National University of Computer and Emerging Sciences
Islamabad Campus

Software Design & Analysis (CS-3004)	Final Exam
Date: (Tuesday) Dec 24, 2024	Total Time (Hrs): 3
Course Instructors	Total Marks: 100
Basharat Hussain, Javaria Imtiaz, Majid Hussain, Sidra Khalid	Total Questions: 5

[redacted] Roll No [redacted] Section [redacted] Student Signature

YOU MUST READ THESE INSTRUCTIONS.

Instructions:

1. Attempt all the questions on the provided **answer sheet.**
2. **EXCEPT** for the multiple-choice questions (the last question), mark your answers directly on the question paper bubble sheet.
3. Tear off the ***LAST*** page only, **STAPLE it at the bottom or last of the answer sheet**, and submit it along with the answer sheet.
4. **Don't** write anything else on the question paper.

IMPORTANT! You will receive **01 BONUS MARK** if:

A. You **attempt all questions in order.**

B. You **attempt all subsections of all questions in order.**

C. You **do not leave any subsection unattempted.**

Additional Note: Start each new question (Questions 1 to 4) from a **new page** on the answer sheet.

Question # 1 [Avg-attempt-minutes= 30, Marks = 7+7+6 = 20]

(a) Discuss the implementation of the **Observer pattern** for a real-time "stock price notification system". Include a brief explanation of its working with a class diagram clearly describing the method calls. Don't write the actual Java code (7 marks). [key components and working: 4 marks, Class Diagram: 3 marks]

(b) Explain how the **Strategy pattern** allows for dynamic behavior changes in an application. Design a strategy-based solution for a text formatter that can format text in different styles (e.g., plain-text, HTML, and Markdown). Include a class diagram. Write the Java code to illustrate your **Context** class only. Don't write extra code. (7 Marks) [overview and class diagrams: 4 marks, context class proper Java code: 3 marks]

(c) Provide a UML class diagram or code example to demonstrate the implementation that bridges the gap between incompatible interfaces. Compare the this pattern with the Facade pattern. When would you prefer this over Facade? (Please don't write any Java Code) (6 marks) [correctly specify pattern: 1 mark, class diagram: 2 marks: comparison: 3 marks]

Question # 2 [Avg-attempt-minutes= 40, Marks = 2

Part: A (Conference management system) (7 Marks)

Scenario

The process of the candidates is to login the conference system and submit the conference paper through online. The system verifies the user's credentials through the authentication module. Then the reviewer reviews the paper and sends the acknowledgement to the candidate either paper selected or rejected. This process of on conference management system are described sequentially through following steps.

* The candidate login to the conference management system.
* The paper title is submitted.
* The reviewer sends acknowledgement to the candidate.
* Based on the selection, the best candidate is selected.
* Finally the candidate registers all details.

Consider above conference system includes a Web server and two database servers. Both database servers are identical: The first acts as a main server, while the second acts as a redundant backup in case the first one fails. Users use Web browsers to access data through the Web server. They also have the option of using a proprietary client that accesses the databases directly.

(a) Draw a package diagram for above system? [2 Marks]

(b) Draw component diagram for above system? [2 Marks]

(c) Draw a UML deployment diagram representing the hardware/software mapping of this system? [3 Marks]

Part: B (Employee Management System) (6 Marks)

Scenario

You are tasked with creating a JavaFX application connected to a MySQL database for managing employee records. The application should allow the user to: View all employee records in a Table View. Add a new employee using a form (Text Field for Name, Combo Box for Department, and Text Field for Salary). **Read the below code and write the code for missing lines (1 to 6).**

```
public class EmployeeManagementApp extends Application
{
 private TableView<Employee> employeeTable = new TableView<>();
 private ObservableList<Employee> employeeData = FXCollections.observableArrayList();

  1. Create TextField for Name?          >>> Write your code here
  2. Create ComboBox for Department?     >>> Write your code here
  3. Create TextField for Salary?        >>> Write your code here
}

// Load Employees
```

```
private void loadEmployees() {
employeeData.clear();
try (Connection conn = connect()) {

    4. String query =?      >>> Complete your code here
    5. Statement stmt =?    >>> Complete your code here
    6. ResultSet rs =?      >>> Complete your code here
    while (rs.next()) {
        employeeData.add(new Employee(
        rs.getString("name"),
        rs.getString("department"),
        rs.getDouble("salary")
        ));
    }
} catch (SQLException e) {
    e.printStackTrace();
}
}
```

Part: C (7 Marks)

Scenario

A customer places a preliminary order with the sales department. The sales department checks the order for accuracy and obtains the availability of order items from the warehouse. The customer is informed about the availability of items and he can decide whether to proceed with the order or to cancel it. If the customers decision is to cancel the order, then the sales department will cancel it. Otherwise, the sales department will bill the customer while the warehouse prepares the order items for delivery. The customer is then asked to pay the bill. Usually the full amount should be paid, but if the customer is creditworthy, then he can make a partial payment. For partial payments, the sales department would get the managers approval. If the manager rejects approval, then the sales department will cancel the order. For fully paid bills, the sales department will close the order and the warehouse will deliver the items. For partially paid bills, the sales department will create a partially paid order and the warehouse will deliver the items.

(a) For above food ordering scenario the following diagram shows an activity diagram created for the above scenario. Without making any assumptions, identify the activities A to G accurately from the activity list given below the diagram and fill in the blanks. Please note that the same activity may appear more than once in the diagram.

Activities:

i Approve Partial Payment
ii Inform Customer
iii Cancel Order
iv Check Order
v Create Partially paid Order
vi Bill Customer
vii Prepare Items

