

Spring 2018 Midterm Exam March 16, 2018
CS 319 Object-Oriented Software Engineering

Instructors: Uğur Doğrusöz and Eray Tüzün

Reminders:

- Time: 150 minutes (2 hours and 30 minutes)
- Write your name and sign only in the last page as indicated.
- Show your work and reasoning clearly and write legibly, only within the space provided for each question. Do not detach any page(s).

Q1	10	
Q2	10	
Q3	15	
Q4	15	
Q5	50	
Total	100	

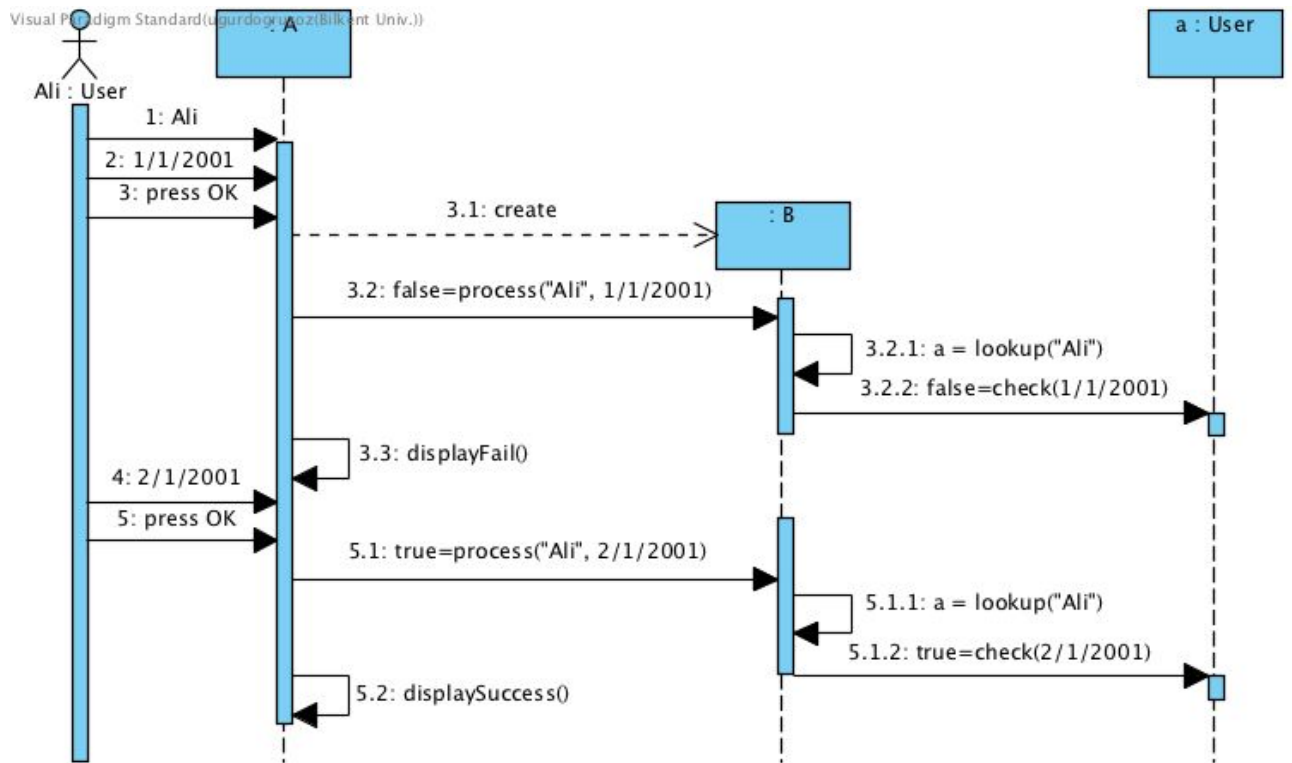
Q1) [10pts] Briefly list and explain the types of UML diagrams used in modeling the *dynamics* of a real life system during analysis.

Q2) [10pts] Consider the following description:

The CS department of SciCity University regularly arranges seminars for its students. A seminar is hosted by a faculty member. Upon proposal of a seminar by a faculty member, an abstract, title, and available time slots are requested from the speaker by the Department Secretary. Once this information is collected and the date is set, the seminar is said to be scheduled. Then the Secretary contacts the IT unit for having the seminar announced and enrollment opened on the Department web page. Once the enrollment count reaches a predetermined attendance limit (set depending on the room in which the seminar is scheduled), the seminar is considered full and the student is put into a waiting list. In cases, where a student changes their mind, they could set their preference through the web page accordingly. Any seat that becomes available is given to any student in the waiting list on a first-come first-served basis. Obviously a seminar might be cancelled for unforeseen reasons at any point.

Construct an appropriate UML diagram for expressing the various states of a specific seminar as described above, along with any events/conditions that trigger state transitions between these various states.

