Name :- Raja Sekhar Reddy Vangala

TxST ID :- A05295215

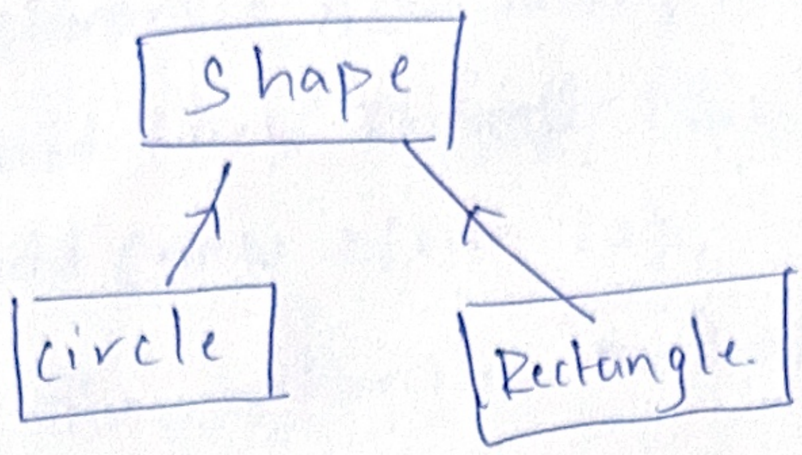
Net ID :- hyf14

1)

Polymorphism :-

Polymorphism means the ability to assume more than one for both in terms of data and operations. It is the capability of an operation exhibiting different behaviour in different instances the behaviour depends on types of data used. It allows an object interface to bind with objects with other classes. The object action will depend on the message passed to the interface and different objects behave differently by receiving some message.

polymorphism result in lack of controllability as actual binding of object reference is not known till runtime some languages like c++ support polymorphism in two ways i.e static binding and dynamic binding. Hence dynamic binding is closely related to polymorphism. It introduces undecidability concern in program based testing as it can lead to message sent to wrong object.




```
List <shape> shape = new Array list <shape> (2);
float sum of Areas = 0f;
for (shape shape : shapes) {
    sum of Areas t= shape.areas²;
}
```

In the above example, it is impossible to say at compile time which implementation of method area will be actually implemented. Hence due to polymorphism, the code associated with polymorphic reference is known only at run time.

If we want to achieve 100% branch coverage, we need to test all possible permutations of instances of circle and Rectangle that could be in the list

```
List <shape> shapes = new Array list<shape> (2);
shapes.add (new circle (2));
shape add (new Rectangle (3,4));
```

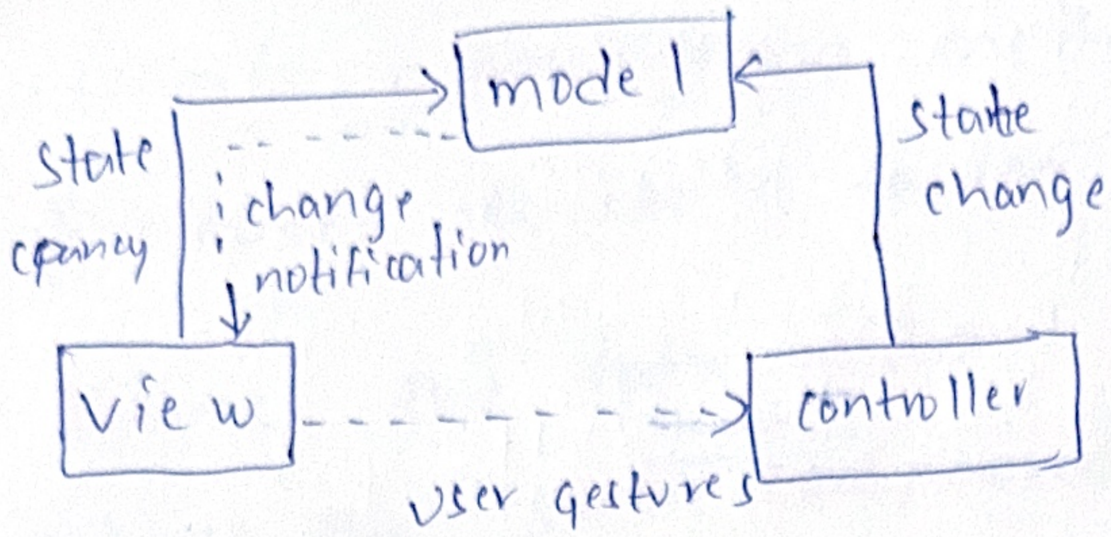
In this case the areas method of the circle class will be called first, followed by the areas method of Rectangle class.

```
List e shapes shapes = new Array list < > (2);
        shapes.add (new Rectangle (3,4);
        shapes.add (new.circle (2);
```

In this case, the rectangle method will be called first followed by circle

By testing both permulations, we can be sure that we are achieving 100 %. branch coverage and that our code is working correctly with polymorphism and dynamic binding.