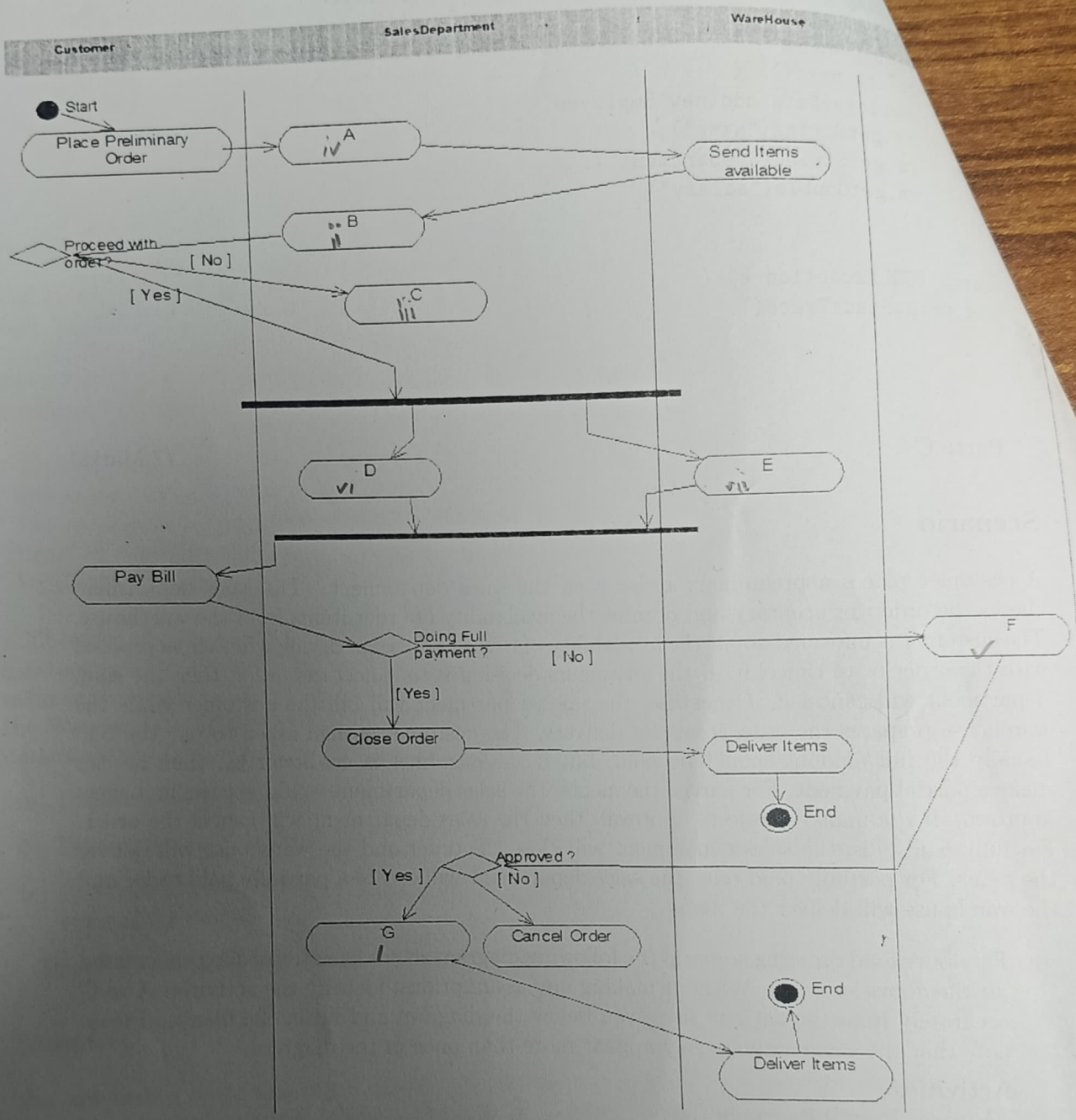


Figure 1: ...

Activity	Option
i	(write your answer on answer sheet, write A, B or C ...)
ii	(write your answer on answer sheet)
iii	(write your answer on answer sheet)
iv	(write your answer on answer sheet)
v	(write your answer on answer sheet)
vi	(write your answer on answer sheet)
vii	(write your answer on answer sheet)

Question # 3 [Avg-attempt-minutes= 40, Marks = 20]

Part: A (07 Marks)

Scenario

The following domain model captures some basic information about a kids hockey league. In answering the following questions, state any assumptions that you make.

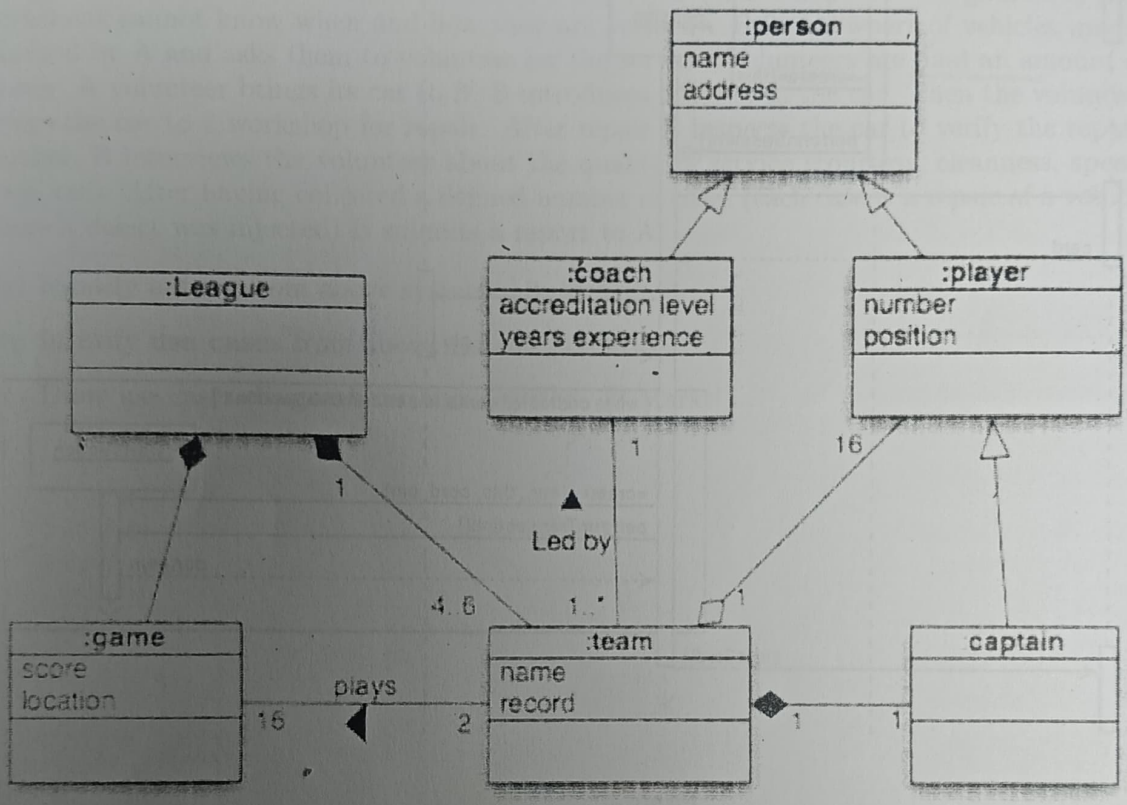


Figure 2: ...

(a) How many games will a captain play in?

(b) What is the maximum number of games that any given coach can be involved in?

(c) In the model, the relationship between player and team is shown as an aggregation, while that between captain and team is shown as a composition. Is this a good choice?

(d) The league actually has two types of game: regular season games and playoffs. Each team plays 4 regular season games against each of the other teams. Every team also gets to play in the playoffs, which are conducted as a knockout a team is out of the playoffs when it loses a playoff game. How would you modify the model to capture this additional information?

Part: B (6 Marks)

Write a piece of Java code for **Session** and **Transaction** classes.
(Note: Please don't write any code for other classes or main function)

