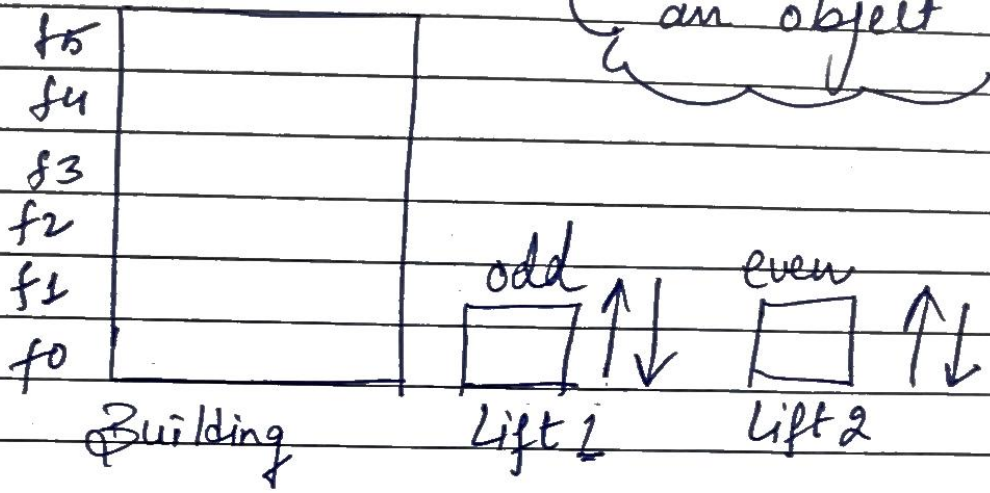
Page No.
Date / /

Elevator Design

Rough Flow

User is an actor not an object



Requirement Clarification

→ How many lift should we consider

→ Lift dispatch Algorith
- ↳ odd even
- ↳ fixed floor
- ↳ Minimum seek (Nearest to user)

Object

→ Building

→ Floor

→ External Button

→ Elevator Car
- → Display
- → Curr floor
- → Direction (Enum)
- → Status (Enum)
- → Internal Button

