National University of Computer and Emerging Sciences

Islamabad Campus

Software Design & Analysis (CS-3004)

Date: (Monday) Nov 04, 2024

Course Instructors

Basharat Hussain, Javaria Imtiaz, Majid Hussain, Sidra Khalid

Sessional-II Exam

Total Time (Hrs): 1

Total Marks: 40

Total Questions: 4

Roll No _____ Section _____ Student Signature _____

Attempt all the questions on the provided answer sheet.
Don't write anything below this line on question paper.
IMPORTANT!: YOU WILL GET (01) BONUS MARK IF ATTEMPT ALL QUESTIONS IN ORDER.
Answer Mode is Displayed!

Question # 1 [Avg-attempt-minutes= 12, Marks = 5+5 = 10]

Scenario: Three entity classes are used in a collaboration CarSharer, Journey and Address. Each of these classes will be implemented by a (.java) source file. These classes are used across a number of use cases and are grouped together into a CarSharing component as Java (.class) files. here we are just dealing with the source files. There are two other classes MCSUserInterface and MCSControl. Each of these will be implemented by a (.java) file. The MCSControl component has a dependency on the CarSharing component and on the MCSUserInterface component.

(a) **Draw** a component diagram showing the source code dependencies. The .class files are grouped together into two Java archive (.jar) files. The MCSControl.class component will need to read a configuration file (MCS.ini) and display a help file (MCS.hlp) when required. The MCSControl (.class) file also has dependencies on the MCSUserInterface (.java) file and the CarSharing (.jar) components.

(b) **Draw** a deployment diagram given that the nodes are three client PCs, a server and a printer. The communications protocol between the clients and server is TCP/IP; and between the server and the printer is a standard parallel printer protocol. The user interface and the control objects will run on the clients.

Answer:

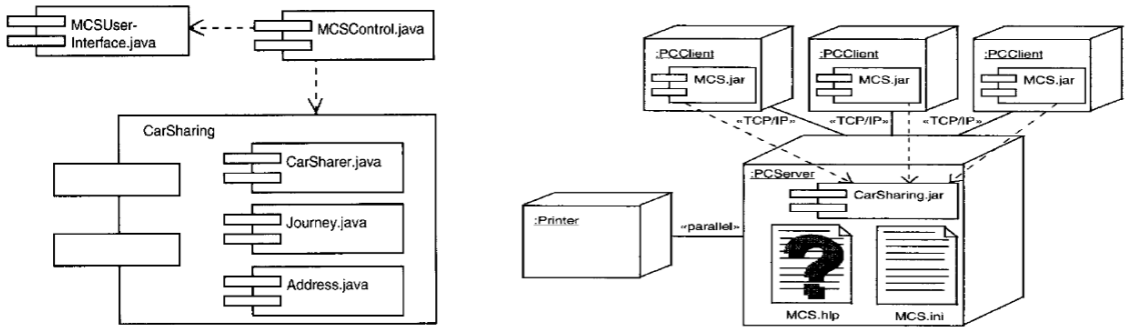

Figure 1: ...

Question # 2 [Avg-attempt-minutes= 20, Marks = 15]
Part: A (Class Diagrams) (4*2.5 = 10 Marks)

Scenario

The following class diagram shows part of the design of for software for EasyDVD, a web-based DVD rental agency. EasyDVD allows customers to browse its website to see the inventory of DVDs, and select DVDs for rent. DVDs are sent to the customer and returned by mail. Each DVD has a set rental fee and a set loan period.