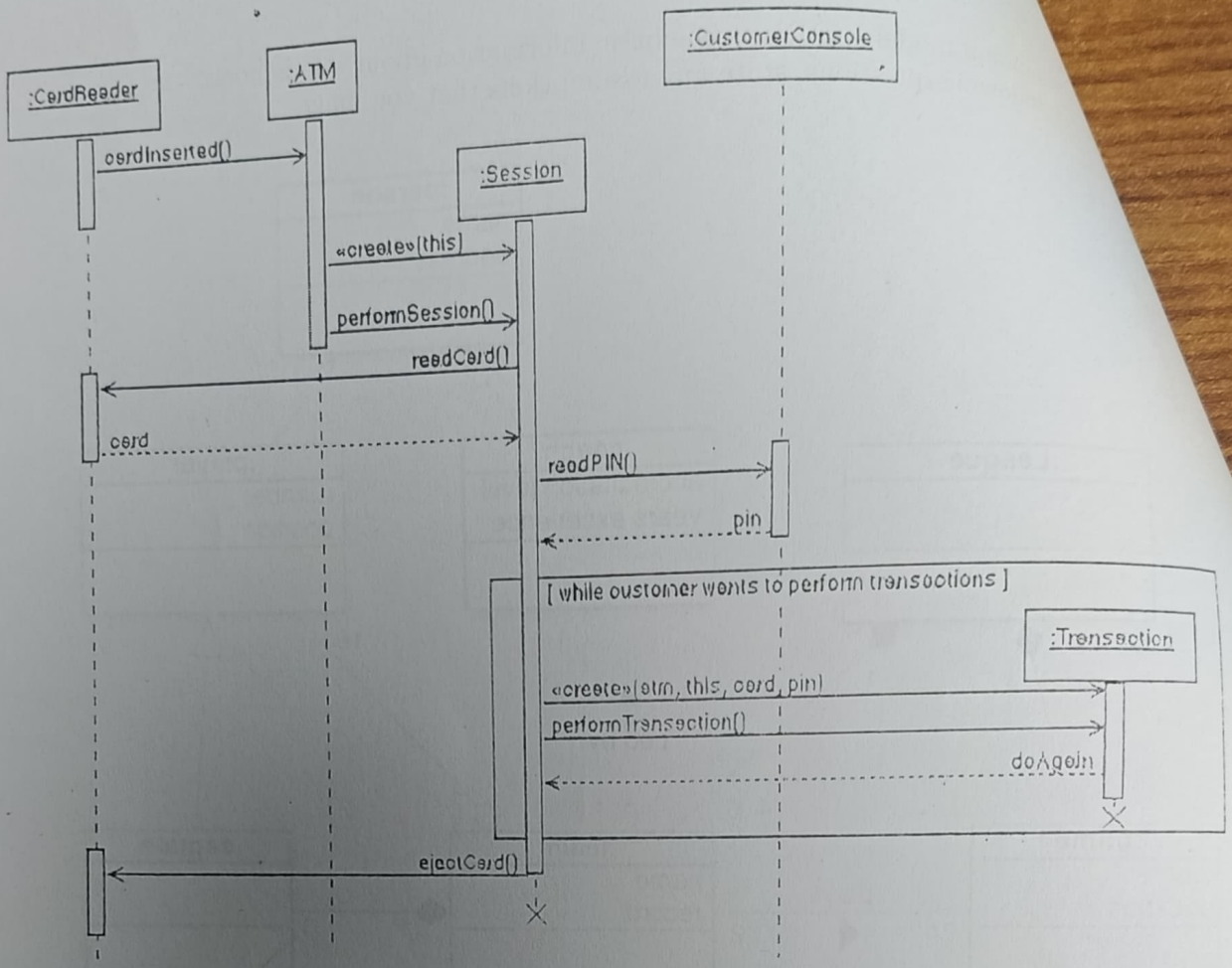


Figure 3: ...

write your code on the answer sheet.

Part: C (7 Marks)

Scenario

The administrator enters the patients name, address, date of birth and emergency contact details into the system. If the patient has only public health insurance, the administrator enters the patients medicare number, and the system verifies this with government health database. If the patient also has private health insurance, then the administrator enters also the patients private health insurance details, and the system verifies these details with the private health insurance system. When these details are verified as correct, the system saves the patient's details and confirms the registration.

(a) Identify the actors and the objects in the following scenario to register a patient in a hospital management system

(b) Sketch a UML sequence diagram for the scenario where the administrator registers a patient who only has public health insurance.

Question # 4 [Avg-attempt-minutes= 15, Marks = 10]

Scenario: **Read the case study below and answer part a, b and c.**

Automotive companies need to verify the quality of service of workshops doing maintenance service on the vehicles. Workshops belong to companies that are independent of the automotive company, and are spread geographically. Phantom test analyzes the quality of service with special care in not making aware workshops that they are tested. Verification of the quality of service is made as follows. Assume that the Automotive company is A. A subcontracts the verification to a specialized company B (this is important to guarantee that workshops cannot know when and how they are verified). B finds owners of vehicles manufactured by A and asks them to volunteer for the service. Volunteers are paid an amount of money. A volunteer brings its car to B. B introduces a defect in the car. Then the volunteer brings the car to a workshop for repair. After repair B inspects the car to verify the repair. Further, B interviews the volunteer about the quality of service (courtesy, cleanness, speed, price, etc). After having collected a defined number of cases (each case is a repair of a vehicle where a defect was injected) B submits a report to A.

(a) Identify **actors** from above system. [3 marks]

(b) Identify **use cases** from above system. [4 marks]

(c) Draw use cases **diagram** for above system. [3 Marks]

Question # 5 [Avg-attempt-minutes= 30, Marks = 2

Attempt All the Questions Below on Bubble Sheet at *LAST* Page
Tear off the *LAST* page only, STAPLE it at the bottom of the answer sheet

Q. 1 Which of these are types of nodes used in the deployment diagram?

a) Device

b) Execution Environment

c) Artifact

d) Device & Execution Environment

Q. 2 Consider the following statements in relation to Activity diagrams.

i) They can be used to show the activities involving different participants, such as different groups or roles in a system.

ii) In an activity diagram, there will be only one ending activity.

iii) In an Activity diagram, a synchronization bar specifies the activities, which cannot be done concurrently.

Which of the above statements is/ are true?

a) Only (i).

b) Only (ii).

c) Only (iii).

d) Only (i) and (iii).

e) All.

Q. 3 Consider the following functions.

i) Update Membership record

ii) Archive Membership

iii) Register New Member

Which of the actors is/are responsible for the above functions?

a) Customer Service Assistant

b) Order Processing Clerk

c) Inventory Control Clerk

d) Mail Order System

e) Customer

Q. 4 Which interface is used to execute SQL queries in JDBC?

a) Connection

b) ResultSet

c) Statement

d) PreparedStatement

Q. 5 Which of the following is used to create a connection to a database?

a) Driver.getConnection()

b) DriverManager.getConnection()

c) ConnectionManager.connect()

d) DatabaseManager.connect()

Q. 6 Which of the following is NOT a node in JavaFX?

a) Button

b) Label

c) TableView

d) Scene

Q. 7 What is the purpose of the High Cohesion pattern in GRASP?

a) To reduce dependencies between classes

b) To distribute responsibilities evenly within a class

c) To keep related functionality together in a single class

d) To ensure only necessary relationships exist between classes

Q. 8 To realize the real power of dynamic binding

a) A superclass reference must be a concrete class

b) A superclass reference must be an interface

c) A superclass reference must be an abstract class

d) A superclass reference can be any type: interface, abstract class, or concrete class

Q. 9 Composition is a form of ———— where the parts are removed when the whole is deleted.

a) Abstraction

b) Association

c) Aggregation

d) Inheritance

Q. 10 In Java, declaring a class abstract is useful when

a) To prevent developers from further extending the class

b) When it doesnt make sense to have objects of that class

c) When default implementations of some methods are not desirable

d) To force developers to extend the class not to use its capabilities

Q. 11 Polymorphism is achieved in Java using

a) Inner class

b) Anonymous classes

c) Method overloading

d) Method overriding

Q. 12 In object oriented analysis, Behavior of a real world concept is reflected as a

a) Object

b) Methods

c) Attributes

d) Class

Q. 13 ———— diagram is time-oriented?

a) Collaboration

b) Sequence

c) Activity

d) None of the mentioned

Q. 14 Which of the following is true of a class diagram?

a) It is completed after the system is built.

b) It is a requirement for all object-oriented design.

c) It can be used as a blueprint to build a system.

d) It does not show relationships or dependencies.

