Scope

Design and implement an efficient and scalable elevator management solution for multi-floor buildings. The system should handle multiple elevators, optimize elevator selection based on user requests, and provide real-time status updates to floor panels.

MoveUpState

- elevator: Elevator

+ moveUp()

+ moveDown()

+ stop()

ElevatorObserver

+ update()

MoveDownState

- elevator: Elevator

+ moveUp()

+ moveDown()

+ stop()

State

- elevator: Elevator

+ moveUp()

+ moveDown()

+ stop()

ElevatorManager

- panels : OuterPanel[]

- elevators : Elevator[]

- strategy: ElevatorSelectionStrategy

- observers: ElevatorObserver[]

+ notify()

+ addToQueue(floor: int, direction: Direction)

+ addElevator(e: Elevator)

+ addPanel(p: OuterPanel)

+ setStrategy(s: ElevatorSelectionStrategy)

Elevator

- manager : ElevatorManager

- state : State

- currentFloor: int

- queue: Queue<int>

- processQueue()

+ addToQueue(floor: int)

+ setState(state: State)

+ getCurrentFloor(): int

Direction
<<enum>>

UP

DOWN

OuterPanel

- manager : ElevatorManager

- floor : int

+ requestElevator(d: Direction)

implements

implements

implements

implements

implements

IdleState

- elevator: Elevator

+ moveUp()

+ moveDown()

+ stop()

ElevatorSelectionStrategy

+ select(): Elevator

implements   *implements*

NearestElevatorStrategy

+ select(): Elevator

LoadBalancerStrategy

+ select(): Elevator