2)



→ Method invocation

→ Events

model: This level represents the data to the user. model doesn't know about views and controllers

view: View is a visual represent the ~~dat~~ MVC models. It creates an interface to show tha actual output to the user. model or controller that tells view of what to display to use

controller: It acts as a link between a user ~~and~~ with the input by providing view and understands output, converts it providing views and understonds output, converts it to appropriate messages and passes it to views

Advantages of MVC :-

a) Faster development process: It supports rapid and parallel development

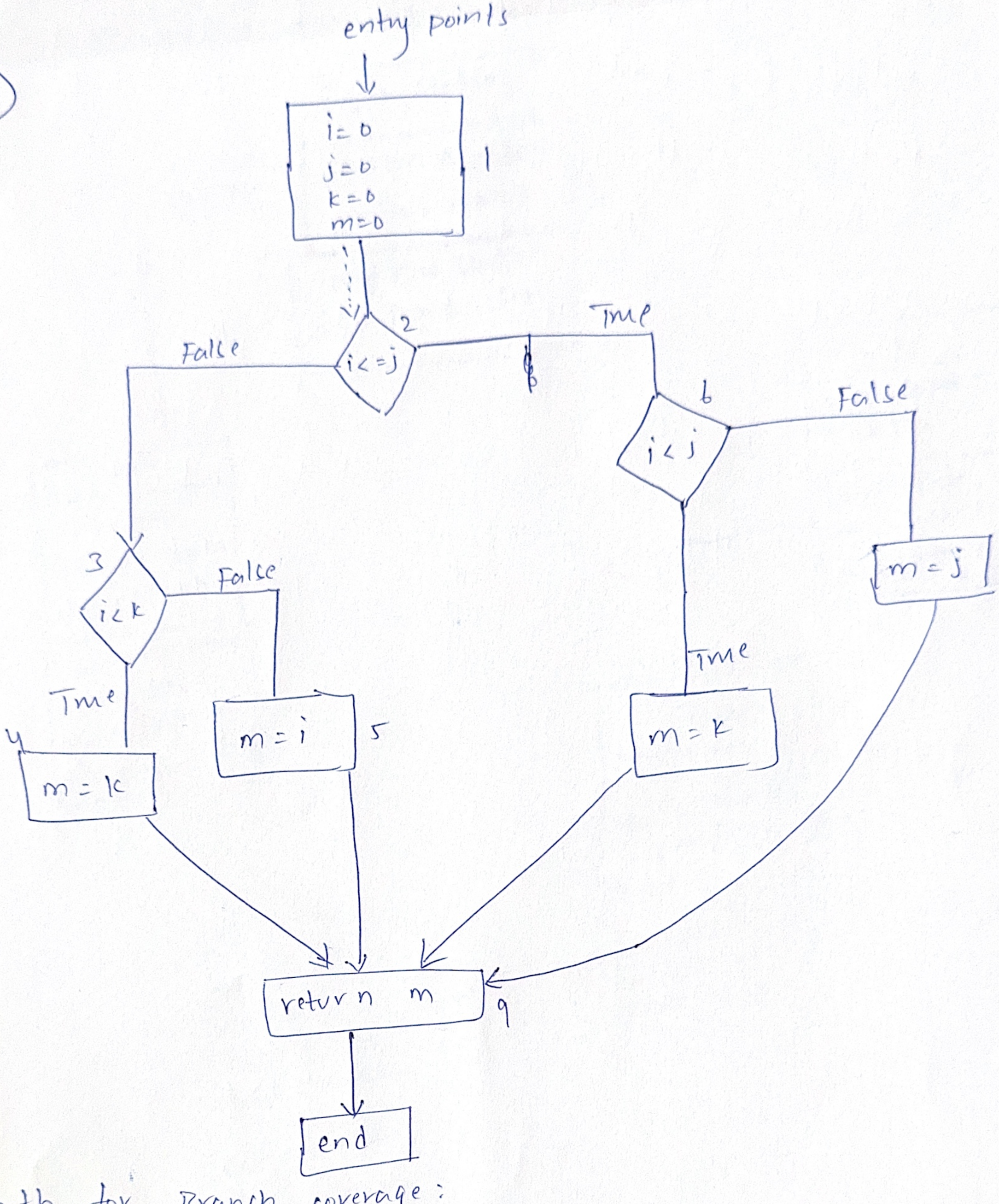
b) Ability to provide multiple views: we can provide multiple views in the MVC model

c) supports asynchronous method invocation: It allows developers to build faster loading web applications

d) Retuns the data without formatting: This helps in calling the same components with any interface.

e) SEO friendly development platform: MVC supports in development of
SEO friendly web pages and applications. It helps in building
SEO friendly URL's generating more visits.