Q3) Study the following sequence diagram about a software system to be built.



- a. [5pts] Identify any actors, boundary, control, and entity objects involved in the scenario described by the above diagram.

- b. [10pts] Construct an object model with an appropriate UML diagram using the information in the above diagram.

Q4) [15pts] Please refer to the following description for the software engineering process life cycle.

Software engineering is performed in multiple iterations nowadays. Each mini-Waterfall is done with a partial set of requirements added until all requirements have been properly addressed. In each iteration, requirements are elicited with input from the customer. Observing the domain in its real-life setting, talking to potential users and the customer, and performing questionnaires are among the independent activities done during this stage. Only when the user agrees to all requirements elicited, we move on to the design phase, which is composed of a high-level system design, followed by a low-level object design. Once the design is complete, implementation and testing are performed. Modern software engineering praises testing while implementation takes place as opposed to waiting for the implementation to be complete. Typically each stage is performed by a different group of people in the development team: system analysts, design architects, coders and quality assurance (test) engineers, respectively.

Express the business logic of this process (from the viewpoint of development staff) using a UML activity diagram.

Q5) Following is what Ali imaged ever since he was a child as an ultimate adventure in outer space:

Ali is in outer space riding his spaceship. His spaceship is a rather peppy one with quick acceleration and maneuver abilities. It can shoot bullets as long as it can capture certain type of resources in the space. It gets damage as it hits meteors or gets hit by other enemy spaceships or their weapons. There are different types of resources, mostly composed of varying kind of bacteria, sitting in outer space. When his ship hits most of these resources, the bacteria in these resources will often help the ship slowly repair any damages encountered previously. Some rare type of bacteria will form a shield for the ship preventing any kind of attack by enemy ships and meteors. Another rare type of bacteria, however, will quickly result in corrosion damaging the ship instantly. Yet another type of resource hard to encounter in the space is jewelry. Jewelry are hard to collect since they move in random manner in the space. Playership can possess at most 2 shields at any time. Other spaceships come in varying size and shapes with varying hostility (some friendly others not), mobility (some are stationary) and damage power. Among the stationary spaceships, some will instantly provide fuel for up to 10 ships.

Ali's mission is to travel to planets with fun activities. When he arrives in such a planet (passes one level), he will take a break to rest and enjoy the particular activity in that planet. Activities include playing games (e.g. chess or tic-tac-toe) with and betting (with the valuable jewelry collected during the trip) against other human beings currently resting in the same planet.

Suppose you want to fulfill Ali's dreams and implement a desktop arcade game towards this goal named Shoot & More! (S&M). The game should have rather smooth graphics and will need to be a network game since the player will play different games (initially only chess and tic-tac-toe but could be extended with others in the future) in resting planets. The game is going to support pausing. When paused, the player may decide to save it to load later and continue without losing game progress. A high score list is also to be maintained for display at will (either when game is paused or as part of a menu before or after playing the game). If the game becomes a hit, you might need to port it to mobile platforms.

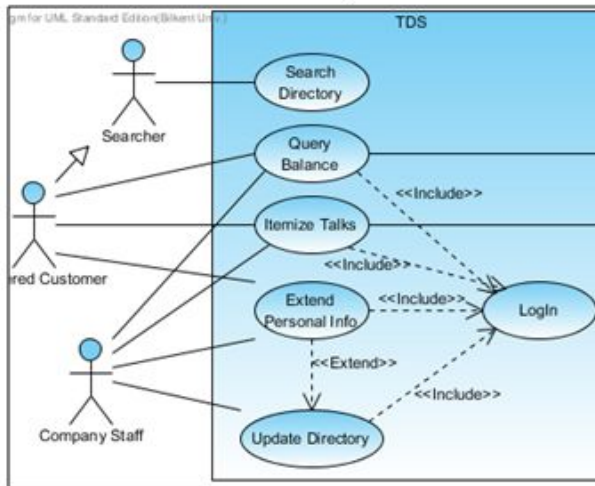
- a. [5pts] Identify and justify a couple of *conflicting* non-functional requirements for S&M, and briefly discuss how you would resolve the tradeoff if you were to develop this game.

- b. [10pts] Identify the use cases of S&M, and draw an appropriate UML diagram summarizing your findings. Make sure to specifically include all fun activities that a player can perform.

- c. [20pts] Perform an initial domain analysis to identify domain classes and their relationships for the objects in the adventures in Ali's mind; focus on travel in space and do *not* worry about activities at planets. Include any crucial control objects that will be needed for implementation but *ignore* any boundary objects. No need to be exact with method parameters or attribute types. Express your findings with an appropriate UML diagram.

