Mangal
Date: / /201

1. Add 2 no.s (integers) by initializing the values.

```
class Add {
  public static void main (String args[])
  {
    int a = 5;
    int b = 10;
    System.out.println ("Addition of two
        numbers : "+(a+b));
  }
}
```

2. Add 2 integers by taking user i/p

```
import java.util.*;
class Add {
  public static void main (String args[])
  {
    Scanner sc = new Scanner (System.in);
    System.out.println ("Enter values for a and
        b");
    int a = sc.nextInt();
    int b = sc.nextInt();
    System.out.println ("Addition of "+a+ "and "
        +b+ "is "+(a+b));
  }
}
```

Scanned with CamScanner

Date: / /201

3. Add 2 no.s by using method / function

```
import java.util.*
class Add {
    void add (int a, int b)
    {
        int c = a+b;
        Sopln("Addition of 2 no.s " +c);
    }
    public static void main (String args[])
    {
        Scanner sc = new Scanner(System.in);
        Sopln("Enter a and b");
        int a = sc.nextInt();
        int b = sc.nextInt();
        Add ob = new Add();
        ob.add(a, b);
    }
}
```

4. Perform the addition without using + operator

```
class Add {
    psvm (String args[])
    {
        int a, b, sum;
        a = 5;
        b = 10;
        sum = a-(-1)*b;
        Sopln(sum);
    }
}
```

Scanned with CamScanner

Mangal
Date: / /201

```
import java.util.*;
5. class Student {
    ~~Student~~
    String name, usn;
    int age;
    int a[];
    Student (String name, String usn, int age, int a[])
    {
        this.name = name;
        this.usn = usn;
        this.age = age;
        this.a = a;
    }
    void display()
    { int sum = 0;
      Sopln ("Name : " + name + "\nUSN : " + usn +
      "\nAge : " + age);
      for (int i = 0; i < a.length(); i++)
      {
          sum += a[i];
      }
      float avg = sum / 6;
      float percent = sum / 600 * (sum / 600) * 100;

      Sopln ("Avg : " + avg + "\n Percentage : "
                + percent);
    }
    psvm (String args[])
    {
    Scanner sc = new Scanner(System.in);
    Sopln ("enter name, usn, age, 6 sub marks");
    String name = sc.next();
    String usn = sc.next();
```

Scanned with CamScanner

Date: / / 201

```
int age = sc.nextInt();  int a[] = new int[6];
for (i = 0; i < 6; i++)
        a[i] = sc.nextInt();
Student obj = new Student(name, usn, age, a);
        obj.display();
  }
}
```

6.
```
class Calci {
    int a, b;
    Calci (int a, int b)
    {
        this.a = a;
        this.b = b;
    }
    void add ()
    {
        Sopln ((a+b));
    }
    void sub ()
    {
        Sopln ((a-b));
    }
    void mul ()
    {
        Sopln((a*b));
    }
}
```

Scanned with CamScanner

Mangal
Date: / /201

```
class Mainclass extends Calci {
    Mainclass (int a, int b)
    {
        super(a, b);
    }
    void div()
    {
        sopln((a/b));
    }
    psvm (String args[])
    {
        Mainclass obj = new MainClass(10, 20);
        obj.div();
        obj.add();
        obj.sub();
        obj.mull();
    }
}
```

7. Bank is a class that provides method to get rate of interest and rate of interest may diff. according to the banks. For eg. SBI, ICICI and axis banks provides 8.4%, 7.3%, and 9.7% respectively. Write a program to compute rate of interest for these 3 banks

```
class Bank {
    float getRateOfInterest()
    {
        return 0;
    }
```

