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.

In the Building model, I am storing the elevators using a List data structure. Each elevator is represented by an object that implements the **ElevatorInterface**. The List allows for dynamic storage of elevators and facilitates easy access and manipulation.

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

Incoming requests are stored in two separate lists within the Building model: **upRequests** and **downRequests**. These lists hold instances of objects implementing the **Request** interface. Storing requests in separate lists based on their direction simplifies the process of distributing them to the appropriate elevators.

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

Distribution of requests is handled by the **distributeRequests()** method within the Building model. This method iterates over the lists of up and down requests, and for each request, it determines the appropriate elevator based on factors such as current position, direction, and capacity. Once the elevator is selected, the request is assigned to it for processing.

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

When a **takeOutOfService** request is received for an elevator, all requests associated with that elevator are removed from the **upRequests** and **downRequests** lists within the Building model. This ensures that no pending requests are assigned to the elevator that is being taken out of service.

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

The stepElevatorSystem() method in the Building model handles updating the elevators by invoking the stepElevatorSystem() method on each elevator object stored in the List.

This method call triggers the elevator to perform its next step of operation, such as moving to the next floor, processing requests, or opening/closing doors.

6) How do you start processing requests?

To start processing requests, the **startElevatorSystem()** method in the Building model is called. This method initiates the operation of all elevators in the system, allowing them to begin processing requests from the **upRequests** and **downRequests** lists.

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

To take the building out of service, the **stopElevatorSystem()** method in the Building model is invoked. This method halts the operation of all elevators in the system and clears all pending requests from the lists. Once the building is out of service, no new requests are accepted until the system is restarted.

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

The **takeOutOfService()** method is provided in the **ElevatorInterface**, allowing individual elevators to be taken out of service. When this method is called for a specific elevator, it sets the elevator's status to out of service and performs necessary cleanup actions such as removing pending requests and resetting the elevator's state.