

[CLO 1: To understand requirement engineering with analysis]

PART B

Case Study:

Question # 02

[20 marks]

The system to be developed is intended to support the day-to-day operations of a restaurant by improving the processes of making reservations and allocating tables to customers. The restaurant currently operates a manual booking system using handwritten forms stored in a large folder. An example of the current manual booking form is given in Figure 1. Each row on this form corresponds to a particular table in the restaurant. Bookings are entered for a particular table, and the number of 'covers', or diners expected, is recorded for each booking, so that a suitably sized table can be allocated. The restaurant runs three sittings in an evening, known as the "pre-theatre", the "dinner" and the "support" slots, but as the form illustrates, these slots are not strictly adhered to and bookings can be made for time periods that span more than one slot. Finally, a contact name and phone number are recorded for each booking.

DINNER BOOKINGS - DATE / /

Table No.	5.30 p.m. – 7.30 p.m.		7.30 p.m. – 10.00 p.m.		10.00 p.m. – 12.00 Mid Night	
	Covers	Name, Phone No.	Covers	Name, Phone No.	Covers	Name, Phone No.
1						
2						
3						

Figure 1: A manual booking sheet

Currently there are tables for 2 covers, 4 covers and 6 covers. Annotations are made to the booking sheet to record various events. When a party arrives and is seated, the corresponding booking in the booking sheet is crossed out. The party can be seated at a table other than the booked if it is vacant. This is shown by drawing an arrow from the original booking to the new table in the booking sheet. If a customer phones to cancel a booking, a note is made in the booking sheet to indicate that the booking has been cancelled. Other pieces of information, such as the time by which a table must be vacated, can also be written on the sheet. Dinner can also be taken at the restaurant without making an advance booking, if a free table is available. This is known as a "walk-in" and is shown on the sheet as a booking to record table occupancy, but no record of the customer's name or telephone number is made.

Once the customers arrive, they are ushered into their respective table/tables, and the dining menu containing menu items (food, beverages and deserts etc.) is presented by the waiter and menu items chosen are noted down. Details of these menu items and their respective quantities are indicated in the invoice items. Customer is presented with the invoice and the relevant payment is made either by cash or using a credit card. Since a customer can have several bookings or reservations (if accommodation cannot be provided at a single table, multiple tables have to be allocated) at a particular time, invoice prepared may cover all such bookings.

The need for a computerized system

The restaurant management has identified a number of problems with the manual system. The system is slow, and the booking forms quickly become difficult to read. This can lead to operational problems, such as customers being prevented from making a booking because it is not obvious from the booking sheet that there is in fact a table free. There is no backup system: if a sheet gets destroyed, the restaurant has no record of what bookings have been made for that evening. Finally, it is very time-consuming to get even simple management data, such as the rate of table occupancy, from the existing booking sheets.

EJAZ + MAJID

The system must be able to record significant events that take place when the restaurant is open, such as the arrival of the customer. Operation of the system will be as far as possible by direct manipulation of the data presented on the screen. For example, it should be possible to change the time of a booking, or the table it is allocated to.