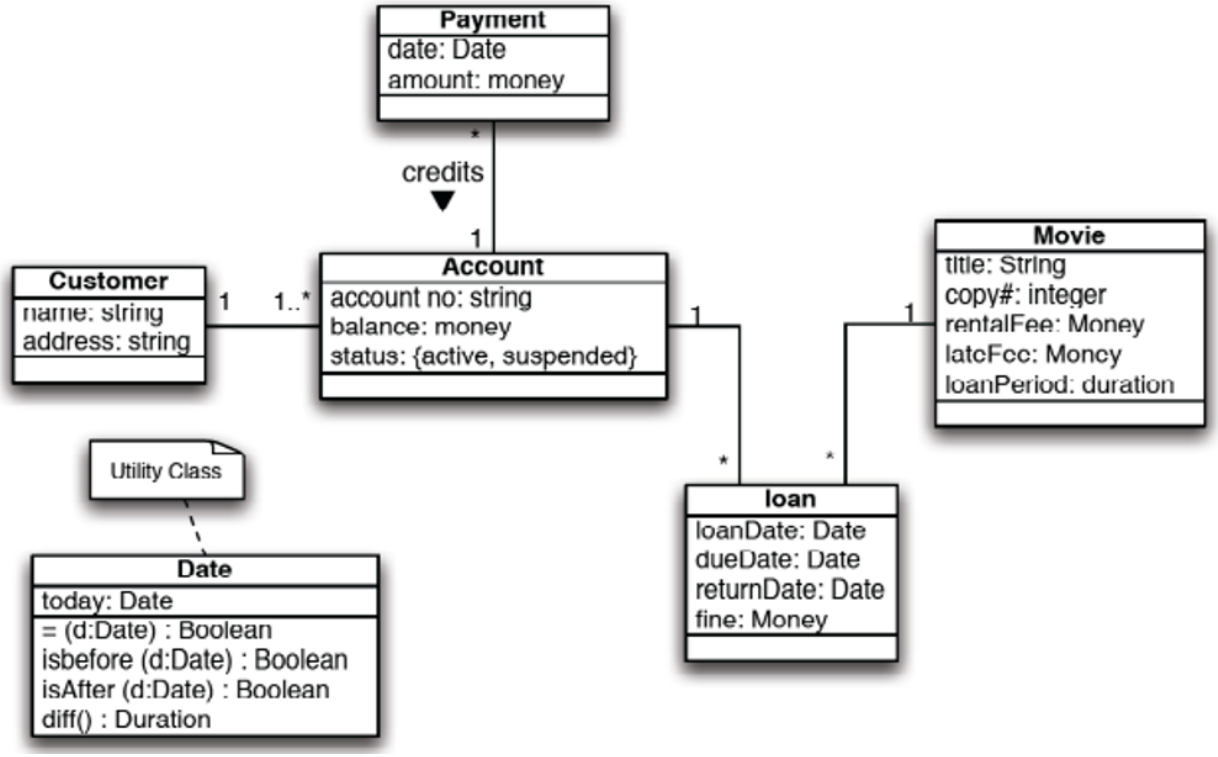

Figure 2: ...

(a) There are no associations shown for the Date class. How would you redraw the diagram to show such associations? What at the advantages and disadvantages of adding these explicitly? [2.5 marks]

(b) EasyDVD wants to restrict customers to have no more than 3 DVDs on loan at a time. How would you modify the diagram to show this constraint? Note: EasyDVD still needs to keep track of (a potentially large number of) past loans for each customers account. [2.5 marks]

(c) EasyDVD wants to simplify its pricing structure, so that there are a small number of categories of movies, where each category has a set rental fee and loan period. How would you modify the diagram to allow this? [2.5 marks]

(d) EasyDVD wants to add a new subscription service in addition to the existing ad hoc rental service. In a subscription service, customers maintain an ordered lists of movies they would like to rent, and whenever they return a movie, the next movie on the list is sent out. For the subscription service, customers pay a fixed monthly fee rather than individual rental fees on each movie, and each rental is open-ended i.e. there is no due date and no fine. How would you modify the class diagram to facilitate this service? [2.5 marks]

Answer:

Part A:

(a) You would need to add three associations from the Loan class to the Date class, labeled loadDate, dueDate, and returnDate, plus another association from the Payment class to the Date class. You would then remove these date attributes from the Loan and Payment class. Adding these to the diagram makes it more obvious how the date class is used, but makes the diagram more cluttered and harder to read.

(b) First, you would need to distinguish between current loans and past loans. A simple way to do this is to create two subclasses of the Loan class. The existing 1-to-many association from Account to Loan would then be moved to the pastLoans subclass, and a new association from Account to currentLoans, with 0..3 at the loan end.

(c) Add an additional class for movie types, which records the rental fee and loan period (and probably the late fee too?) and link this to the Movie class via a 1-tomany association. These attributes would need to be deleted from the Movie class:

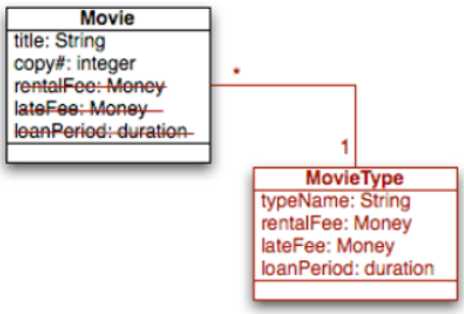

Figure 3: ...

