Software Requirement Specifications: Elevator system

Hanne-Grete Alvheim & Hao Peng

May 16, 2022

Contents

1 Introduction

1.1 Purpose of this document

The purpose of this document is to present a detailed description of an elevator system consisting of 3 elevators that are supposed to work together across 4 floors.

1.2 Scope of this document

This document specifies requirements made for a simple elevator system, that is not supposed to carry people, but instead be a test project. It contains a general description of how the elevators should work together, as well as requirements for the system.

2 General description

2.1 User Needs

The user of the elevators is a person. When the user presses a button on the buttonpanel, an elevator is supposed to come and get the user to its destination, specified by the button the user pushes inside of the elevator. If the doors don't close, the elevator won't run to ensure the safety of the user, and optimally the closest elevator should get the user. Expectations of the user can be that

- The order of the user is not lost
- Multiple elevators should get the user to its desired destination faster than only one
- An individual elevator should behave sensibly and efficiently, meaning that it should get the user (or assign another elevator to do it), get it to the desired floor, open doors when it is safe and not run if it is not safe. The lights and buttons should also function as expected.

2.2 Assumptions and Dependencies

Already made Software was used, this includes

- An algorithm for finding the optimal elevator to go (link: https://github.com/TTK4145/Project-resources/tree/master/cost_fns)

- Code for sending and receiving messages (link: https://github.com/TTK4145/Network-go/tree/master/network)

It is assumed that these run perfectly. The code for the elevator system was written in Go. It is assumed that the commands from Go will work, and that there won't happen any shift of bits or any natural disasters that affect the elevators.

3 Functional requirements

- The system must remember the orders of the users, for instance using primary- and backup memory log
- The system must execute every order requested by the user, even if some of or all the elevators stop working for a while
- After reaching a floor, a timer should check that the doors are open at least 3 seconds, to ensure that the user has gotten out of the elevator
- The elevator must stand still at the closest floor if it is affected by an obstruction
- The lights on the hall panel and the floor panel must be switched on if the buttons are clicked by a user, and switch off when the elevator has gone to the desired floor
- The elevators must communicate and message with each other via a network, so that they don't try to pick up the same user
- The elevator that gets the order by the user on its hall panel must distribute the order to the elevator that should optimally take it. If the elevator is not able to take the order, the other elevator must reassign and resend the order to the optimal elevator once again
- If the elevators lose contact with each other, the elevator(s) must take the orders of the elevator(s) they have lost contact with.
- In case of motorstop of one or more elevators, the functioning elevator(s) must take on the order(s) of the elevator(s) that got the motorstop

4 Interface requirements

4.1 User Interface

The elevator communicates with the user of the elevator via the hall- and floor buttons.

4.2 Elevator communication

The elevators use P2P communication in order to communicate with each other, and broadcast their elevator struct sent as a Json-object to a common port that every elevator is listening to. They also send messages in the form of slices if they want to give another elevator an order they received on their own panel, and placed-messages telling an elevator that has given the order that the order will indeed be taken.

5 Performance Requirements

The user is expected to get an elevator to its location at least after 30 seconds worst case.

6 Design Constraints

There are no specified design constraints.

7 Non-functional Attributes

The elevator system should be scalable, so that it can work for M elevators with N floors.

8 Preliminary Schedule and Budget

Overall time dimension: 1 semester, overall budget: 0 czk.

9 Appendices

- Finished code: https://github.com/TTK4145-Students-2021/project-gruppe64