Scanned with CamScanner

Mangal
Date: / /201

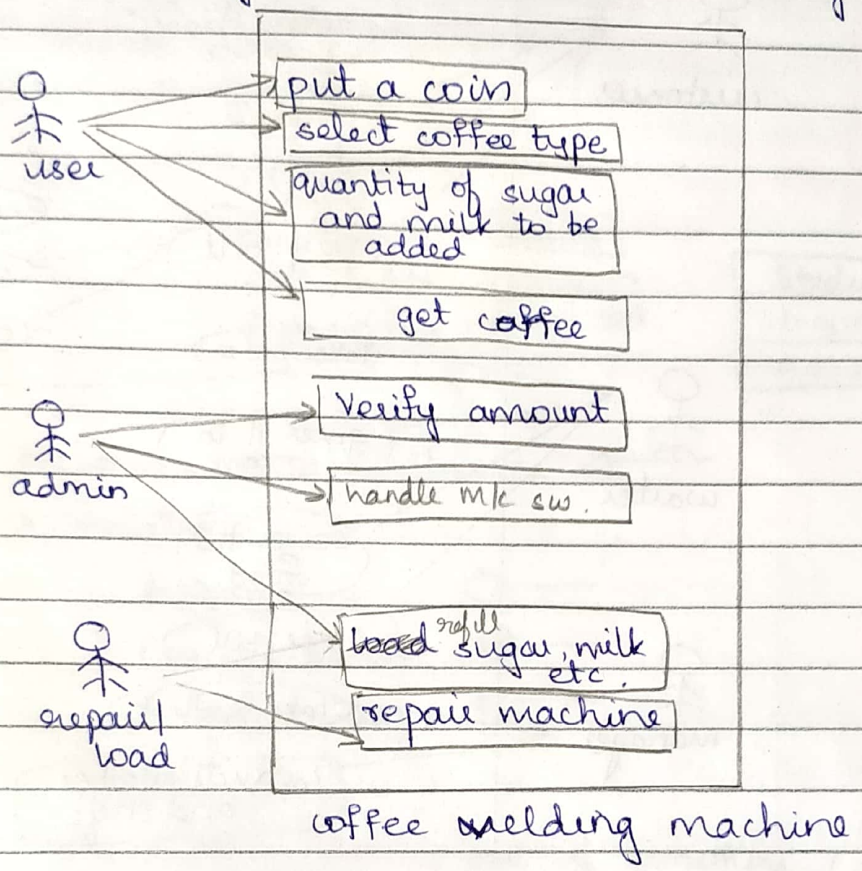
```
class SBI extends Bank
{
    float getRateOfInterest()
    {
        return 8.4f;
    }
}
class ICICI extends Bank
{
    float getRateOfInterest()
    {
        return 7.3f;
    }
}
class axis extends Bank
{
    float getRateOfInterest()
    {
        return 9.7f;
    }
}
class Polymorphism
{
    psvm(String args[])
    {
        Bank b = new SBI();
        Sopln("SBI rate ",+b.getRateOfInterest());
        b = new ICICI();
        Sopln("ICICI rate ",+ b.getRateOfInterest());
        b = new Axis();
        Sopln("Axis rate ",+b.getRateOfInterest());
    }
}
```