(d) You would need to add a special type of account, which could be done by subclassing the Account class. Note that to keep a request list, we need to distinguish between movies (which are subject to requests) and the actual DVD, which is what gets loaned. Heres one possible solution:

Part: B (Sequence Diagrams) (5 Marks)

Sketch a UML Sequence Diagram for the process of renting a DVD for the EasyDVD service

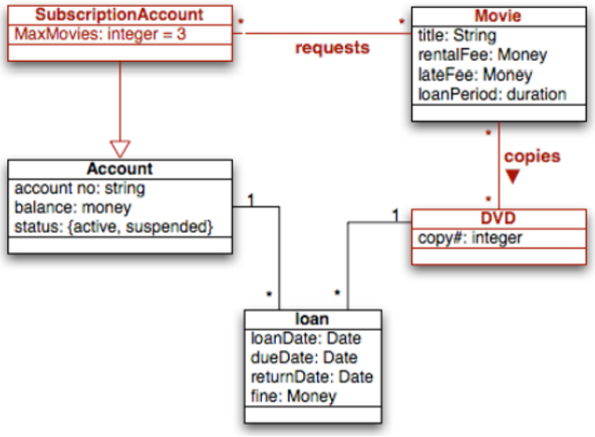

Figure 4: ...

described at the start of the previous question (i.e. the regular rental, not the subscription service). Be sure to show selecting a DVD to rent, and show how the system sets up a new loan with the appropriate dates set, using the Date class to calculate these.

Answer:

Part B:

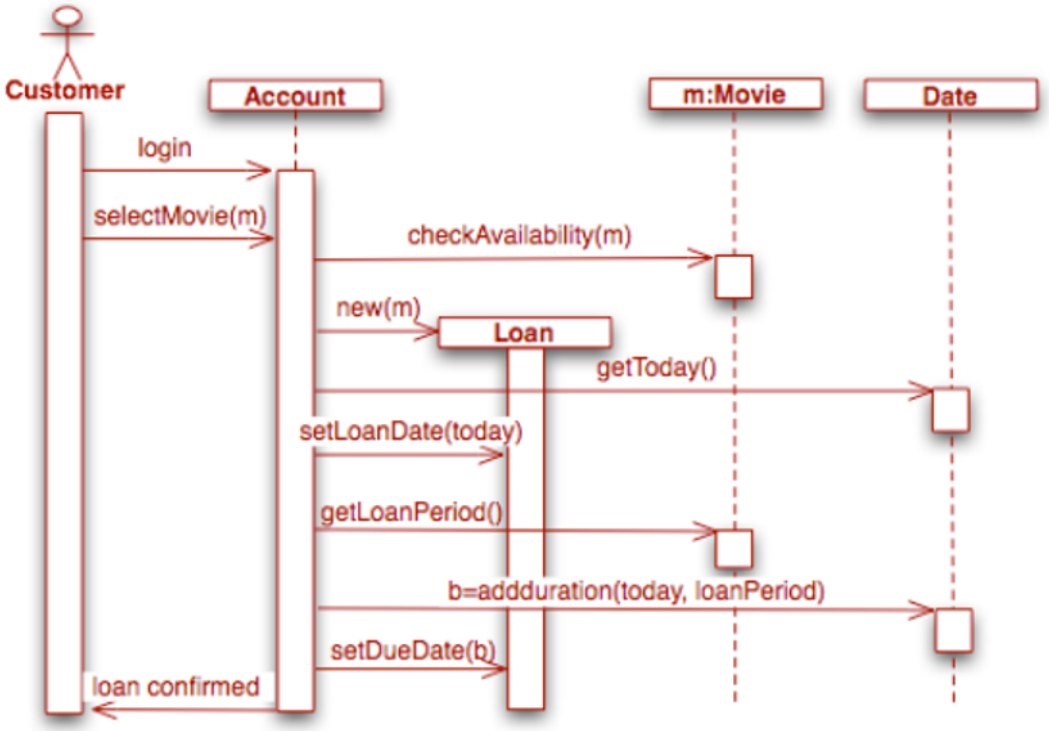

Figure 5: ...