Q. 15 In the diagram below, what type of association exists between Tree and Deciduous?

a) Composition

b) Aggregation

c) Association

d) Generalization

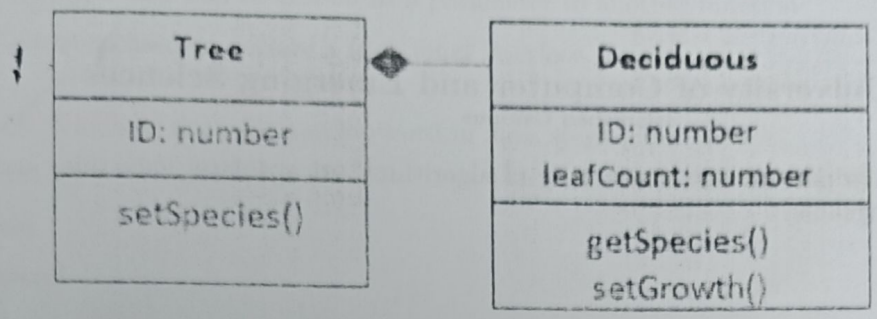


Figure 4: ...

Q. 16 In a package diagram, what does a dependency arrow indicate?

a) One package is a subtype of another

b) One package uses or depends on another

c) One package is contained within another

d) One package is equivalent to another

Q. 17 Which of the following design patterns ensures that a class has only one instance and provides a global point of access to it?

a) Singleton

b) Factory

c) Observer

d) Strategy

Q. 18 Which design pattern allows a class to delegate responsibility to another class by adapting its interface?

a) Adapter

b) Facade

c) Singleton

d) Strategy

Q. 19 In which design pattern do you provide a simple interface to a complex subsystem by delegating requests to a set of objects in the subsystem?

a) Facade

b) Factory

c) Observer

d) Singleton

Q. 20 Which design pattern defines a family of algorithms, encapsulates each one, and makes them interchangeable?

a) Strategy

b) Factory

c) Adapter

d) Singleton

Q. 21 The `protected` access modifier allows visibility to:

a) The same class only.

b) The same package and subclasses.

c) Subclasses only.

d) The same package only.

Q. 22 Which of the following is NOT an example of polymorphism in Java?

(a) Method overriding in a subclass.

(b) Implementing multiple interfaces in a class.

(c) Constructor overloading in a class.

(d) Using lambda expressions to define anonymous functions.

Q. 23 In the GRASP pattern of Information Expert, which of the following is a criterion for deciding which class should be assigned responsibility for handling an operation?

a) The class that is closest to the client

b) The class that contains the data needed for the operation

c) The class that performs the most computations

d) The class that handles user interface interactions

Q. 24 In the GRASP pattern, which class is responsible for receiving the input from the external world and delegating to other classes to process the request?

a) Creator

b) Information Expert

c) Controller

d) Pure Fabrication

Q. 25 Which of the following best describes a lambda function in Java?

a) A method that is automatically invoked at runtime

b) A concise way to define an anonymous class

c) A function that can be passed as a parameter to another function

d) A keyword used to declare a functional interface

Q. 26 Which of the following keyword in Java stops inheritance?

a) abstract

b) final

c) extends

d) const

Q. 27 Which of the following diagram types is used to represent the behavior of a system?

a) Use case diagram

b) Sequence diagram

c) Class diagram

d) Activity diagram

Q. 28 Which of the following is a characteristic of an interface in UML?

a) It has implementation details

b) It can be instantiated as an object

c) It specifies a contract for a set of operations

d) It represents a physical component of the system

Q. 29 In Object-Oriented Programming (OOP), which principle is used to restrict access to certain components of an object?

a) Encapsulation

b) Abstraction

c) Inheritance

d) Polymorphism

Q. 30 Which of the following is a correct syntax for a lambda expression that takes two integers and returns their sum?

a) `(a, b) → { return a + b; }`

b) `(a, b) ⇒ a + b;`

c) `(a: int, b: int) → a + b;`

d) `(int a, int b) ⇒ { return a + b; }`

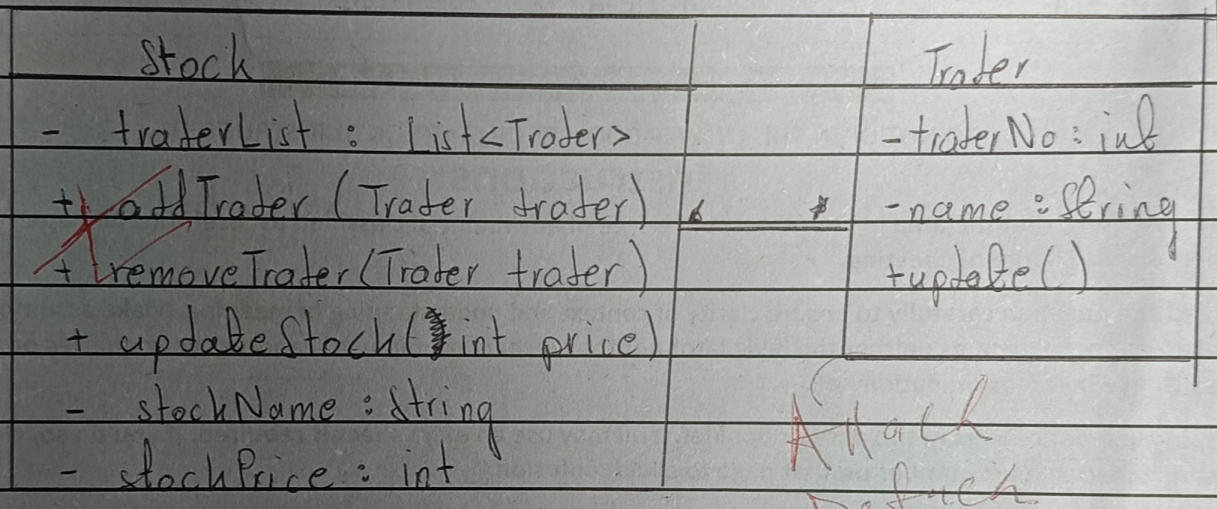
END OF QUESTIONS.

Q.1

(a)

- Stock Price Notification System

The stocks will be the subjects & the stock traders will be the observers.