Scanned with CamScanner

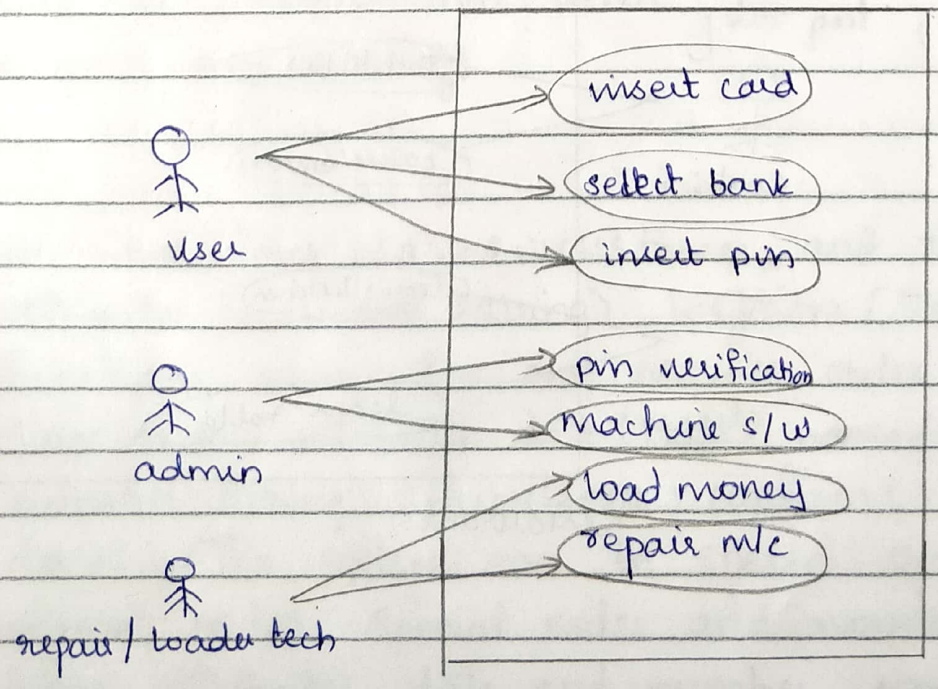
Mangal
Date: / /201

~~8. class Shape ~~

* coffee vending machine - use case diagram



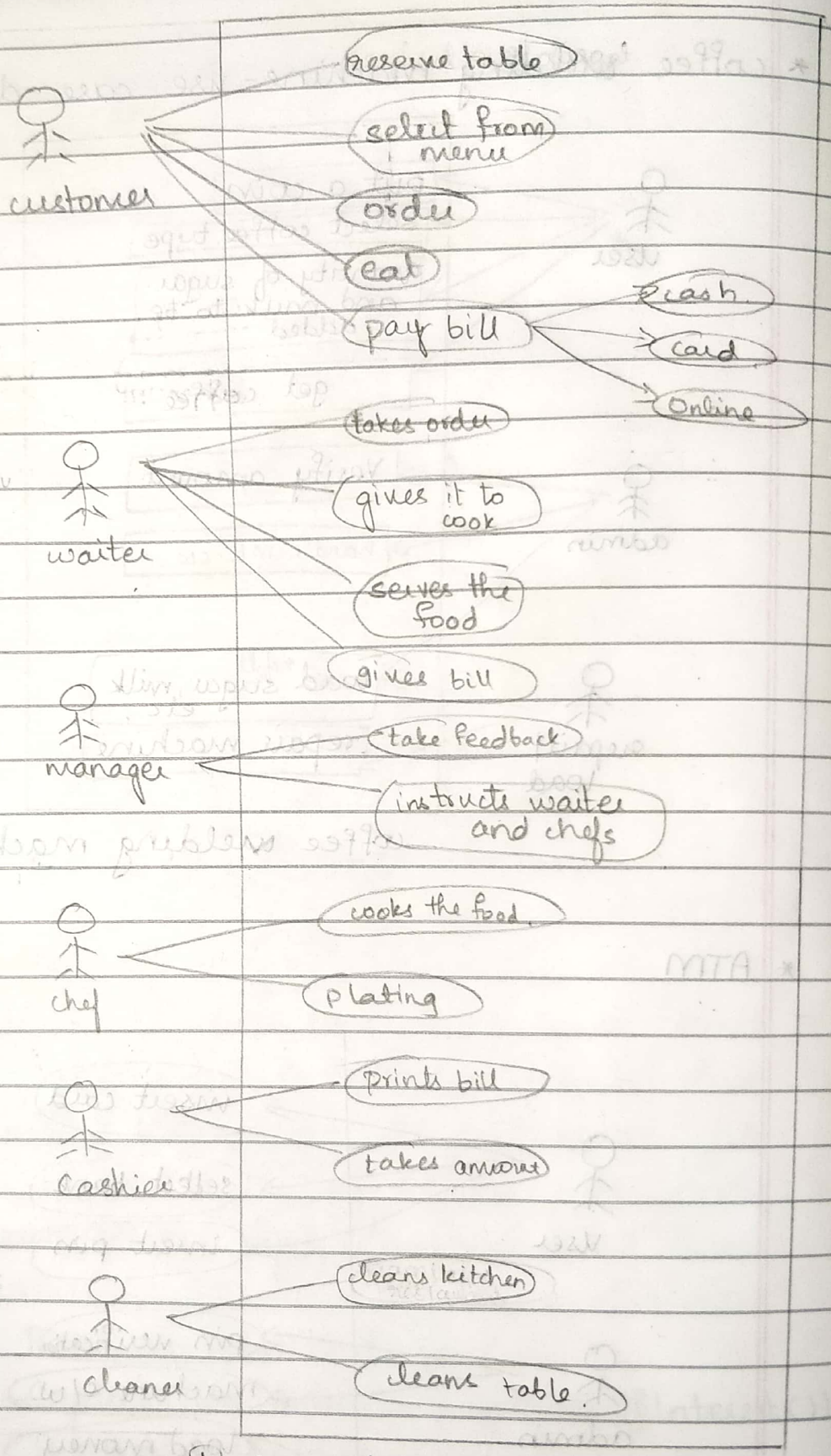
* ATM



Scanned with CamScanner

Mangal
Date: / /201

* Restaurant



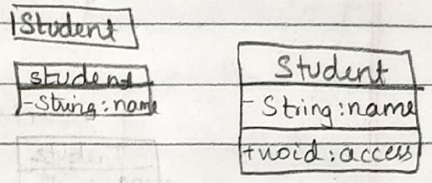
Scanned with CamScanner

Mangal
Date: / /201

* Online Shopping

Usecase Diagram → User → relationship → Use cases
Class diagram
Sequence diagram

classes
attributes
functions



Is a relation → Inheritance
has a relation
Aggregation
Composition
Dependancy

car has wheel
house has window } part of something (composition

