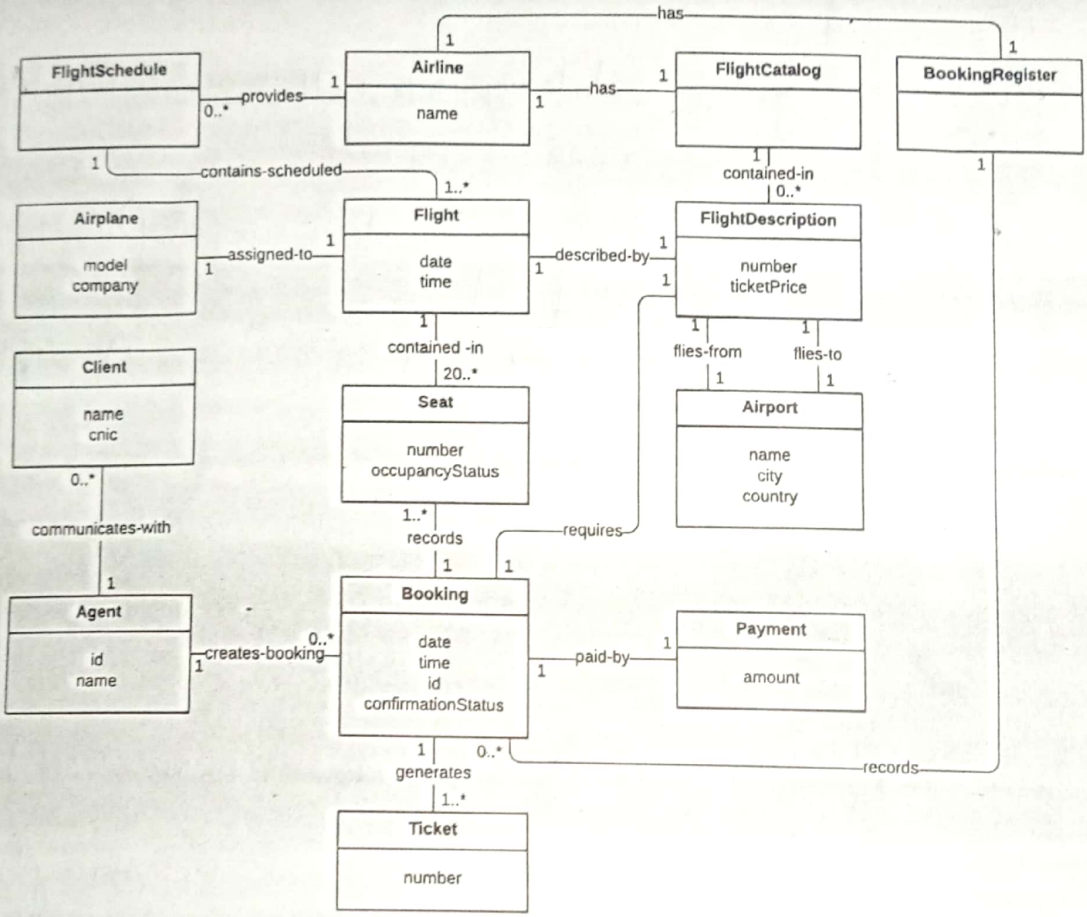
Question 01: Provide the pre-condition, post-condition and alternate scenarios for the given "Book Flight" Use case. [2+2+6]

Use case Name	Book Flight
Actor Actions	**System Response**
Pre-condition: System Availiable, flights Availiable according to client requirements, seats availiable	
1. The use case begins when the agent specifies a travel itinerary (schedule) for a client.	
	2. The system searches a set of appropriate flights and present them to the agent.
3. The agent selects the flight.	
	4. The system verifies the space is available on the flight and reserves a seat on the flight.
5. The agent finalizes the booking by supplying payment information.	
	6. The system books the seat and issues the ticket.
Post-condition: Seat booked successfully, information saved correctly.	