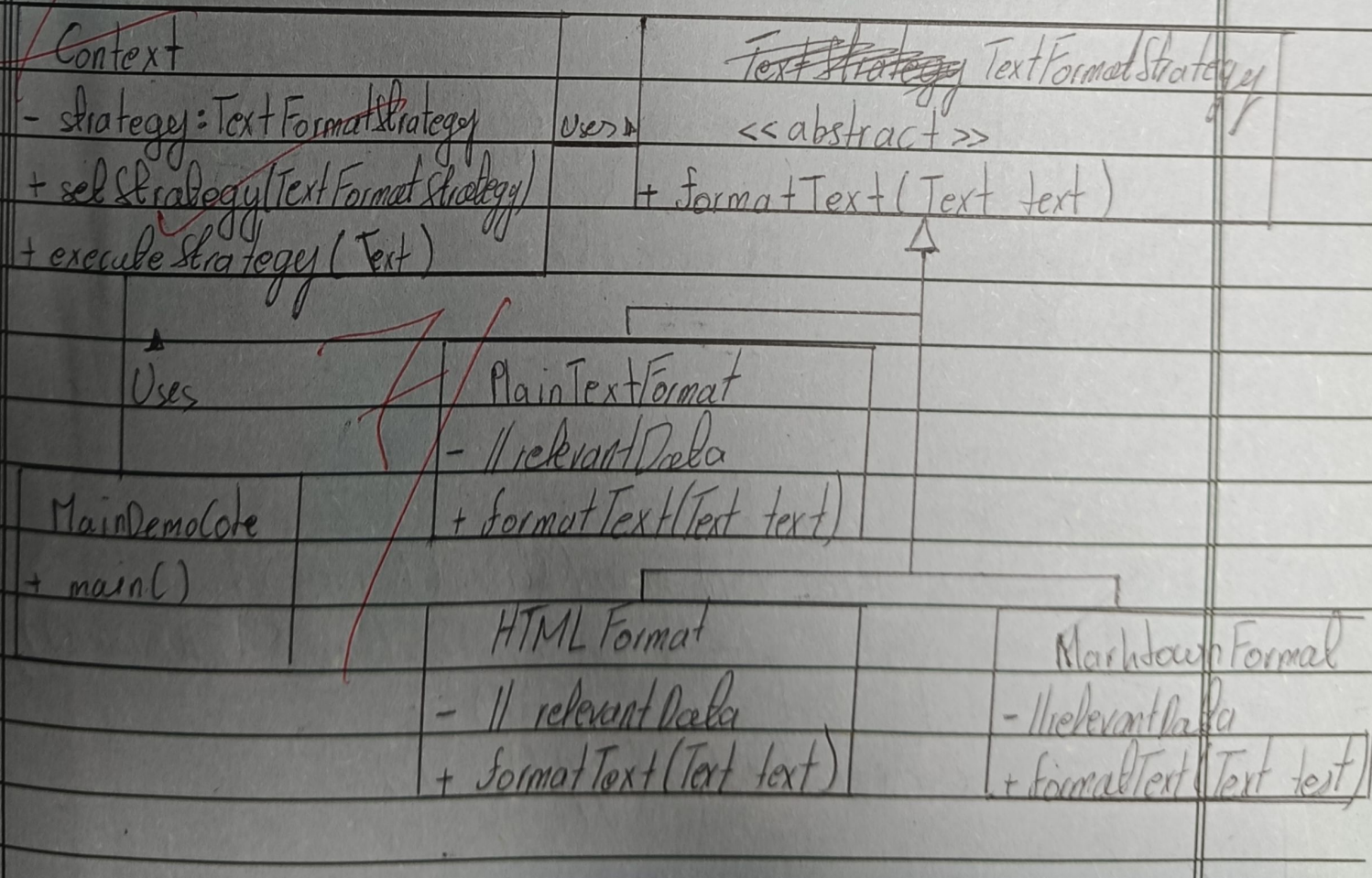


The stock will maintain a list traders. Traders can be added as an observer of a stock using the addTrader() method. When the state of the stock updates using the updateStock() method all the traders added will get notified using the update() method.

3/

(b)

=> Strategy Pattern is used when ~~mult~~ there are multiple algorithms for a specific problem. It enables to encapsulate each algorithm in a separate class & a common interface to access them. This makes each algorithm independent of the other, hence, making it easier to modify it without affecting the other algorithms.



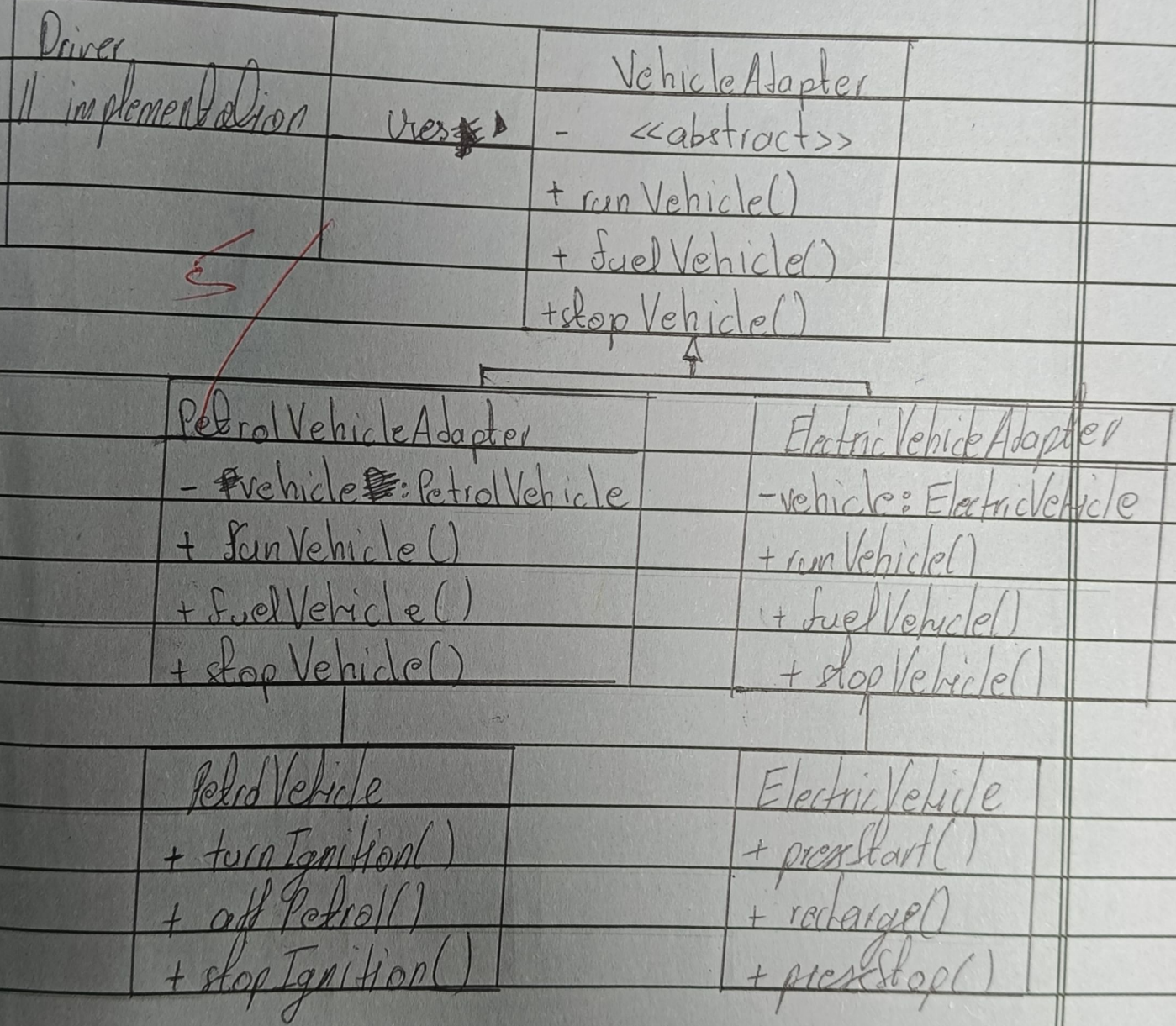
• Java Code for Context class

```
Class Context {
    private TextFormatStrategy strategy;

    public Context(TextFormatStrategy str){
        strategy = str;
    }

    public void setStrategy(TextFormatStrategy str){
        strategy = str;
    }

    public void executeStrategy(Text text){
        if(str != null)
            str.execute;
    }
}

~~public~~ Class Demo{
    public static void main(String[] args){
        Context c = new Context(new HTMLFormat());
        Text text = new Text();
        c.executeStrategy(text);
    }
}
```

(c)

=> Pattern used to bridge the gap between incompatible interfaces is the adapter pattern.

