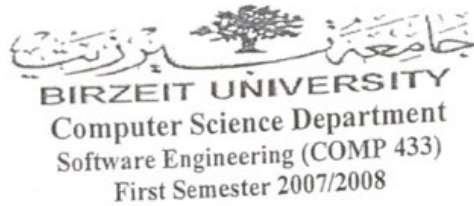


There are  
complex



39  
44  
83

25/11/2007

Midterm Exam

Question 1 [20 mark, 5 each]

- a) What is the traceability and what affect has on maintainability of the software.

Traceability is the ability to move forward and backward along the progress and development of the features. This means that a feature should be traceable backwards to the requirement so that we can the requirement of any given feature, and a requirement should be traceable forward to a ~~requirement~~ feature. Traceability makes the maintainability of the system (software) simpler and easier, because the maintenance people can know which requirement they are dealing with when looking at the features of the system and vice versa, which saves effort and therefore time and cost.

- b) Complexity is an inherited attribute of the software development, discusses the software complexity, in your discussion consider the role of modeling with dealing with software complexity.

Software complexity is inevitable in any software project. It is usually the result of the users' (customers') desire of having a system that would do many things and the developers' need of learning about a system they don't know much about. Complexity results in having too many irrelevant information and having one system do many jobs that can be separated into simpler and smaller systems. The role of modeling when dealing with complexity is omitting all the irrelevant information and breaking down the system into smaller and clearer subsystems.

(4)

4/20

1

- c) Differentiate between application domain and the solution domain, and what is/are the advantages that object oriented approach has over the structured approach.

The application domain is the domain stated by the user / client and which describes the current application and the desired features. The solution domain is stated by the developers and it describes the solutions that are offered by the new system. The object oriented approach is clearer and more organized than other approaches. It supports reusability and easy maintenance of the system and reduces and prevents redundancies.

- d) How is use-case diagramming related to functional modeling?

Use cases group the functional requirements related to each other and arrange them. Functional modeling is best illustrated and clarified using use-case diagramming because all the functional requirements in the functional model will be put in use-cases which are diagrammed in a use case diagram.

It is possible to say that functional modeling evolves into use case diagramming, and therefore can be traced backwards and forward.



Question 2 [25 mark]

Part One: Specify Which of these statements are functional which are non-functional. [10 mark, 2 each]

- a) The system should provide automatic verification of corrupted links or outdated data  
functional ✓
- b) An internal naming convention should ensure that records are unique  
nonfunctional ✓
- c) Communication between the database and servers should be encrypted  
nonfunctional ✓
- d) Files should be organizable into groups of file dependencies  
functional ✓
- e) The system must handle 50,000 users concurrently  
nonfunctional ✓
- (6/10)

Part Two: What is wrong with the following set of requirement and comment in their solution/s. [15 mark, 3 each]

- a) The system must be robust.  
the requirement isn't clear. Robust is not define. It should be something like this: Any crash in the system shouldn't affect any transactions except the last one at most.
- b) The system should use the binary search for all search operations based on numeric key.  
Algorithms are not defined in <sup>the</sup> requirements, they are left for later parts of the project. A requirement can state that the system should search for all search operations without stating how.
- c) The database should be distributed over many servers.  
Requirements don't deal with the technicalities of the system.  
Design decision
- d) The system must be produced at minimum cost  
Minimum cost should be more specific. For example: The budget of the project to develop the system should not exceed one million dollars.
- e) The system should be fast  
Fast is vague. It should be more clear and detailed: 20/25  
A user should get a response in 15 seconds in 30% of the time and in less than 30 seconds 100% of the time when using a processor of speed 100k/s with users on the system of maximum 1,000 users.
- (14/15)

Question 3 [10 mark, 5 each]