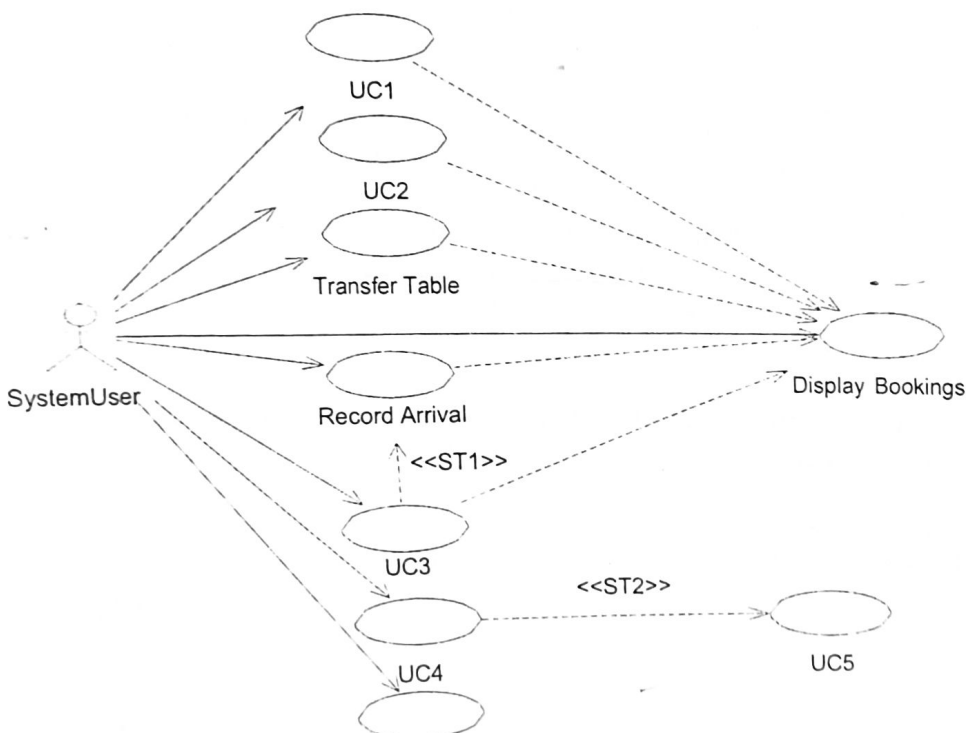
System should also support the following:

2. Allow cash or credit card payments. ✓

If a customer uses two or more tables with more than 8 covers, 7% discount is given. This will be applied for any kind of payment. Further 3% discount is given for credit card payments.

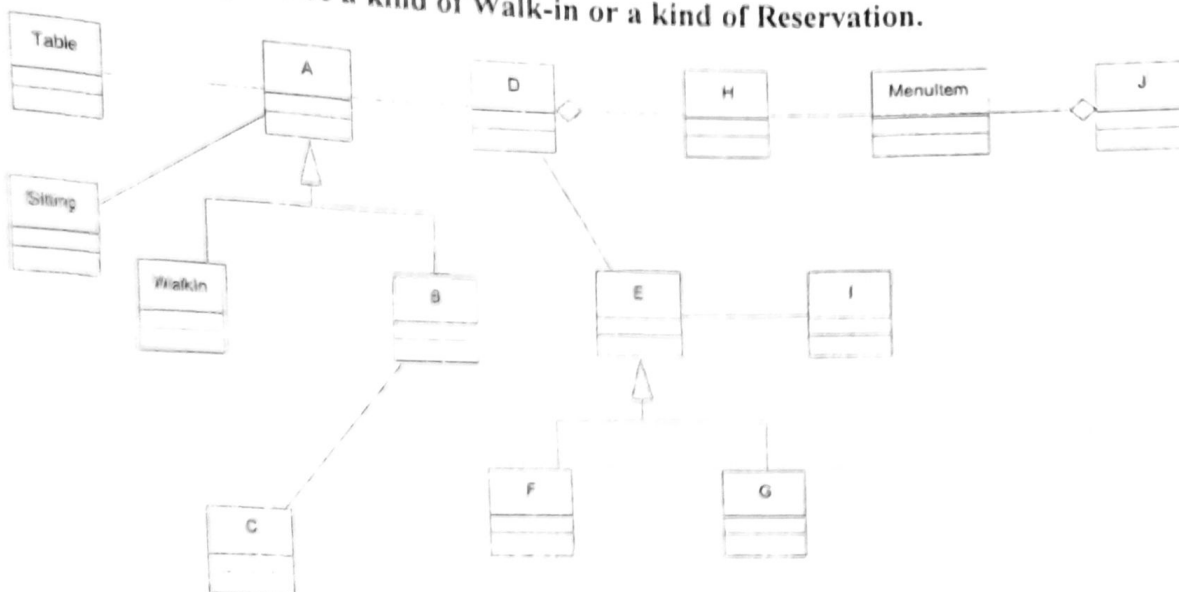
3. Generate reports to support the management such as,
- Details of Regular customers,
 - Customers who pay by credit cards (sorted according to the type of credit card),
 - Customers who do online reservations,
 - Table-wise demand for reservations etc.,
 - Total discounts given for credit card customers monthly (sorted according to the type of credit card),
 - Monthly Income for the restaurant,
 - Most popular menu items for a given month, year etc.