=> UML Class Diagram is as follows.

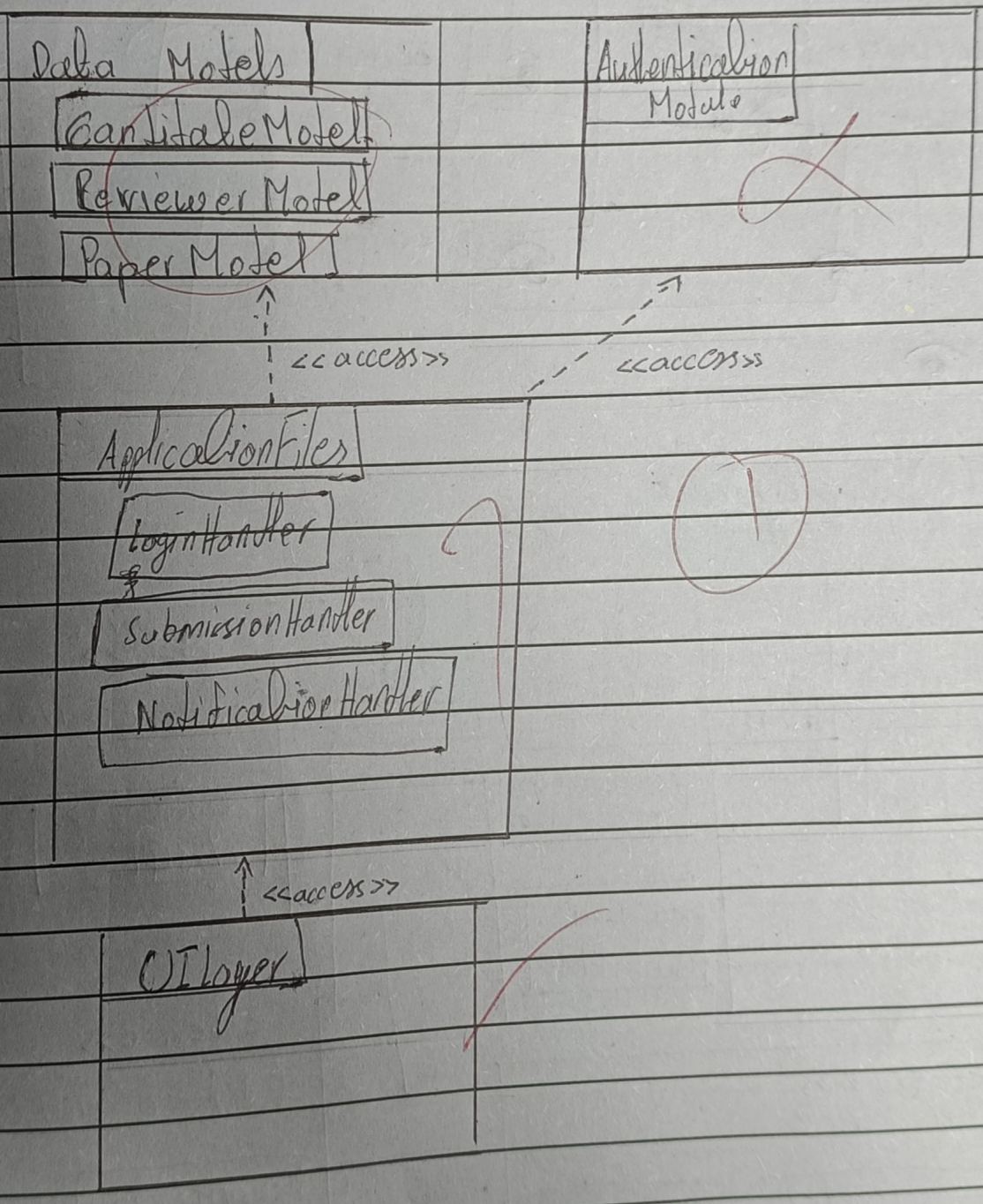


=> If I am adding new code to old code such that they are incompatible, then adapter pattern will be used. Facade pattern will be used to simplify the public interfaces available ensuring encapsulation & abstraction.

