# Elevator System Design

Residential/Hotel/Office – Requirements differ.

## Requirements

* States : UP, DOWN, IDLE
* Transfer Passengers from one floor to other.
* Open door when idle at a floor.
* 200 elevator floor – 50 elevator cars
* Specs of elevator cars
  + No of passengers.
  + Max Load
  + Max Speed
* Minimize
  + Wait time of system.
  + Wait time of passenger
* Throughput
* Power Usage: Minimization
* Zone covered by elevator – 1-50 floor Zone#1, 51-100 – Zone#2.
* Emergency Alarms, Brakes
* VIP, Utility Elevators
* Monitoring System

# Actor

* ~~Passenger~~  : not required as we are designing elevator control system and not elevator simulation system.
* Elevator Cars
* Floor
* Elevator Doors
* Button Panels – Two types: Out side and Inside Panel
* Dispatcher/Scheduler
* Elevator System – Singleton Object
* Monitoring System

## Use Case

* Calling the elevator.
* Move/Stop the elevator.
* Open/Close the doors.
* Indicating elevator direction.
* Elevator position – Floor Indication inside.
* Emergency brakes.
* Emergency calls.

Class Button (Interface)

+ PressDown()

+IsPressed()

Class Elevator Button (inside)

Class Halls Button

Class Door

+ Open()

+ Close()

+ IsOpen()

Instances as number of floors.

Elevator Motion

+ Move(Destination Floor)

+ Stop()

## Elevator Schedule Algorithm

### FCFS (First Come First Serve)

* 4 States
  + Idle
  + Moving Same direction towards Passenger and direction where passenger wants to go.
  + Moving same direction towards passenger but opposite to direction passenger wants to go.
  + Elevator car is going away from the passenger.

## Shortest Seek Time First (SSTF) Class

* Using Priority Queue (Min Heap) – Requests are put in.
* Use array
* Flaw: Starvation, Multiple Request in parallel

### SCAN (Elevator Algorithm)

* 2 Boolean Array : One for UP and on for DOWN.
* Each elevator has 2 Priority Queue for Min Heap and Max Heap.
* Multiple Requests are handled in parallel.
* Flaw: Elevators are always moving, Power cost & Maintainence Cost.
* Flaw: Always changes direction when it reaches top.

### LOOK

* Tackles the flaw of SCAN
* Look Ahead. If elevator stop as soon as there is no further request and can change direction immediately if request comes.
* Can use binary search tree to check if elevator is at floor X and is there any entry greater than floor X in bst to decide the further direction of movement.

### Destination Dispatch Algorithm

* Tries to match the floor of all passenger requests.