Question # 3 [Avg-attempt-minutes= 12, Marks = 3+3+4=10]

Scenario: Imagine you have a Java-based POS (Point of Sale) system that simulates an enhanced sales process. The program starts by initializing a new sale and allows the user (acting as the cashier) to enter multiple items, one at a time, including their IDs and quantities. The cashier can continue entering items until they type "done" to indicate the end of item entry. After this, the system checks for available loyalty points for a customer, applies them if desired, and allows for the entry of a discount code to adjust the total. The sale then proceeds to a checkout stage where the final total is displayed, including any adjustments for loyalty points and discounts. The cashier selects a payment method, processes the payment, and the system generates a receipt before closing the sale.

(a) Draw an activity diagram that captures the flow of actions, including decision points.

(b) Draw a SSD (system sequence) diagram that captures the flow of actions.

(c) Write the java code of this SSD. Dont write the domain code except what you write in the main().

Answer:

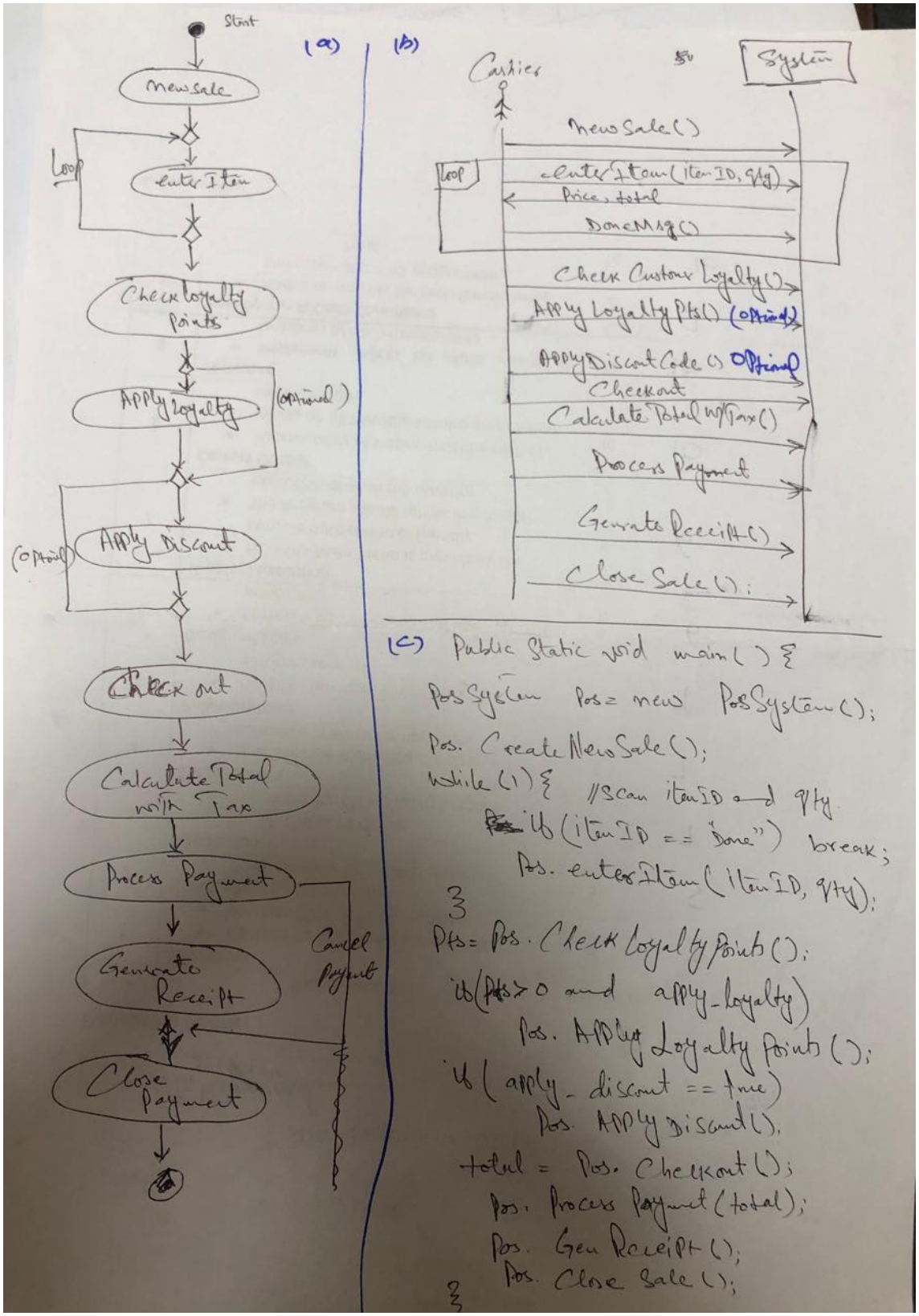

Figure 6: ...

