

**Faculty of Informatics and Computer Science**

**Real Time Systems**

**Elevator Controller**

**Supervised by**

**Dr. Amr S. Ghoneim**

**TA. Israa Lotfy – TA. Maher**

|  |  |
| --- | --- |
| Name | ID |
| Amr Ahmed Abd El Rahman | **162697** |
| Ahmed Mohamed Halim | **162097** |
| Mahmoud Sami | **164778** |
| Abd El Rahman Essam | **155295** |
| Abd El Rahman Kassabi | **153232** |

# **Overview**

The elevator system controller designed in our project is considered an “ideal” elevator in which some of the technical corners are cut. Our elevator has the basic functionalities and essentials that usual elevator systems have, such as moving up and down, open and close doors, and of course, alarming of excess capacity, emergency brakes and basic car and call buttons.

# **Use Case Diagram**



Figure 1. UML documentation of an elevator system

All systems interact with human or automated actors (i.e. controllers) that use the system for some purposes, and both human and actors expect the system to behave in predictable ways. In our project, a use case model the main actors and functionalities related to those actors, whereas it shows set of actors and their relationships.

* **Passenger**

In our system, the passengers considered one of the cores and role actors who interact with the elevator system. The passenger interacts with the elevator by making car calls and also interact with the car button inside the car it ‘self (i.e. requesting a desired floor)

* **Elevator controller**

In our system, the elevator controller is considered an automated actor, where it plays an important role in managing the calls and processing it accordingly to the car (elevator cabinet)

* **Button controller**

In our system, the button controller is considered a mediator between the button and the system, it’s considered a sub system of the elevator controller, where it’s main functionality is to illuminate the car call button and to turn it off once the car has reached it’s requested car call floor.

# **State Machine Diagram**



Figure 2. State Machine Diagram

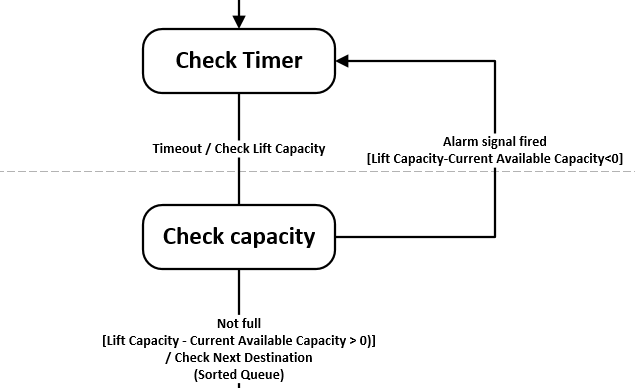


Figure 3. Checking Capacity

When the doors are closing but not fully closed, if there are passengers who want to get into the car, or if there exists an excess amount of capacity in the elevator car, the doors should open again for another period of time, then close again to check if the condition satisfies.

# **Sequence Diagram**



Figure 4. Floor Request Scenario of our system

# **Class Diagram**



# **Component Diagram**

Future work will be dedicated toward this diagram

# **Conclusion**

In this report, a detailed UML documentation for an elevator control system is designed.

The UML diagrams used in this documentation are Use Case Diagram, Class Diagram, Sequence

Diagram, and State Machine Chart Diagram. Every diagram in UML is just a graphical presentation of some of the aspects in our system. Which will implement it using Java programming language. The functions of the elevator system described in our project are still limited with little additional features that are more likely needed in the real world. Robustness of our elevator system is needed, as result that human lives will be involved during using the system. Further, our software system is considered a critical software. In our system hardware components are involved, we have tried to provide more pragmatic and comprehensive approach to our system development rather than just software.