Harshil Chudasama
CS5004 Winter 2024

Building Design

This is a complicated project. The purpose of this design step is to help you succeed in this project. We have asked you to build a UML diagram of the entire class structure.

Include the UML Question 9 and 10 will assess this.

Answer the following questions in this document and upload with your UML diagram.

1) How are you storing your elevators in your Building model.

   Elevators are stored in a List<Elevator> within the Building class. This list allows for easy management of multiple elevators, facilitating operations like iteration, status updates, and request distribution.

2) How are you storing the incoming requests so you can distribute them to the elevators.

   Incoming elevator requests are stored in two Queue<Request> fields within the Building class: upRequests for requests to move up and downRequests for requests to move down. This categorization helps in efficiently managing and assigning requests based on their direction and the current state of the elevators.

3) How are you distributing your downRequests and your upRequests to the elevators?

   Requests are distributed to elevators through the distributeRequests() method in the Building class. This method examines the direction and current floor of each elevator, along with the direction and destination of each request, to optimally assign requests to elevators. The method prioritizes proximity, direction, and elevator capacity in its distribution logic.

4) How are you removing all requests when a takeOutOfService request is received.

   When an elevator is taken out of service (via the takeOutOfService method), the Building class responds by clearing both the upRequests and downRequests queues, effectively removing all pending requests. Additionally, the specific elevator's clearRequests() method is called to remove any internally stored requests.

5) How does your step method handle updating the elevators?

   Each Elevator has a step() method that updates its state with each call—advancing the elevator by one floor, opening/closing doors as needed based on floor requests, and

updating its direction if necessary. The Building class can invoke this method on each elevator in its list to simulate real-time movement and operation.

6) How do you start processing requests?

   Request processing is started by calling the startElevatorSystem() method in the Building class. This method sets the system status to RUNNING, allowing elevators to begin accepting and fulfilling requests. It also triggers the initial distribution of any queued requests to available elevators.

7) How do you take the building out of service?

   The building is taken out of service by calling the stopElevatorSystem() method in the Building class. This method sets the system status to OUT_OF_SERVICE, stops all elevators from accepting new requests, and initiates the process of bringing all elevators to their designated rest positions.

8) How do you take the elevators out of service?

   Individual elevators are taken out of service using their takeOutOfService() method. This method marks the elevator as out of service (isInService = false), clears any pending internal requests, and may direct the elevator to move to the ground floor or another specified floor to await maintenance or inspection.