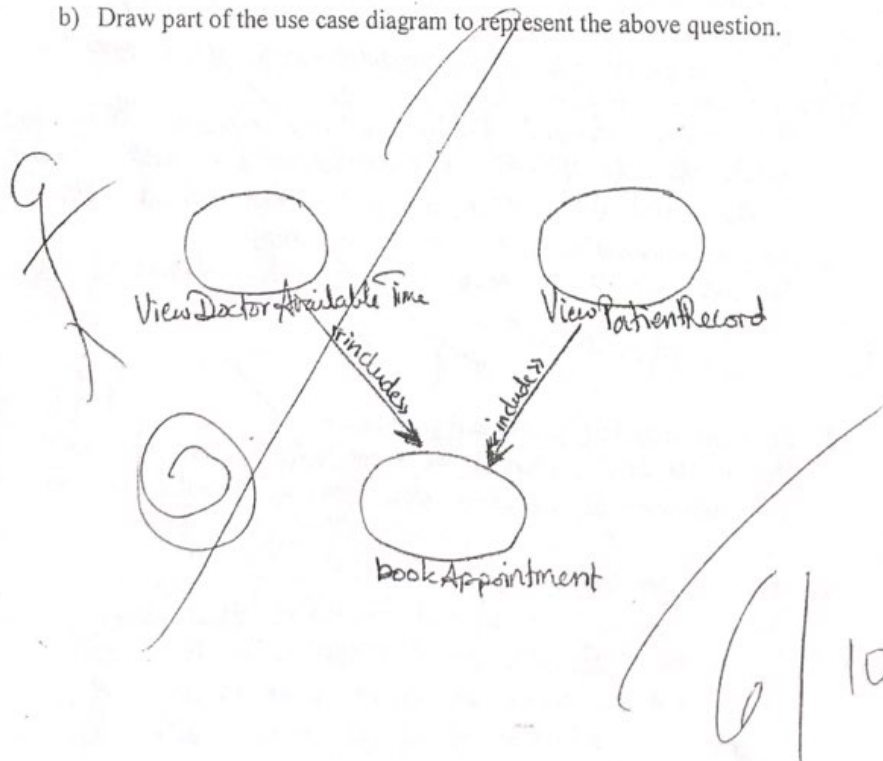
Assume the following:

- a system at the dentist office allows a patient to review his/her record when requesting an appointment to meet a dentist. The ViewPatientRecord feature provides a list of the past visits.
- the system retrieves a list of available time from which the patient can select one slot that suits him/her. The ViewDoctorAvalableTime feature retrieves a doctor's available time table which is divided into slots each slot 30 mins.

a) What is the relation that you should use to relate the bookAppointment with ViewPatientRecord and with ViewDoctorAvalableTime. Justify your answer.

3  
bookAppointment includes ViewPatientRecord, because according to the assumptions when a user is in the bookAppointment use case he could need the ViewPatientRecord use case from within the bookAppointment use case most of the time.  
bookAppointment also includes ViewDoctor AvailableTime use case because when a user is in the bookAppointment use case he will need access to the ViewDoctor AvailableTime use case from inside (without leaving the bookAppointment use case)

b) Draw part of the use case diagram to represent the above question.



6/10 4

39

Question 4 [45 marks]

Draw a use case diagram for an ATM machine with the following functionality. A customer's interaction with the system involves logging in, making withdrawal requests, deposit requests, and balance inquiry requests. If the customer logs in, there is a possibility (although rare) that the system will confiscate the card (for example, if the login is incorrect or if the bank determines that the customer's balance is significantly under zero). Card confiscation involves triggering an alert, which will be received by the branch manager of the bank. If the customer does a balance inquiry, this of course causes the system to check the balance. Checking the balance is also done when the customer requests a withdrawal, since the system must verify that enough money is in his account. In principle, checking the balance is a function that may be performed to support many use cases throughout the system.

- a) Write two scenarios for the customer to withdraw amount from the ATM. The first scenario the customer get the money, the second scenario the customer has entered a wrong amount and notified by the ATM, i.e the customer enters 50 JD and the accepted amount should be multiple of 20 JD note). [10 mark, 5 each]

Scenario 1:

1. Tom goes to the ATM machine and inserts his ATM card.  
2. The system asks TOM to log in.  
3. Tom enters his ATM pincode.  
4. The system verifies that the pincode is correct and displays a menu of options on the ATM machine.  
5. Tom requests making a withdrawal from his bank account.  
6. The system ~~displays~~ asks Tom to enter the amount of money he wants to withdraw.  
7. Tom enters the amount of money he wants and confirms.  
8. The system checks that the amount of money he entered was a multiple of 20 JD note and checks that this amount is available in his bank balance. The system then produces the required amount of money to Tom, deducts the amount from his balance, and ejects the ATM card and produces a receipt.  
9. Tom takes the ATM card, the receipt and the money and leaves.