→ Display

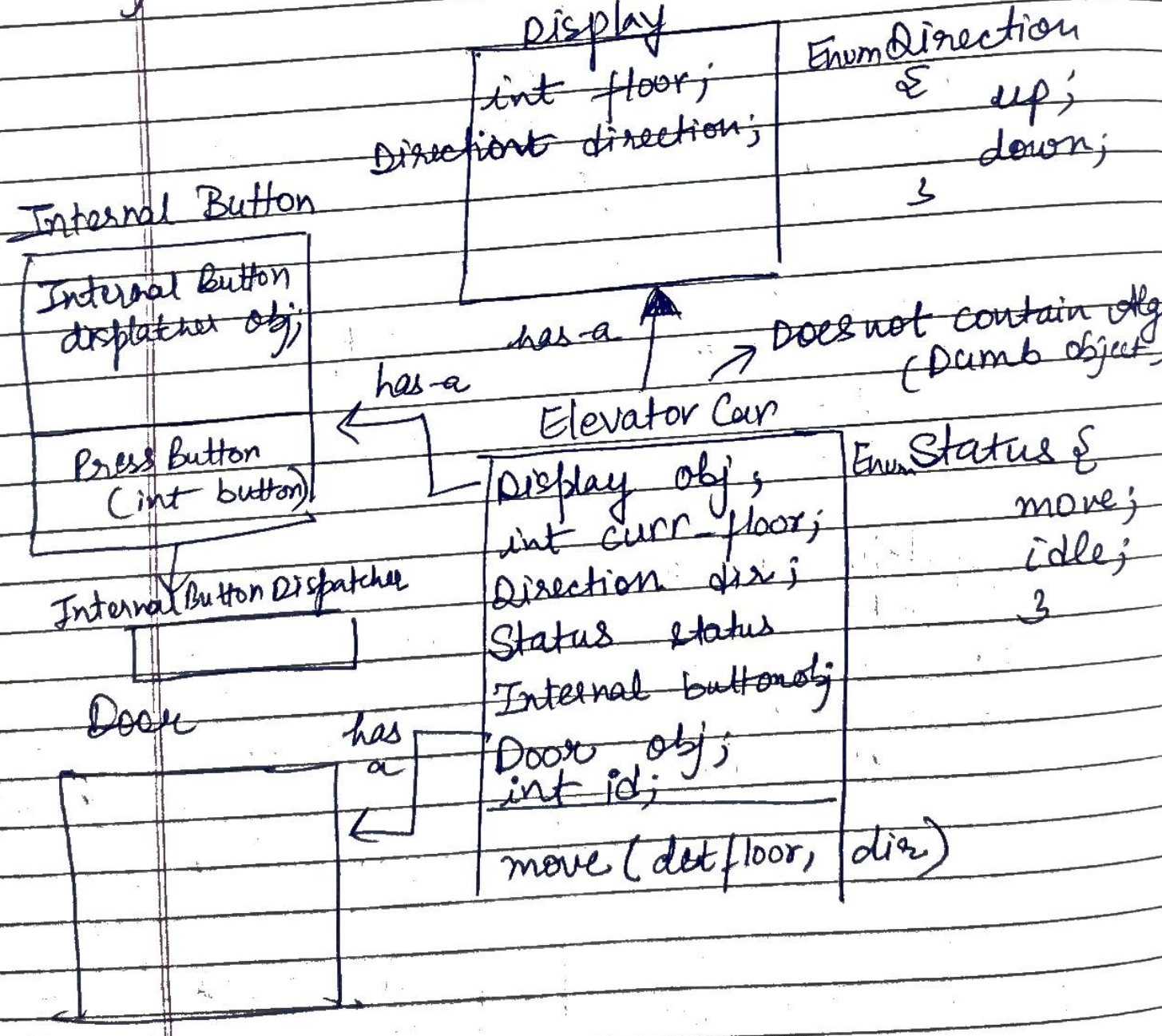
→ Doors

Page No.
Date / /

Elevator System

- Algorithm
- Design.

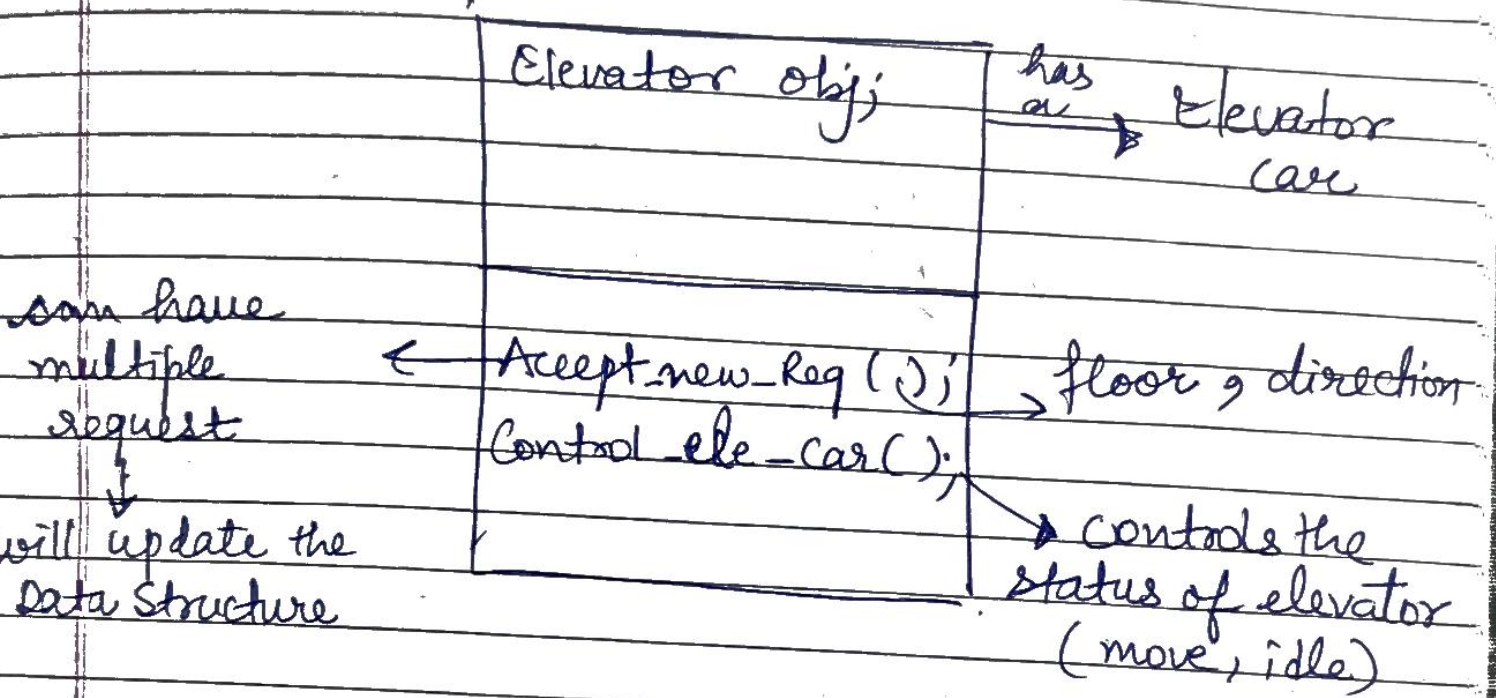
1.) Design / UML Diagram.