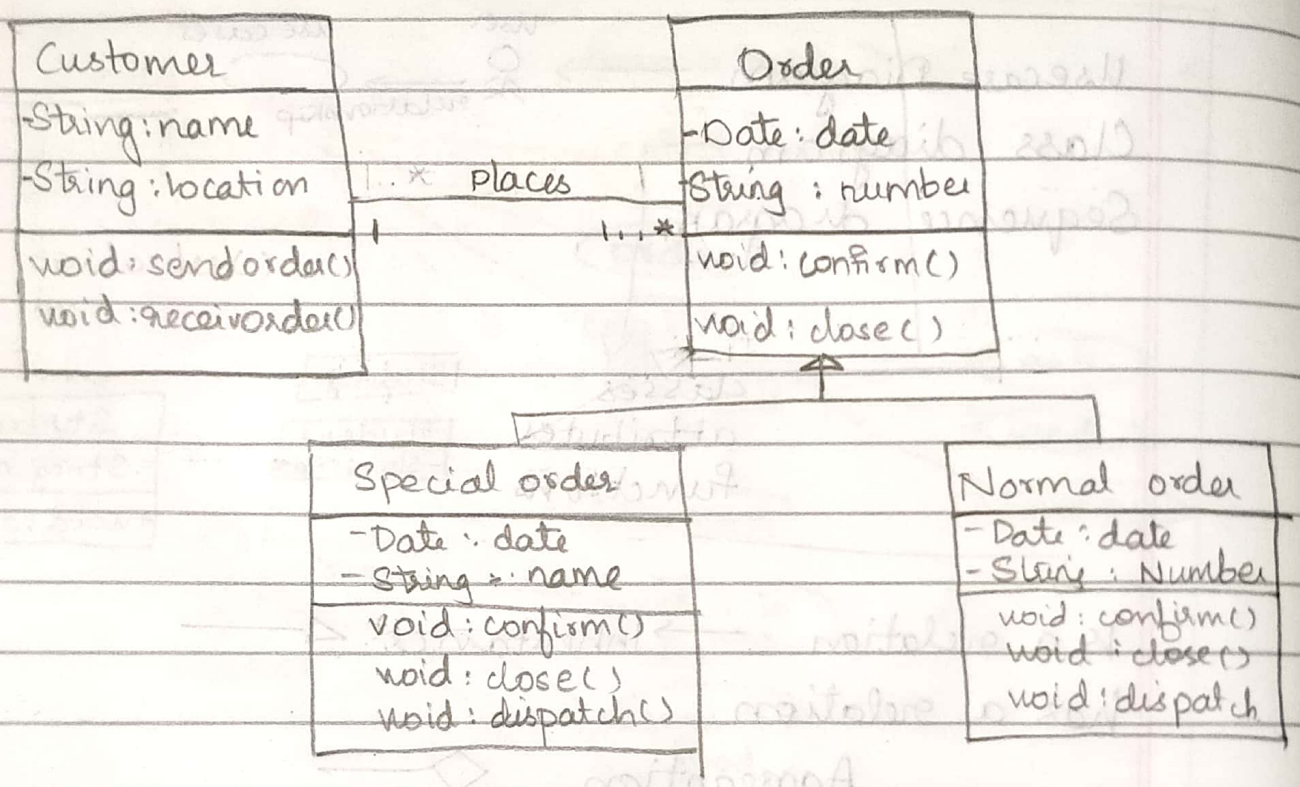
Railway station has train
Airport has aircraft } not part (Aggregation)

We have four classes (i) customer, and the attributes are name (String), location (String), functions send order and receive order.
(ii) Class order attributes date (Date), ~~name (String)~~, number (String), functions: confirm() and close(). The orders can be special order and normal order. Special order and normal order have attributes date and number, confirm(), close(), dispatch() as functions. And a

Scanned with CamScanner

Mangal
Date: / /201

customer can have 1 to many orders.