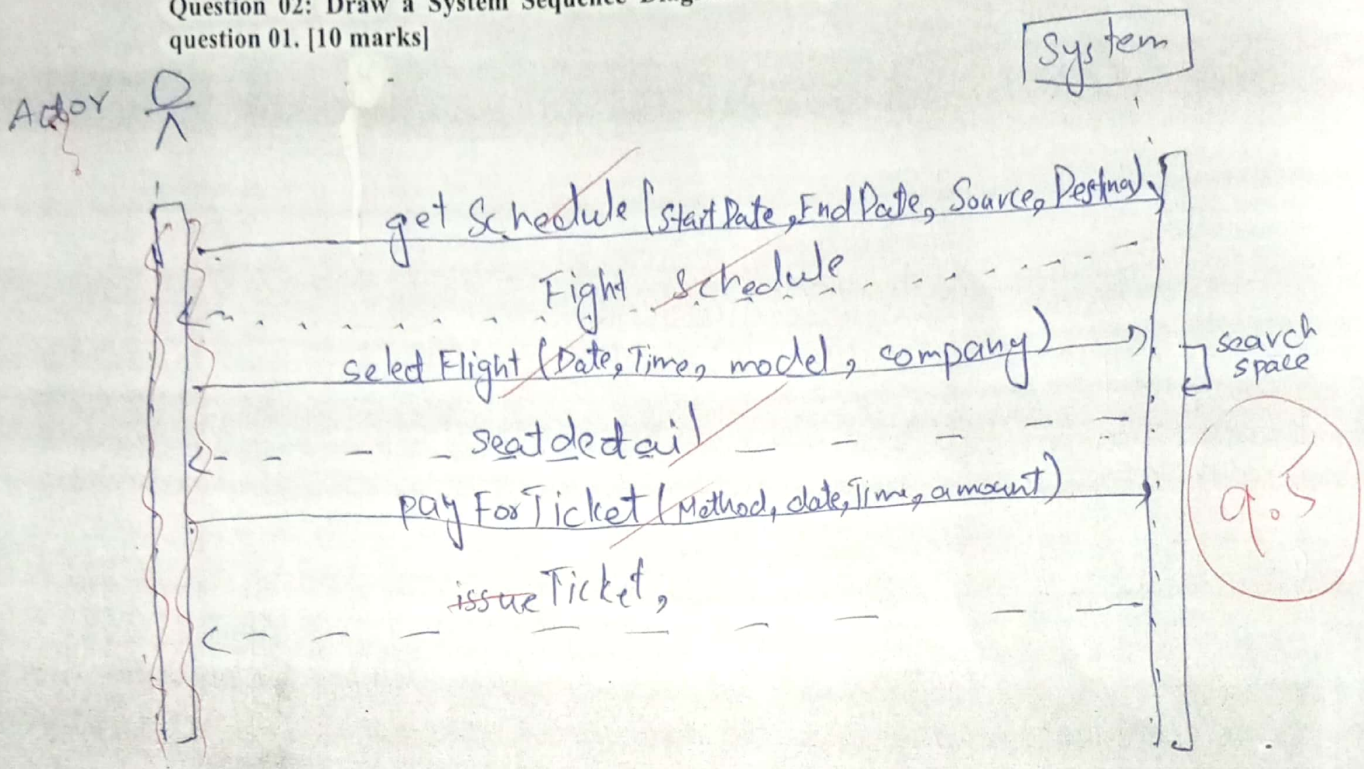
Alternative Scenarios:

1. In step 4: Seat not availiable on flight
   Repeat step 3 and 4 for alternative flight
2. In step 1: flights not availiable
   Change schedule requirements
3. In step 6: Information not ~~store~~ correct
   ~~Ticket~~ Cancel Ticket and Repeat all steps again with correct Information

