Software Requirements Document for Elevator Control System

By: Jonah Weisse

October 4, 2016

Version 1.0

Table of Contents

1. Introduction

1.1. Purpose

This document describes the requirements for an elevator control system simulator. The simulator is meant to model an actual elevator system for a building with four floors. It is based on the format from a simplified version of IEEE Standard 830-1998 for writing software requirements specification. The project is designed for educational purposes at UW-La Crosse for the course CS-741 – Software Engineer Principals.

1.2. Scope

The system described in this document is a simulation of an actual elevator control system. It mimics the movements, buttons, and other fixtures of a single elevator that would be run in a building. The product could be used to demonstrate how an elevator works.

1.3. Problem Description

For the purpose of this project, let us assume that the elevator serves a building with four floors. The floors will be labelled 1, 2, 3, and 4. The control system should display the current position of the elevator at any time including its movements. This will be displayed in the GUI with an image of the elevator showing the direction of movement. It should also show the status of buttons and lights both inside the elevator and in every floor. The buttons inside the elevator will consist of one button for each floor, for which each will be lit up if they are pressed. These floor buttons (inside the elevator) will stay lit up until that floor is reached. Inside the elevator will also be a screen indicating which floor the elevator is currently at as well as the direction it is travelling.

Every floor will have two buttons: an up and a down arrow. Both will light up if pressed and stay lit up until the elevator reaches that floor and is travelling in the correct direction. Every floor will also have a screen indicating the current position of the elevator. The screen will also show which direction the elevator is travelling. In the event that no one is using the elevator, it will be considered idle and stay on its current floor. The screens will not show any direction if the elevator is idle. The GUI should be interactive so that it should be possible to test all scenarios. The buttons that will be able to be pressed are the up and down arrows on each floor and the floor buttons inside the elevator. Preference will be given first to the buttons inside the elevator. That is, once the elevator reaches a floor where passengers are waiting, the buttons inside the elevator will indicate where the elevator is travelling as long as it is in the correct direction. If no buttons are pressed or lit up when a passenger is picked up, the elevator will wait 10 seconds for the passenger to choose a floor in the correct direction. If no floor is chosen, the elevator will respond to the buttons from other floors or become idle. If the elevator is heading in some direction to pick someone up or drop someone off it will only stop and pick up additional passengers if they are travelling in the same direction as the elevator and are on the current path of the travelling elevator.

1.4. Assumptions

The following assumptions are made to support the development of the elevator control system:

1. The building that this system is designed for only has one elevator. This elevator is assumed to only travel between four floors.
2. The elevator is assumed to never have mechanical or load bearing issues. The simulation only mimics the functioning movements of the elevator, not any motor control problems that the system may face.
3. It is assumed that when the system is started, the elevator will always reside on the first floor.
4. Control of the elevator is strictly based off of the buttons pressed by the user in the simulation. There are no outside factors such as a control panel or an emergency stop button for the elevator.
5. The elevator's doors are only controlled by a timer. There are no door open or door close buttons, as it is assumed the passengers will get on and off the elevator within the allotted time. The doors are always closed when the elevators travels between floors, and they always open when it reaches a floor that it was called to. When the elevator is idle, the doors will be closed.

1.5. System Requirements

The product must be able to run on a computer that has Java installed.

1.6. User Characteristics

No additional knowledge is required from users.

2. References

1. ISO/*IEEE Standard 29148-2011 Systems and Software Engineering – Life Cycle Processes – Requirements Engineering*, IEEE Publications, 2011.

3. Functional Requirements

Each functional requirement is given in the following format:

Index:
Name:
Purpose:
Input parameters:
Action:
Output parameters:
Exceptions:
Remarks:
Cross-references:

Index refers to a unique index assigned to this functional requirement. It will be used for cross-referencing this requirement from other requirements and also from other documents such as the design document.

Name is a descriptive name given to this functional requirement. This name does not need to be unique.

Purpose is a short description (in a line or two) of the functionality. It is used to quickly understand the functionality and is also used to search the required functionality when browsed through.