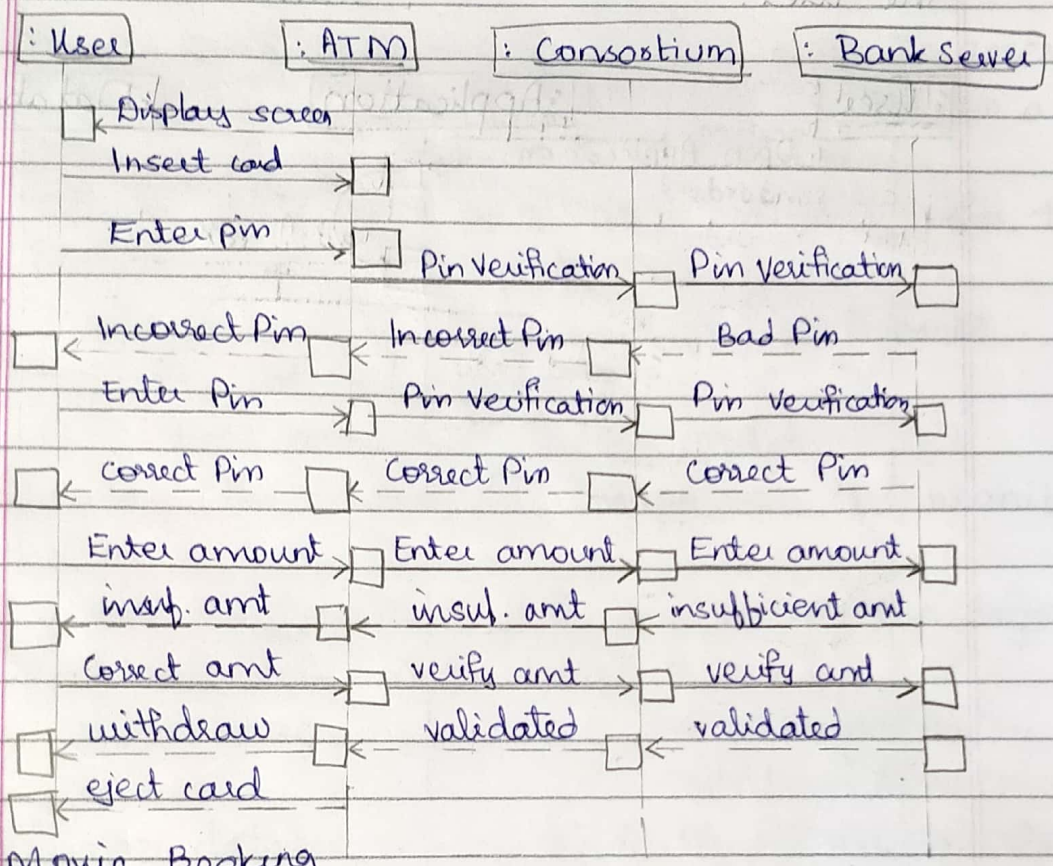
not as relationship

2. Create classes called enrollment.
A student has to enroll to the seminar given by the professor.

Scanned with CamScanner

Mangal
Date: / /201

Interaction Diagram



* Movie Booking
* McDonald Food Order System.
* Emotion based music player

Sequence of operations need to be carried out

(i) emotion based music player

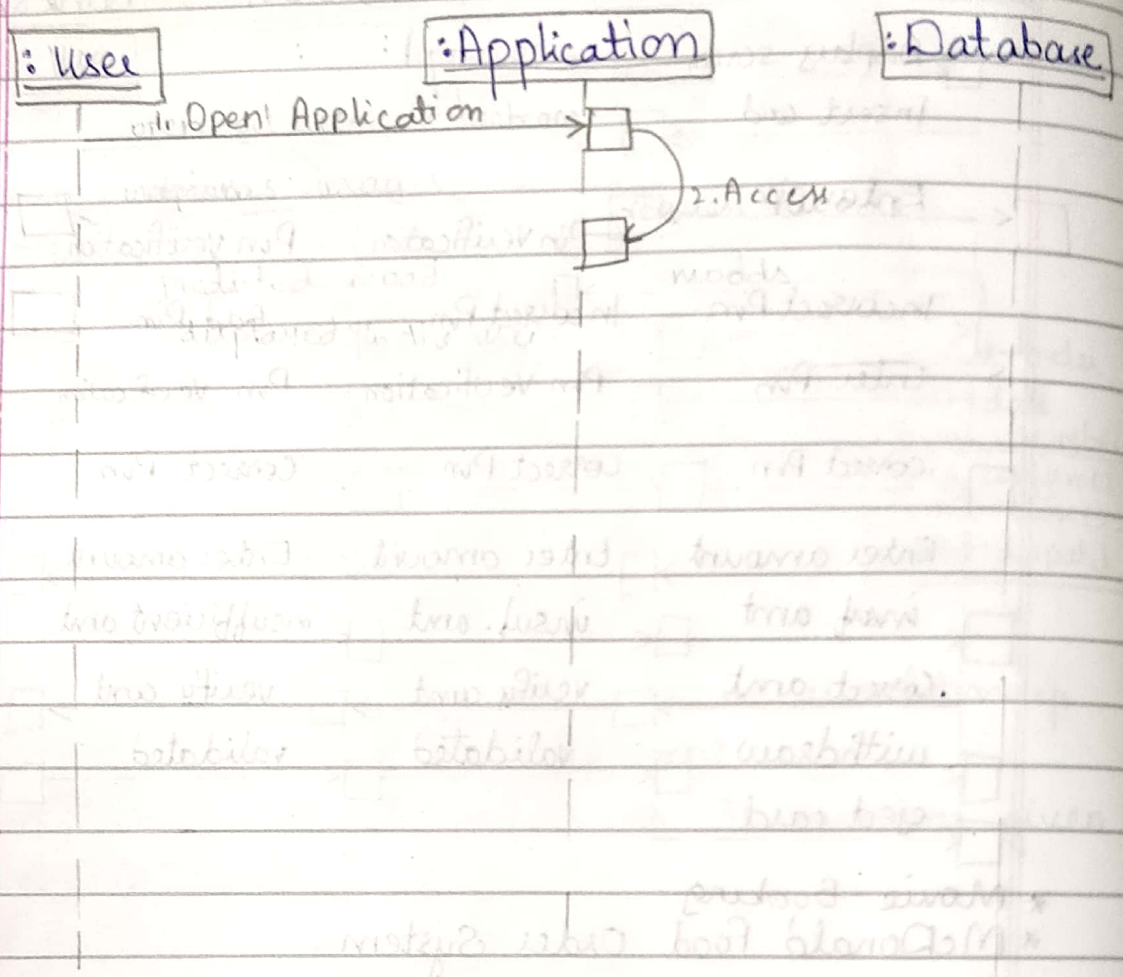
1) open the application by the user
2) device that gets access to web cam
3) webcam captures the image of the user
4) Device uses algorithm to detect face and predict the mood
5) Request database for dictionary of possible moods
6) The mood is retrieved from database
7) Mood is displayed to the user
8) Music is requested from database

Scanned with CamScanner

SDL

Mangal
Date: / /201

9) Playlist is generated and finally shown to the user.



* Dealing with changing requirements:
* Flower's perspective of s/w development

a) Conceptual : what concepts should be included
   eg. objects, methods, behaviour
b) Specification : How to specify the concepts
   eg. interface, abstract class
c) Implementation : How to implement the specification
   eg. coding (algorithms).

Scanned with CamScanner

Mangal
Date: / /201

Q) Find out the common approach for the following:

(Q) If the task is given as writing code to access description of shapes that was stored in a database and then display them
The problem can be further divided into the following steps:
(1) locate the list of shapes in the database