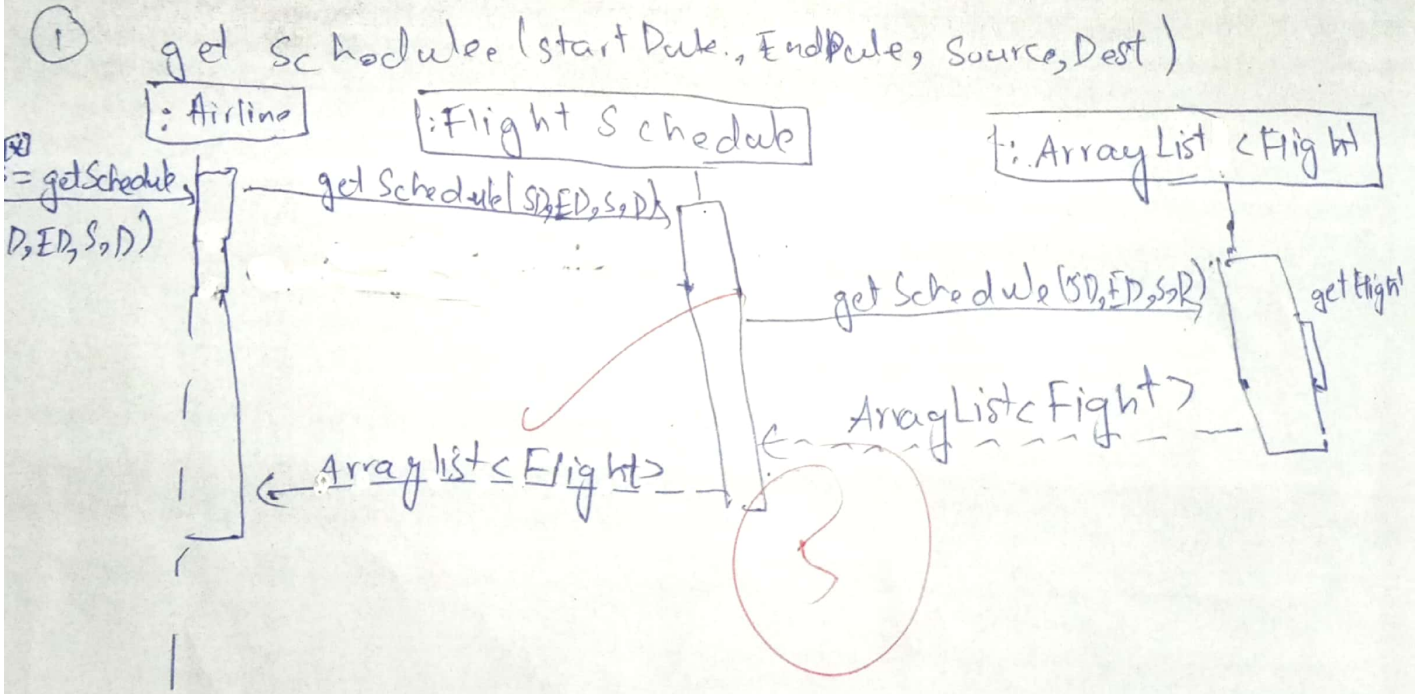
Domain Model



Question 02: Draw a System Sequence Diagram corresponding to the scenario given in the question 01. [10 marks]



Question 03: Produce design sequence diagrams for "Book Flight" use case by apply GRASP patterns. [15 marks]