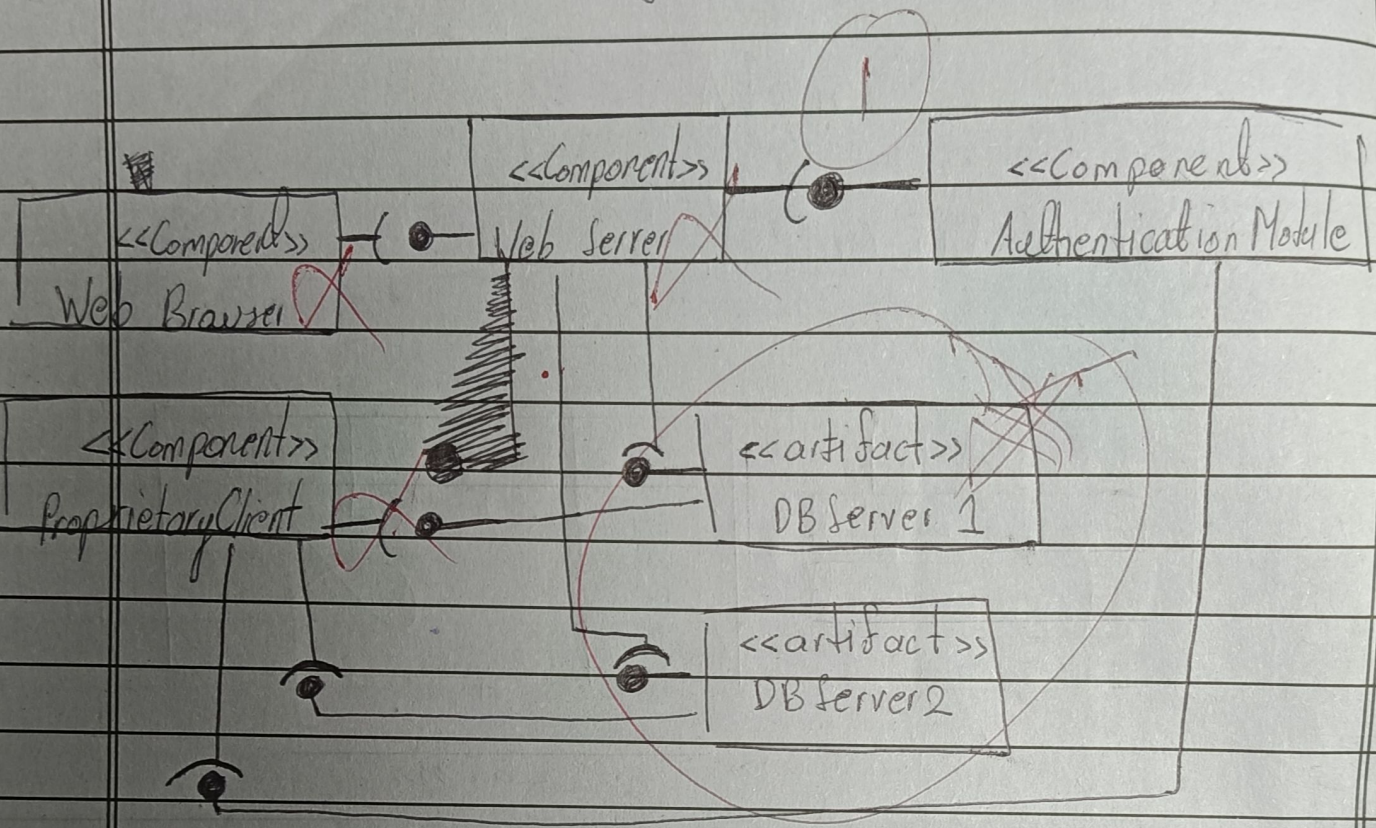
Q.2

Part : A

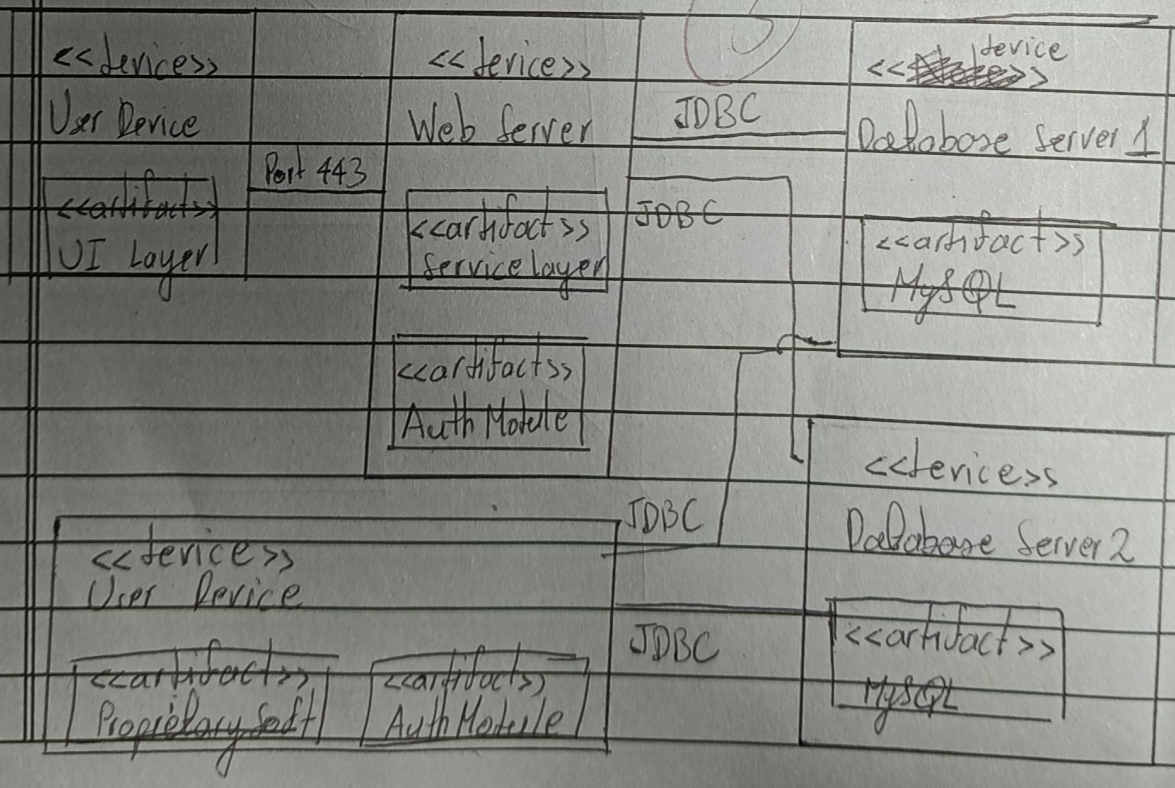
(a) Package Diagram



(b) Component Diagram



(c) Deployment Diagram



Rough

Part B

1. private TextField nameTxtFld = new TextField();
2. private ComboBox dptComboBox = new ComboBox();
3. private TextField salTxtFld = new TextField();

4. String query = "Select * From Employee";
5. Statement stmt = conn.prepareStatement(query);
6. ResultSet rs = stmt.executeQuery();

5

Part C

Activity	Option
i	G
ii	B
iii	C
iv	A
v	F
vi	D
vii	E

Q.3

Part A

(a)

Captain will play in 16 games (2)

(b)

A coach can lead several teams
A league can have 6 teams max
A team can have 16 games max
Assuming ~~a~~ a person can be a coach in only league then

$$\underbrace{1}_{\text{No. of leagues}} \times \underbrace{6}_{\text{Max teams}} \times \underbrace{16}_{\text{Max Matches}} = 96$$

16
6
36
6x
96

Ans. 48, as two teams play in a single match.

(c) Yes. As ~~composition is a much stronger relationship~~ and a ~~most~~ team must have a captain, hence, this is appropriate. If a

No, both should be modelled using composition. (2)

Reason:

A team requires players to exist. If players are removed so does the team. Thus it needs to be modelled using composition as it is

a strong relationship. It ensures the existence of players for the team to exist.

In case of captain, ~~it should be modelled using aggregation as a team can still exist if there~~ it should also be modelled using composition, as a team requires one captain.

Domain Model also shows a team must consist of 16 players and similarly a team must ~~at~~ ~~to~~ have a single captain.

(d) Protected Variations

Make game abstract
And extend it using RegularGames & PlayoffGame classes.

01

Part B

- Java Code:

```
// import relevant packages

class Session {
private ATM atm;

public Session(ATM a){
    atm = a
}

public void performSession(){
Card card = atm.getCardReader().readCard()
    CustomerConsole cC = new CustomerConsole();
    int pin = cC.readPin();

while(1){
    syso("Do you want to perform transaction: ?");
    syso("1 for Yes. 0 for No");
    int choice = sc.getInt();
    if( choice == 1){
        Transaction t = new Transaction(atm, this, card, pin);
        t.performTransaction();
    }
else
    break;
}
```

```
        atm.getCardReader().ejectCard();
    }
}
```