3)



path for Branch coverage:

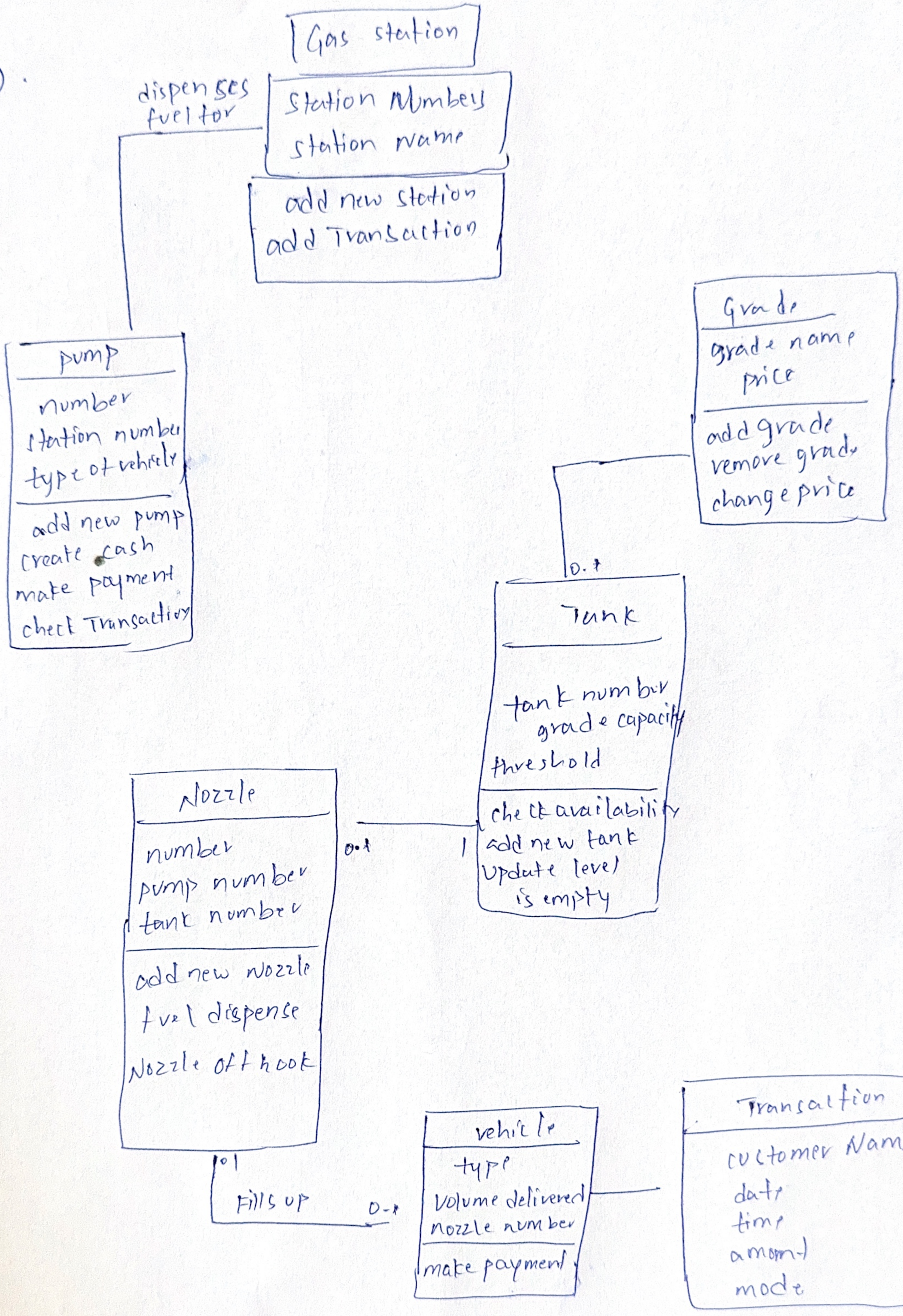
path : 1 → 2(F) → 3(T) → 4 → 9

path2 : 1 → 2(T) → 6(T) → 7 → 9

path 3 : 1 → 2(T) → 6(F) → 7 → 9

path 4 : 1 → 2(F) → 3(F) → 5 → 9

4).


Gas station
dispenses fuel for
Station Number
station name
add new station
add Transaction
pump
number
station number
type of vehicle
add new pump
create cash
make payment
check Transaction
Grade
grade name
price
add grade
remove grade
change price
0..*
Tank
tank number
grade capacity
threshold
check availability
add new tank
update level
is empty
Nozzle
number
pump number
tank number
add new nozzle
fuel dispense
Nozzle off hook
0..1
1
0..1
Fills up
0..*
vehicle
type
Volume delivered
nozzle number
make payment
Transaction
customer Name
date
time
amount
mode