Scenario 2:

1. Tom goes to the ATM machine and inserts his ATM card.  
2. The system asks Tom to log in.  
3. Tom enters his ATM pincode.  
4. The system verifies that the ~~pincode~~ <sup>Pincode</sup> is correct and displays a menu of options on the ATM machine.  
5. Tom requests making a withdrawal from his bank account.  
6. The system asks Tom to enter the amount of money he wants to withdraw.  
7. Tom enters 50 JD and confirms.  
8. The system checks Tom's balance and the amount he entered and finds that it is not a multiple of 20 JD note. The system displays an error message and asks Tom to enter another

b) Consider the scenario you have wrote in part a, identify the actors, then write the withdraw use case according to the template given in the lecture. [10 marks]

The actors in part a: Tom: customer.

Use case #: 01

Use case name: Withdraw

Participating actors: customer

Entry condition: The customer is logged in at the ATM machine.

Success condition: The customer has his money and the ATM card

Failure condition: - The amount of money isn't available in the balance  
- The amount of money isn't a multiple of 20 JD.

~~Pre-conditions:~~

Use case trigger: The customer makes a withdrawal request.

Flow of events: 1. The customer enters required amount of money  
2. The system checks ~~that~~ the balance  
3. The system checks if the amount is a multiple of 20 JD note.  
4. The system makes withdrawal and produces money, receipt and ATM card

Exceptions:

2.1 balance isn't enough to make withdrawal

3.1 amount isn't multiple of 20 JD

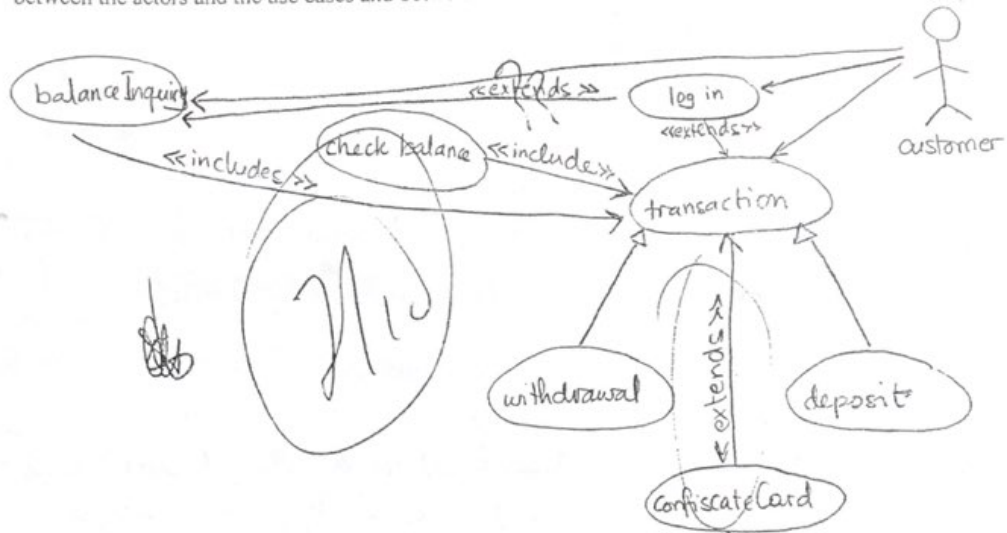
Actions

2.1.a display error message and return to menu

3.1.a display error message and return to menu

10

- c) Draw the use case diagram in which you show the actors, use case an relationship between the actors and the use cases and between use cases. [10 marks]



- d) From the use case event you have developed in part a and b identify a list of classes and their responsibilities. [5 marks]

- Customer
  - make withdraw request (trigger use case)
- Balance
  - stores the amount of money available in the balance.
- Balance check
  - checks if the amount of money entered is available in the balance.

- e) From the Use case in part b derive two functional requirements and two non-functional requirements (one usability and other efficiency). Your requirements should be clear and verifiable. [10 marks, 2.5 each]

### Functional Requirements

1. The system should allow the user to enter the amount of money he wants
2. The system should allow the user to return to the main menu before making a withdrawal.

### Nonfunctional Requirements:

#### - Usability:

There will be a help manual for every page display for the user which he can access.

#### - Efficiency:

The system should produce the ~~max~~ required amount of money in 15 seconds ~~at~~ in 80% of the time and in less than 30 seconds 100% of the time, given the the amount required is valid and available.