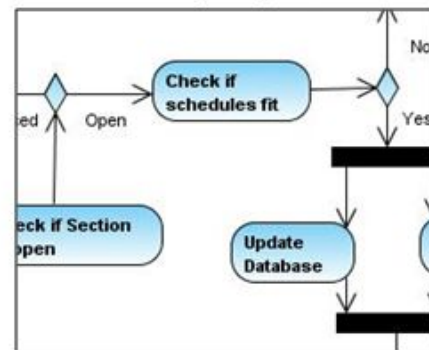
- d. [10pts] Draw a *simple* (e.g. a rectangular shape for a ship) mock-up as a sample screenshot from S&M *during* game play (travel in space, *not* during activities at planet stops).
- f. [5pts] Briefly discuss and justify what kind of an architectural style would be suitable for a high-level design of S&M.

Use Case Diagram

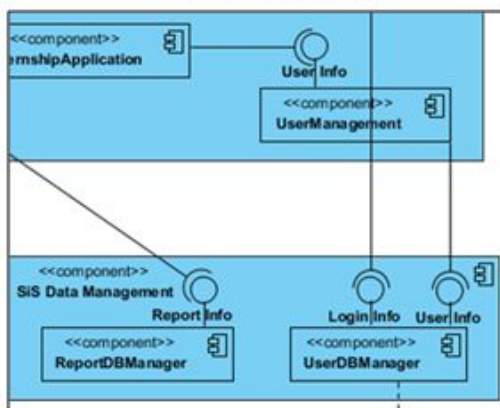


Sample UML Notation

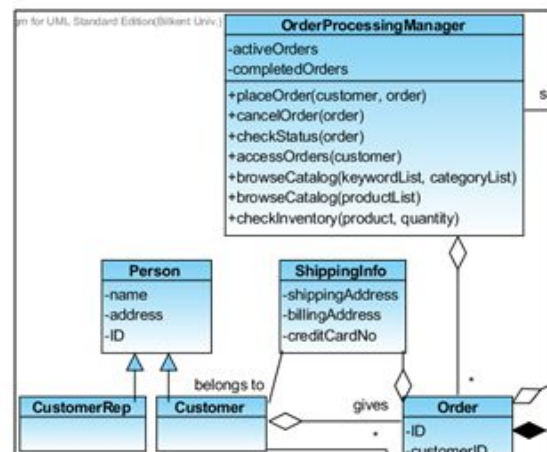
Activity Diagram



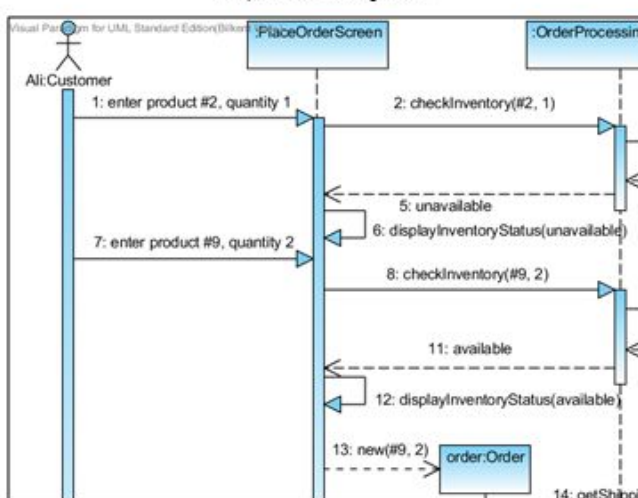
Component Diagram



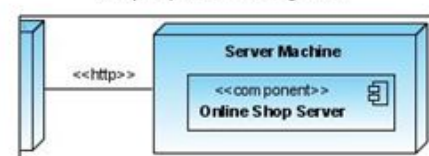
Class Diagram



Sequence Diagram



Deployment Diagram



Mini Dictionary

<i>Adventure</i>	Macera	<i>Assurance</i>	Güvence
<i>At will</i>	İstendiğinde	<i>To bet</i>	Bahis, iddaa
<i>Crucial</i>	Kritik	<i>To encounter</i>	Karşılaşmak
<i>Enrollment</i>	Kayıt	<i>Faculty member</i>	Öğretim üyesi
<i>To fulfill</i>	Yerine getirmek	<i>Hit</i>	Çok sevilen
<i>To host</i>	Ev sahipliği yapmak	<i>Hostility</i>	Düşmanlık
<i>Jewelry</i>	Mücevher	<i>Peppy</i>	Canlı, enerjik
<i>To praise</i>	Övmek	<i>Progress</i>	Gelişim, ilerleme
<i>Proposal</i>	Öneri	<i>To rest</i>	Dinlenmek
<i>Resource</i>	Kaynak	<i>Shield</i>	Kalkan
<i>Smooth</i>	Pürüzsüz	<i>Stationary</i>	Hareketsiz
<i>Towards</i>	-e doğru	<i>To trigger</i>	Tetikleme
<i>Unforeseen</i>	Öngörülemeyen		

Name & Signature: _____

Section: 1 ☐ ET2 ☐ UD