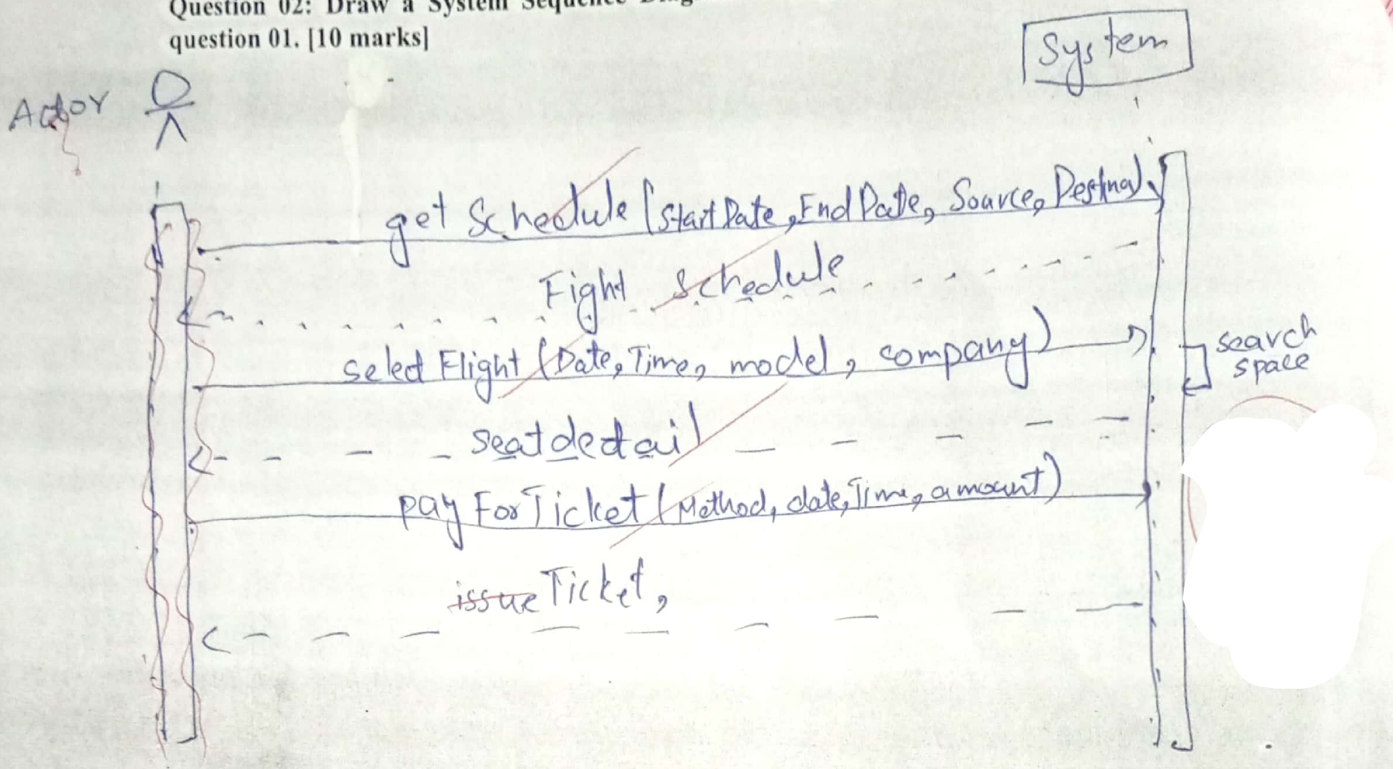


SD = StartDate
ED = EndDate
S = Source
D = Destination

