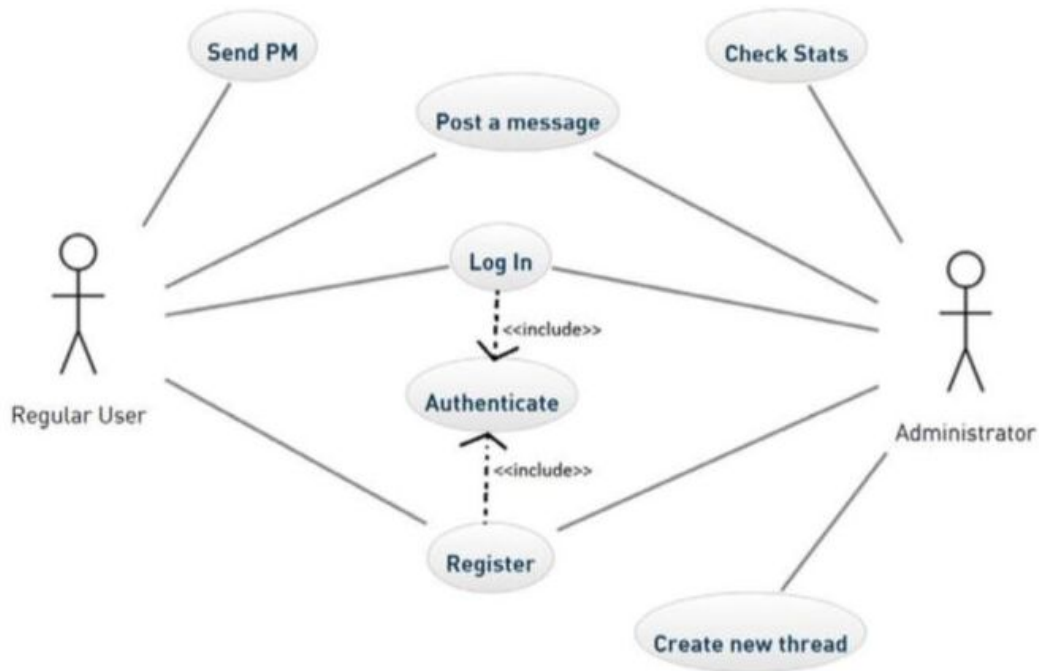
Use identifiers (class, attributes and method names) from the following list (you can repeat them if you need it.) (Hint: you should include all of them in your solution):

- Duck, MallardDuck, RubberDuck, walk, quack, fly, weight, canFloat, setWeight



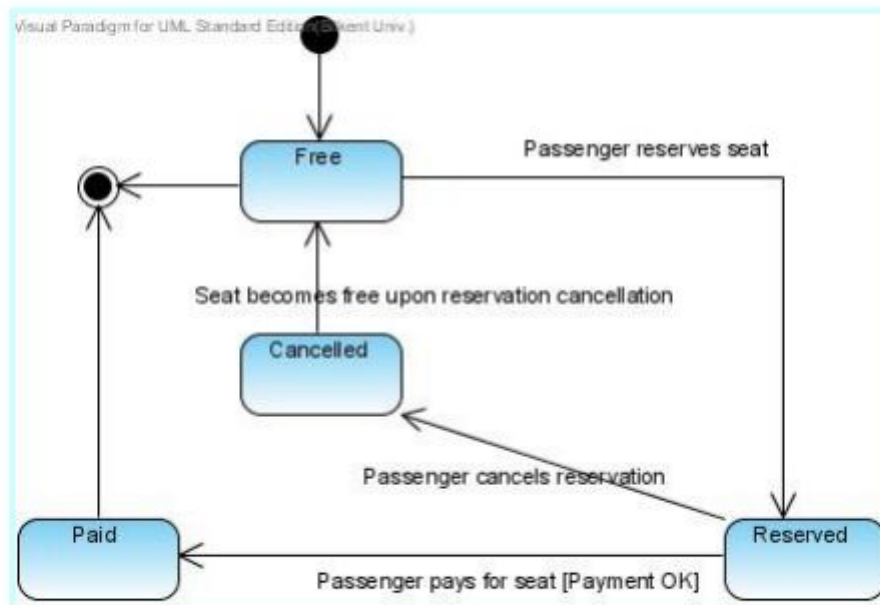
4. [10 points] Imagine that you are analyzing requirements for an online forum system. Forums can get very complex, but imagine that we have only two kinds of users that interact with our system with different responsibilities: Regular Users and Administrators. Both can log in to the system, and part of logging in is an internal authentication process. Both can also register with the system, which also uses internal authentication. After logging in, everybody can post new messages to the board, however only Administrators can check statistics and create new threads. Regular users on the other hand can send private messages to other users, while administrators do not have this ability.

Draw a Use Case diagram that contains Actors, Use Cases and their relationship from the scenario described above.



Seats can be reserved by customers on the web site of the bus company. The customer has the option to directly pay for the seat through the website. In that case, the seat cannot be cancelled (neither by the customer nor by the bus company). If the customer has not paid for the seat, the bus company can cancel the seat if the customer does not show up one hour before the trip. When the reservation is cancelled, the seat will become free and can be sold to another customer. Both the customer and the company staff must authenticate themselves for performing operations with the system.

f) [10 pts] Draw a state diagram for describing the details of the `Seat` object of the above system.



4. [Class Diagrams, 30 marks] Consider the world of libraries. A library has books, videos, and CDs that it loans to its users. All library material has a *id#* and a *title*. In addition, books have one or more authors, videos have one producer and one or more actors, while CDs have one or more entertainers. The library maintains one or more copies of each library item (book, video or CD).

Copies of all library material can be loaned to users. Reference-only material is loaned for 2hrs and can't be removed from the library. Other material can be loaned for 2 weeks. For every loan, the library records the user, the loan date and time, and the return date and time. For users, the library maintains their name, address and phone number.

Draw a class diagram (or two, if this is more convenient) for the description above. Make sure to show attributes, multiplicities and aggregations/compositions, where appropriate. No need to show any operations.