(a) Identify correct Use Cases from the above scenario to represent Use Cases designated from UC1 to UC5 in the Use Case diagram shown below. Where <<ST1>> AND <<ST2>> represents include relationship in following digarm. (10 Marks)



Given use cases	Correctly identified use cases
UC1	CreateBooking ✓
UC2	CancelBooking ✓
UC3	Assign Table ✗
UC4	Make Payment ✓
UC5	Generate Invoice ✗

(b) From above scenario, identify classes, which are best suited for classes designated from A to J in the class diagram given below.
Hint: A Booking can be a kind of Walk-in or a kind of Reservation.
(10 Marks)



6

Given classes	Correctly identified classes
A	Booking ✓
B	Reservation ✓
C	Customer ✓
D	Invoice ✓
E	Payment ✓
F	Cash Payment ✓
G	Card Payment ✓
H	Order ✗
I	Discount ✓
J	Dining Menu ✓

9

[CLO 2: To manage design artifacts and generate code]

Question 3. [20 marks]

- a) Draw a system sequence diagram of the following form with boundaries user, Form (UI) and Controller, navigations of form are explained in (b).
- b) Write down Java Code using Controller class for the following form, managing database connection. User enters employee number by pressing Find button; Update button modifies any information of an existing displayed record. Add button saves new record using this form. Message box displays all possible buttons' events.

id **Employee's Information**

? *Employee=

*Employee Name

Job Title

Manager ID

Hire Date

Salary

Commission

Department No

[Return Main Menu](#)

Message:

[CLO 3: To identify different design patterns in real world problems]

Question 4. [10 marks]

For the following scenarios you need to just identify, the best design pattern suitable for given set of pattern and provide the name for that pattern once identified correctly in given table.

1. You are developing an online shopping platform where users can browse products, add items to their cart, and proceed to checkout. Behind the scenes, the platform interacts with various subsystems such as product inventory, payment processing, and shipping logistics. As the complexity of these subsystems grows, it becomes cumbersome for client code (e.g., web controllers) to directly interact with each subsystem. What pattern would best support this? *Indirect*
2. Your application relies on a legacy database system with its own set of query methods. However, you need to switch to a modern database system with a different query interface. Describe how you would find solution to maintain compatibility with existing code while transitioning to the new database system. What pattern would best support this?
3. You have created a subsystem with 25 classes. You know that most other subsystems will only access about five methods in this subsystem; how can you simplify the view that the other subsystems have of your subsystem?

4. Your company sells computer parts. You are building an inheritance hierarchy of different kinds of parts. The hierarchy should contain the superclass Part and the three subclasses CPU, HardDisk, and CDROM. However, your supplier Franz already has classes for two of these parts: namely, FranzCPU and FranzHardDisk. How can you build a class hierarchy for your system and reuse Franz' classes?
5. You are building an e-commerce platform that supports multiple payment methods: CreditCard, PayPal, and Bitcoin. Each payment type requires creating a different payment processor object. You want the system to decide at runtime which payment processor to create based on the user's selection, without changing the client code.

Given Options	Design Pattern
1	Indirection
2	Adapter
3	Polymorphism
4	Polymorphism
5	Factory

3

[CLO 3: Apply programmatic approach to implement different patterns in code]

QUESTION 5 [10 marks]

Scenario: You are working on a multi-threaded application and need to implement a logging mechanism. You want to ensure that there is only one instance of the logger to avoid issues with concurrent access and resource contention. For this problem identify a suitable design pattern and provide a complete java code with main function.