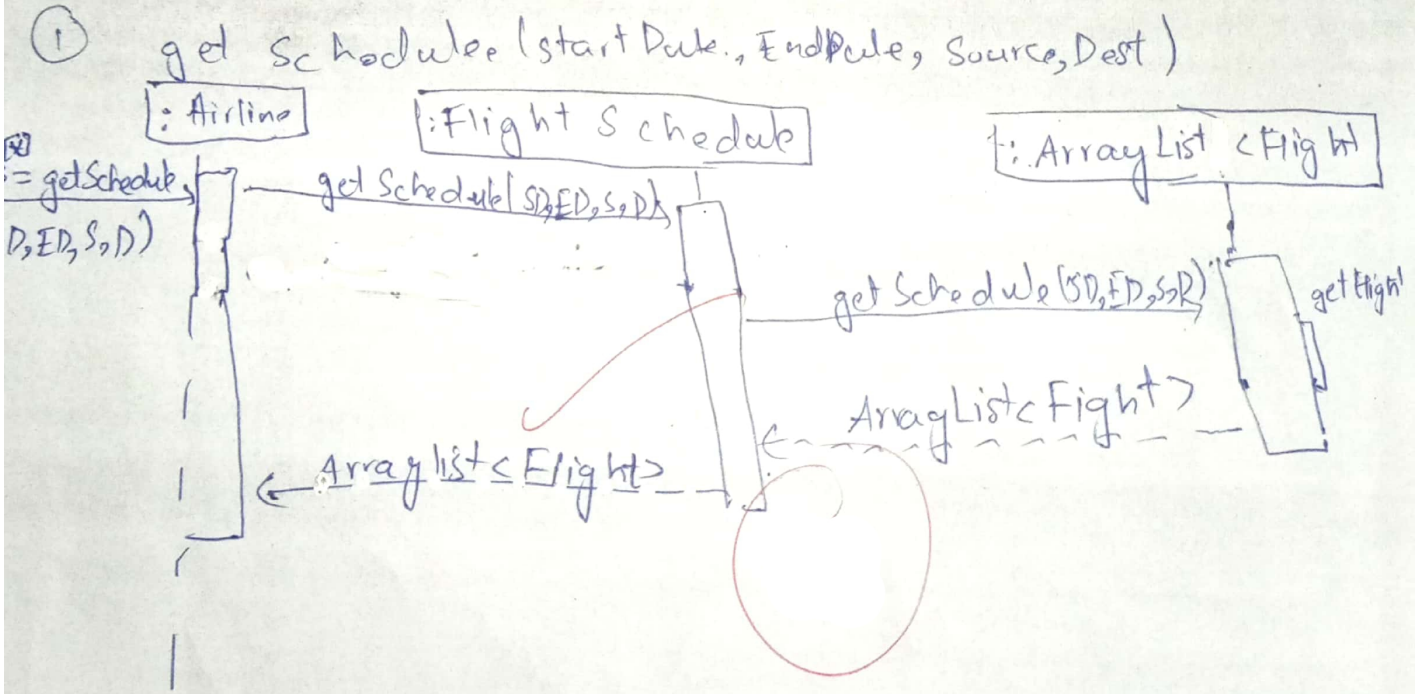
F = Flight
C = Client
M = Method of Payment