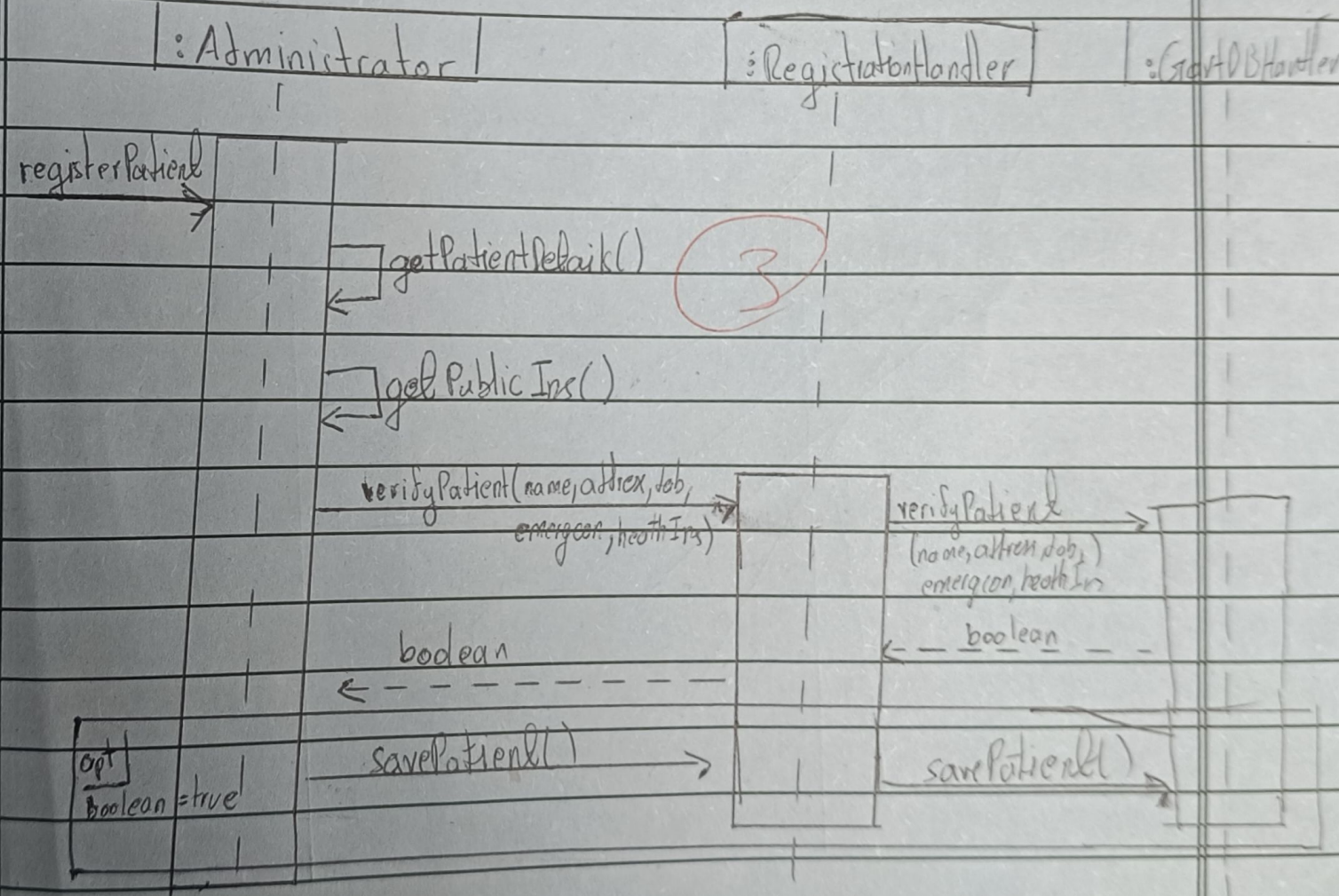
```
class Transaction {
    private ATM atm;
    private Session session;
    private Card card;
    private int pin;

    public Transaction(ATM a, Session s, Card c,
                       int p) {
        atm = a
        session = s;
        card = c;
        pin = p;
    }

    public void performTransaction() {
        // implementation
    }
}
```

Part C

(a) Actors : Patient, Administrator

Objects : Patient, Administrator, GovtDBHandler, RegistrationHandler

(b) UML Sequence Diagram

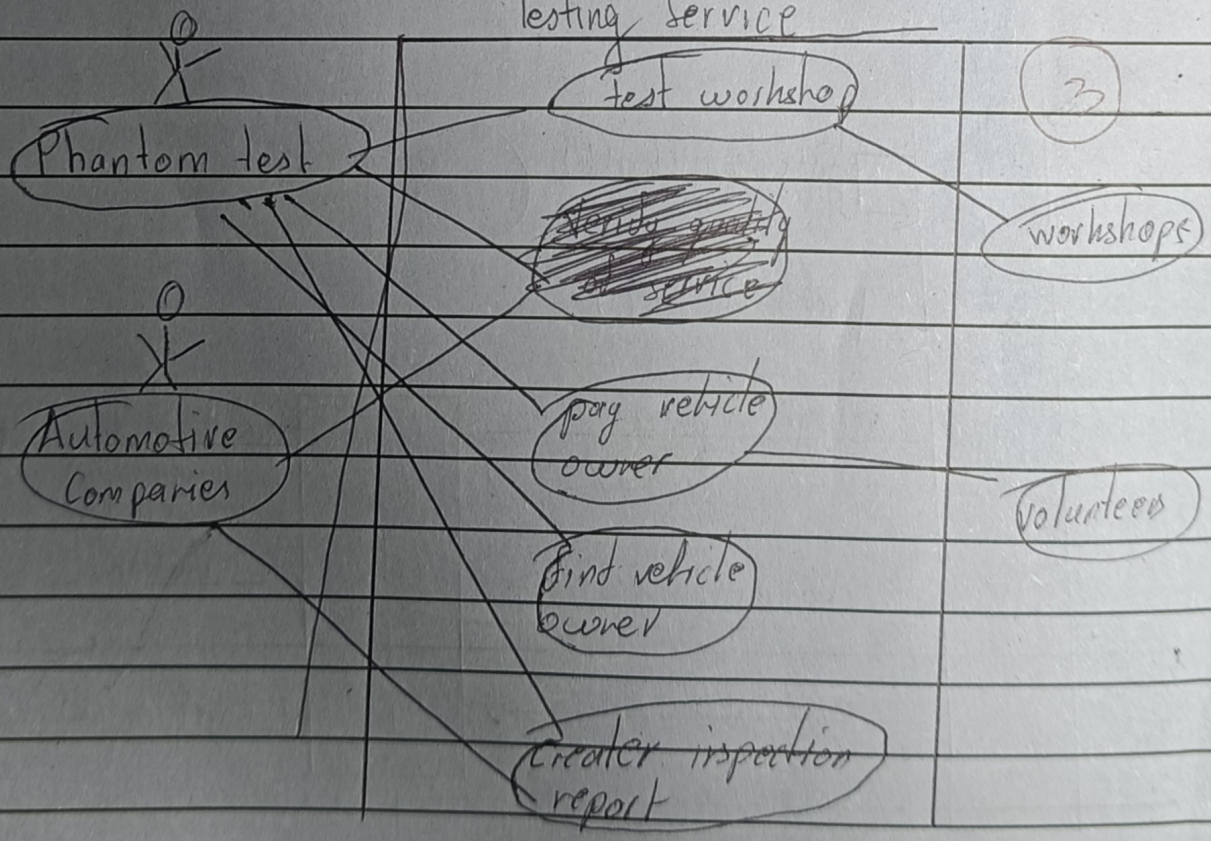


Q.4

(a) Actors : Automotive Companies, workshops, Phantom test, volunteers, (3)

(b) Use Cases : ~~verify quality of service,~~ ~~create a defect~~, test workshop, find vehicle owner, ~~inspect car repair~~, create report, pay vehicle owner (2)

(c)



Back

1	A	B	C	**D**	E
2	A	B	C	**D**	E
3	**A**	B	C	D	E
4	A	B	**C**	D	E
5	A	**B**	C	D	E
6	A	B	C	**D**	E
7	A	B	**C**	D	E
8	A	B	C	**D**	E
9	A	B	**C**	D	E
10	A	**B**	C	D	E
11	A	B	C	**D**	E
12	A	**B**	C	D	E
13	A	**B**	C	D	E
14	A	B	**C**	D	E
15	A	B	C	**D**	E
16	A	**B**	C	D	E

Clear Scan Key Save

17	**A**	B	C	D	E
18	**A**	B	C	D	E
19	**A**	B	C	D	E
20	**A**	B	C	D	E
21	A	**B**	C	D	E
22	A	B	**C**	D	E
23	A	**B**	C	D	E
24	A	B	**C**	D	E
25	A	B	**C**	D	E
26	A	**B**	C	D	E
27	A	B	C	**D**	E
28	A	B	**C**	D	E
29	**A**	B	C	D	E
30	**A**	B	C	D	E

Clear Scan Key Save