Page No.
Date / /

Since Elevator is a Dumb object So we need a Controller of Elevator (holds all algorithm)

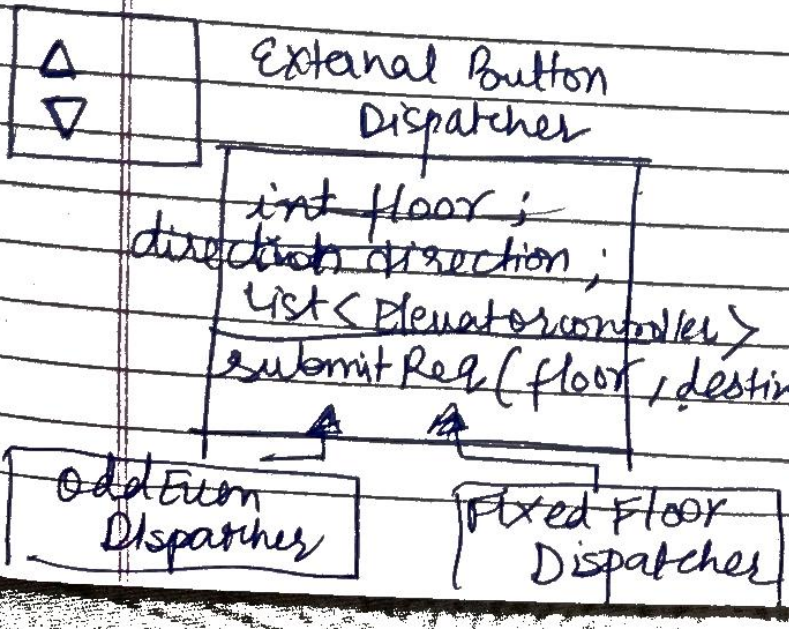
Elevator Controller



Now From where the Elevator Controller receives the requests.

Internal Button

External Button



External Button

pressButton(floor, direct)

has a

External Button Dispatcher

List<ElevatorController> obj

submitReq(floor, dir)

has a

Elevator Contr...

Elevator obj;

submitReq(floor, ...)