A = Amount
T = Ticket

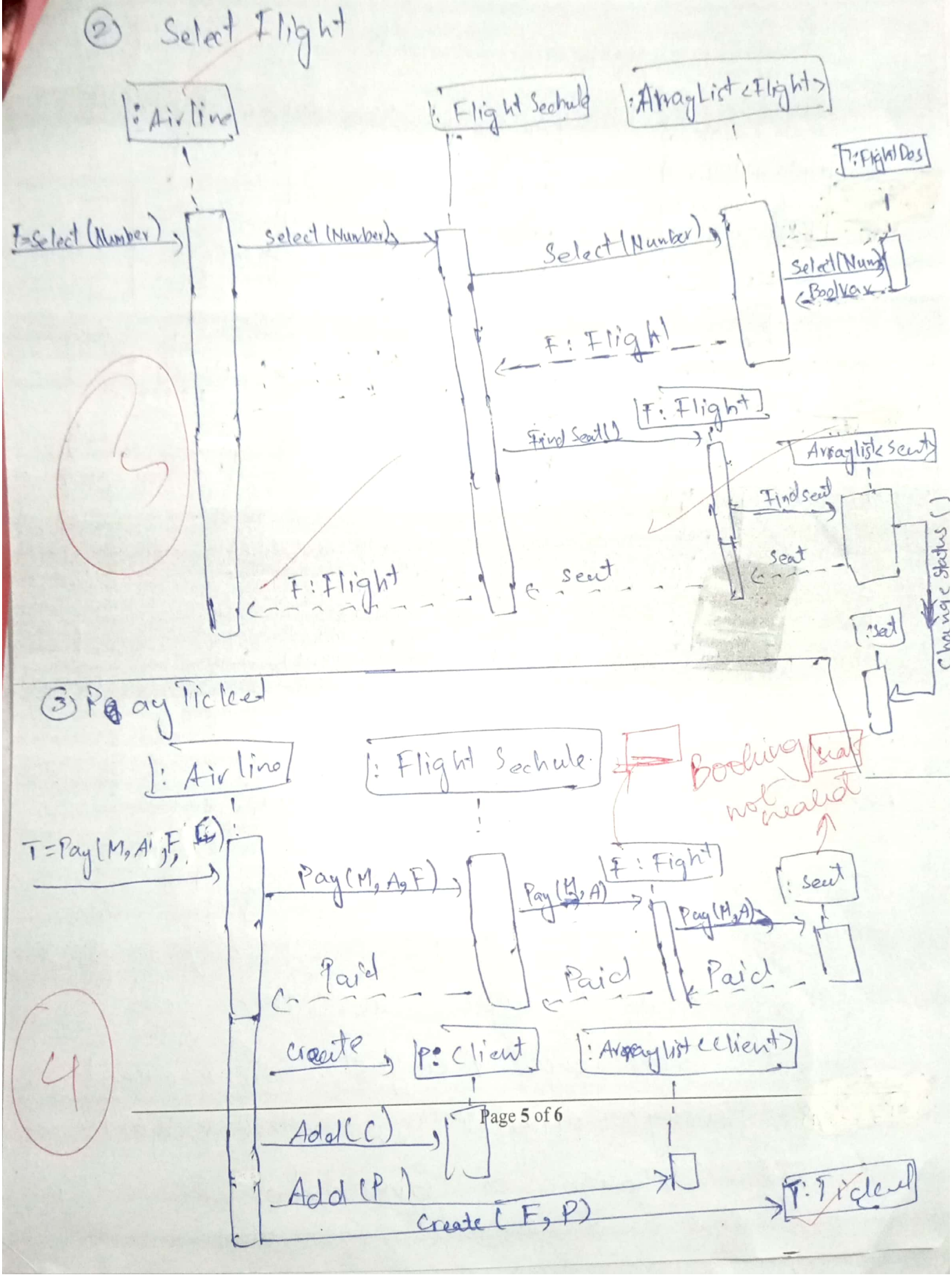
Question 02: Draw a System Sequence Diagram corresponding to the scenario given in the question 01. [10 marks]