Question # 4 [Avg-attempt-minutes= 10, Marks = 5]

Scenario: The below class diagram is partially designed for Bus reservation system to man-

age the trip Legs. Each leg has its own departure and arrival times, start and end locations, and possibly other specific details. A "leg of the trip" refers to a specific segment or portion of a larger journey. For example: If you are traveling from City A to City C with a stop in City B, your journey could have two legs:

Leg 1: City A to City B
Leg 2: City B to City C

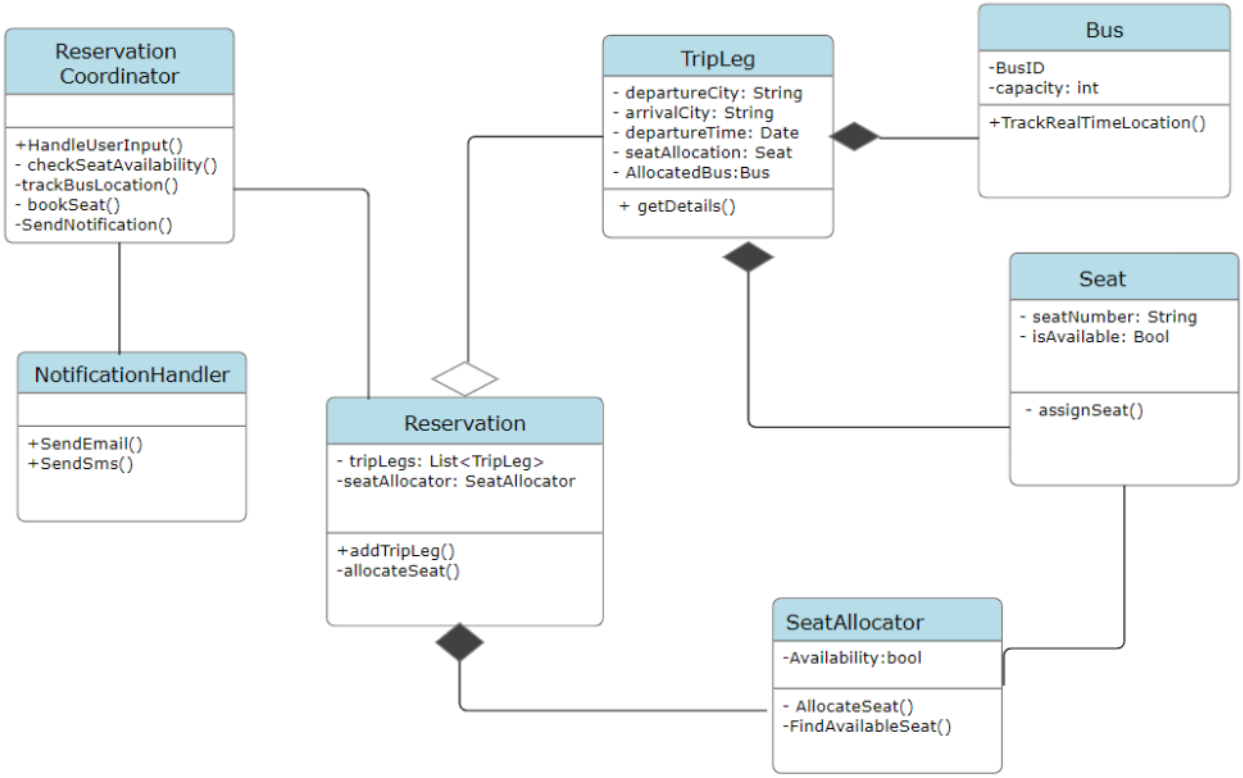

Figure 7: ...

Carefully analyze the Class diagram and attempt part a-e.

(a) **Which** class is responsible for handling the requests for making a reservation in the bus reservation system and can act as the Controller Class.

(b) **Which** class is the best candidate for Information Expert to determine and manage the departure and arrival city of a specific segment of a journey.

(c) **Which** class is an example of Pure Fabrication designed without representing a real-world entity?

(d) **Which** class is the most suitable candidate to be the Creator of TripLeg object?

(e) Which class is Information Expert for managing seat assignments and availability.

Answer:

(a) ReservationCoordinator

(b) TripLeg

(c) NotificationHandler

(d) Reservation

(e) SeatAllocator