→ Next we have floor, building

Floor

floor_id

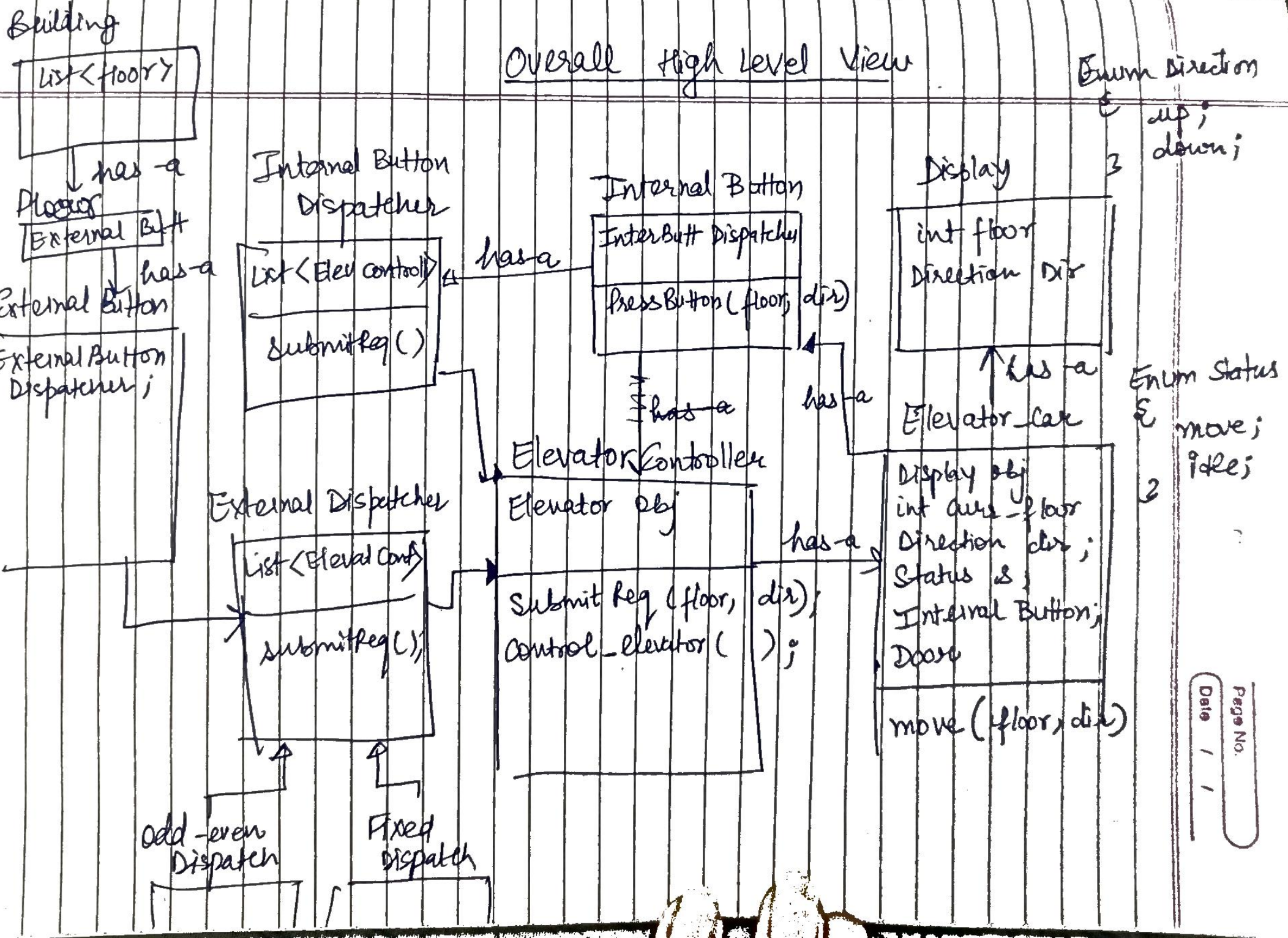
External buttons

has a

Building

List<Floors>

Overall High Level View

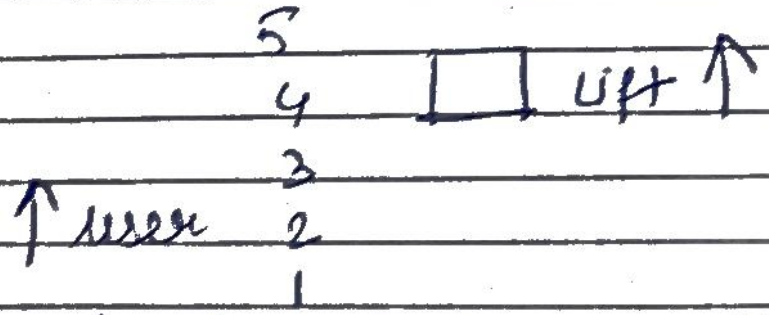


Algorithm

- dispatcher
- Elevator controller

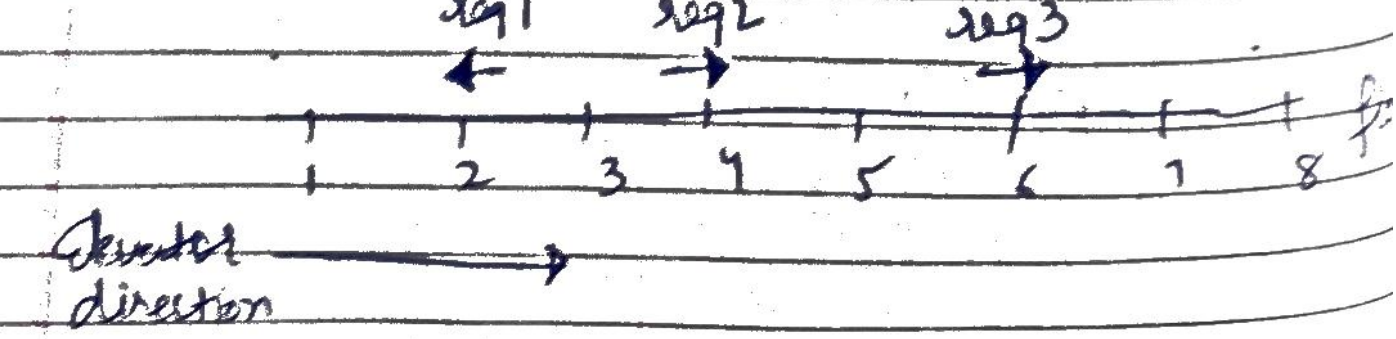
usecase

	user	Lift
usecase 1	↑	↑
2	↑	↓
3	↓	↑
4	↓	↓



user and lift both has up direction But this condition cannot be fulfilled in this scenario.

Algorithm (son)



Page No.
Date / /

Now in this Algorithm the elevator will scan the request in one direction & fulfil those request and when it reaches end then it will change the direction and fulfil requests in that directions.

Disadvantage.

1. If we have request of 4th floor to 5th floor then lift will come but it will go till last even if it does not have any request further.

2.) Look Algorithm

When the elevator is moving, it is looking one step ahead whether their is any request or not.
If not then it will change direction and come back.

- Which data Structure to be used to implement Look?

Current floor of Lift 2
Now Req 5, 4, 3 comes
So the lift first fulfil 3 then 4 & 5
(in ascending order)

But if curr floor 7
and req 5, 2, 6 comes
so it will first serve 6, 5, 2

Page No.
Date / /

We will use two priority queue

Min-heap (for up direction)

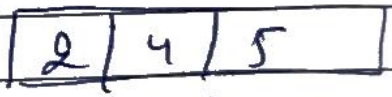
Max-heap (for down direction)

We will store pending jobs in an array/queue

- **Working :**

Let say lift is at 0th floor and req comes from floor 5, 2, 4

When it reaches serves req for floor 2 then req for floor 1 comes So we will store it in pending queue



Min heap

Pending.

Now when lift reaches last floor req comes for 6, 7, 2, 1

Max heap

and then it will serve these request and after reaching floor 1 it will add the next pending req in Min-heap for further.