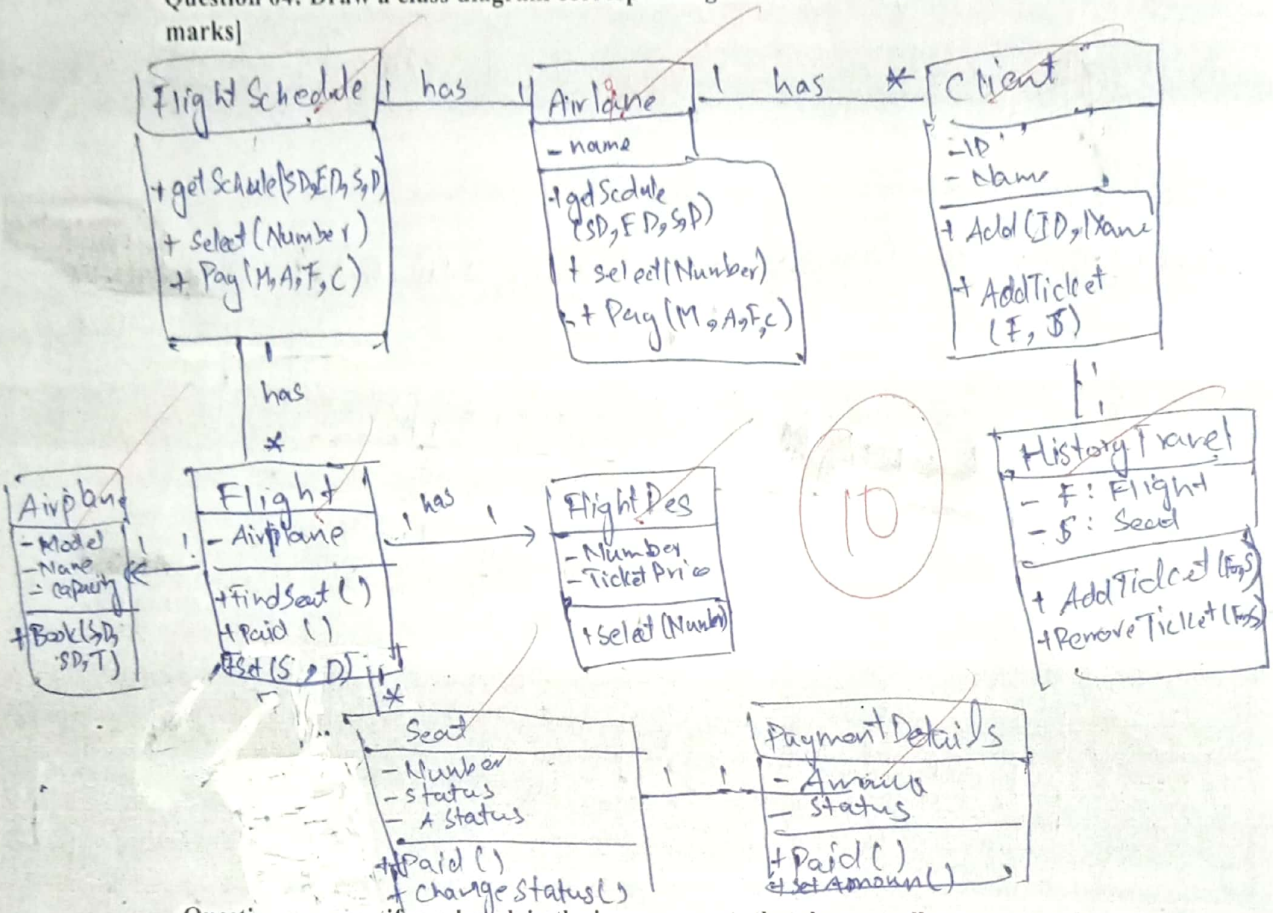
Question 03: Produce design sequence diagrams for "Book Flight" use case by apply GRASP patterns. [15 marks]



SD = StartDate
ED = EndDate
S = Source
D = Destination
F = Flight
C = Client
M = Method of Payment
A = Amount
T = Ticket



Question 04: Draw a class diagram corresponding to the scenario given in the question 01. [10 marks]



10

Question 05: Identify and explain the improvements that the controller, expert and creator patterns introduce to the behavioral design for the "Book Flight" use case. [5 marks]