(2) open the list of shapes
(3) sort them according to the rules
(4) display the individual shapes on the monitor

Solu: Create different objects and assign the different responsibilities.

Class	Responsibilities / Methods
→ Shape database	Get specified collection of shapes using get collection method
→ Shape (Abstract Class)	Defines interface for shapes using display method. Returns 'x' location of shape using getX() method, Y using getY() method.
→ Square shape (derived from shape)	Display a square using display abstract method.
→ Circle shape (derived from shape)	Display a circle using display abstract method
→ Collection	It tells all contained shapes to display using display method.

Scanned with CamScanner

Mangal
Date: / /201

	Sort the chapes using sort method.

UNIT-2

Design Pattern : It is a repeatable solution to a commonly recurring problem in software engineering.

Key features:
* Name * Intent
* Problem
* Solution
* Participant and Collaborations
* Consequence
* Implementation
* Generic Structure

Advantages:
1) Reuse of sol^n
2) Establish common terminology
3) Determines which part of the code can be modified

1) Facade Pattern:
It provides a unified interface to a set of interfaces in a subsystem.

Problem:
Accessing subsystem of system
Solution:
Providing interface
Participant
Client and system

Scanned with CamScanner

Mangal
Date: / /201

Consequence
Simplicity, unavailability of other functionalities
Implementation:
Create a class
Generic structure
Class diagram

Scanned with CamScanner