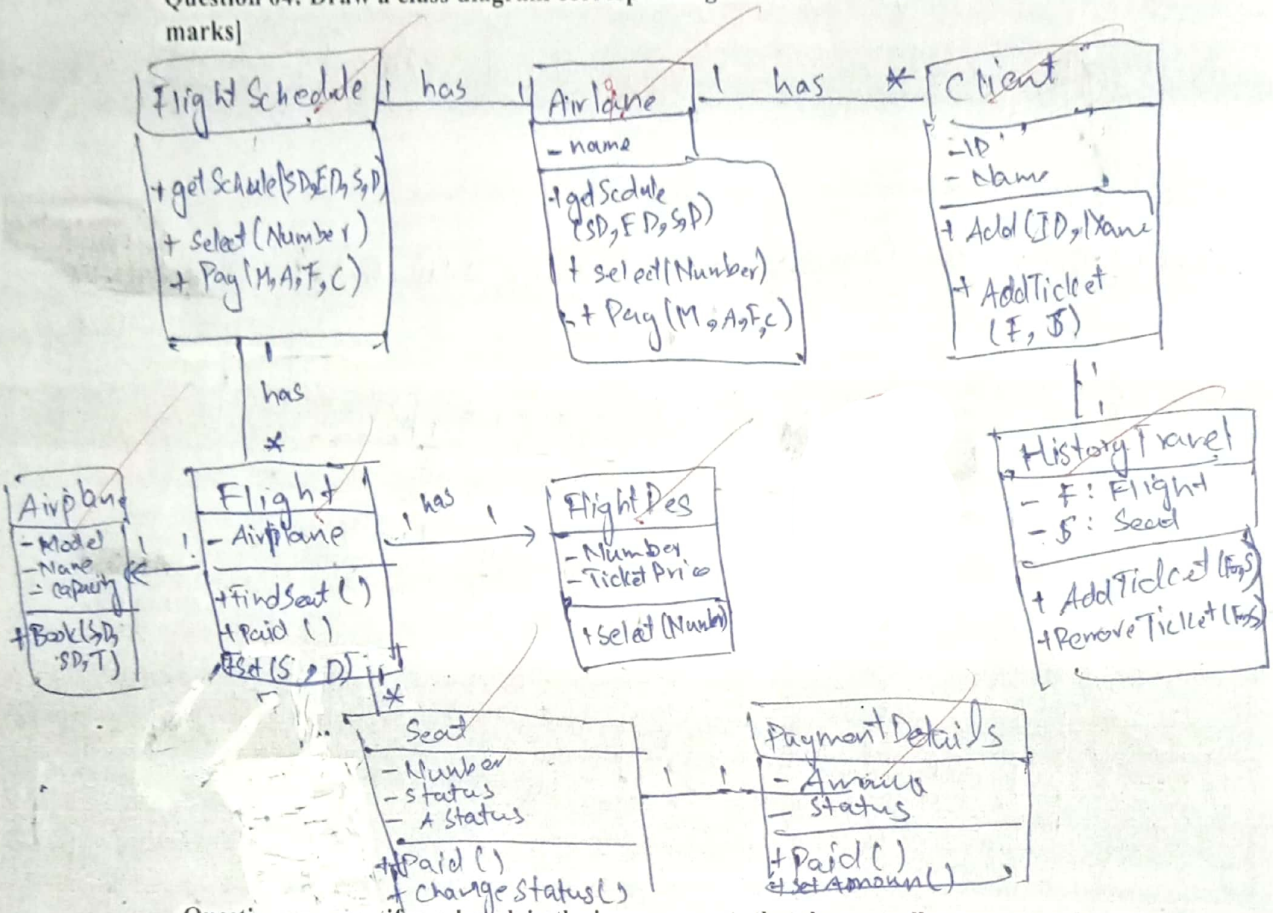
(1) It helps to reduce the complexity by adding expert for Flight Schedule and Clients.

(2) Controller class is helps to easily move to any direction from one place.

(3) Crator create the object and then add to respect class and Arraylist. So no need to go through arraylist or other objects to make change or save date.

2.5

Question 04: Draw a class diagram corresponding to the scenario given in the question 01. [10 marks]



Question 05: Identify and explain the improvements that the controller, expert and creator patterns introduce to the behavioral design for the "Book Flight" use case. [5 marks]

(1) It helps to reduce the complexity by adding expert for Flight Schedule and Clients.

(2) Controller class is helps to easily move to any direction from one place.

(3) Crator create the object and then add to respect class and Arraylist. So no need to go through arraylist or other objects to make change or save date.

2.5