Input parameters refers to a set of parameters that the given functionality accepts as input. These parameters are required in order to design and implement the current functionality. No type information will be included for parameters in this document.

Action refers to a set of activities or tasks to be completed in order to implement the given functionality. The ordering of these tasks is not specified in this document although some ordering may be explicit from the description. No implementation details must be given in this document.

Output parameters refers to a set of parameters that are output/exhibit by the current functionality, when implemented. No type information will be included for parameters in this document.

Exceptions refers to a set of conditions, each of which indicates a situation in which the implementation of the given functionality will stop. Notice that this column only lists a set of exceptions that might occur but does not suggest any action that must be taken when the exceptions occur. These actions will be included in the design document.

Remarks includes a set of comments that explain more about the functionality. It also describes hints to the designer and implementer that are suggested by the requirements analyst.

Cross-references refers to a set of other functional requirements that are related to the current functionality.

3.1. Functional Requirements for Any User

Index: F1
Name: Make a request to travel up
Purpose: To call the elevator and a choose a floor to travel up the building
Input parameters: Floor to which the elevator is travelling
Action: Bring the elevator to the floor that the passenger is currently on, and wait for them to choose a floor to travel to
Output parameters: Floor button that the passenger is travelling to lights up in the elevator
Exceptions: If the elevator is currently picking up or dropping off passengers going down, it will not stop for this passenger until it has serviced each of these passengers.
Remarks: Passenger will press the Up arrow on their current floor.
Up arrow will light up until the elevator arrives and is travelling up.
Doors of the elevator will open, and then close after 5 seconds.
Passenger is expected to choose a floor to travel to when elevator arrives, and then the doors will close.
Cross-references: F3, F5.

Index: F2
Name: Make a request to travel down
Purpose: To call the elevator and a choose a floor to travel down the building
Input parameters: Floor to which the elevator is travelling

Action: Bring the elevator to the floor that the passenger is currently on, and wait for them to choose a floor to travel to

Output parameters: Floor button that the passenger is travelling to lights up in the elevator

Exceptions: If the elevator is currently picking up or dropping off passengers going down, it will not stop for this passenger until it has serviced each of these passengers.

Remarks: Passenger will press the Down arrow on their current floor.
Down arrow will light up until the elevator arrives and is travelling up.
Doors of the elevator will open, and then close after 5 seconds.
Passenger is expected to choose a floor to travel to when elevator arrives, and then the doors will close.

Cross-references: F4, F5.

Index: F3

Name: Elevator travels up the building

Purpose: To travel to floors above the elevator's current position

Input parameters: Floor to which the elevator is travelling

Action: Bring the elevator up one or more floors in the building.

Output parameters: None.

Exceptions: None.

Remarks: After the doors of the elevator are closed and a button is selected, the elevator will move up the building. The screen will also show the floors the elevator is passing as well as arrow in the up direction

Cross-references: F1, F5.

Index: F4

Name: Elevator travels down the building

Purpose: To travel to floors below the elevator's current position

Input parameters: Floor to which the elevator is travelling

Action: Bring the elevator down one or more floors in the building.

Output parameters: None.

Exceptions: None.

Remarks: After the doors of the elevator are closed and a button is selected, the elevator will move down the building. The screen will also show the floors the elevator is passing as well as arrow in the down direction

Cross-references: F2, F5.

Index: F5

Name: Elevator reaches a floor

Purpose: To arrive at a floor in the building and let passengers get on or off
Input parameters: None
Action: Open the doors the elevator
Output parameters: None.
Exceptions: None.
Remarks: After the elevator has travelled up or down to the destination floor, the doors will open to allow passengers to get on or off. Additionally, the screen will change to this destination floor and the direction of elevator will be blank since the elevator is done travelling in this direction.
Cross-references: F1, F2, F3, F4.

Index: F6
Name: Travel to the first floor
Purpose: To travel from the second, third, or fourth floor to the first
Input parameters: None
Action: Bring the elevator down to the first floor of the building, and open the elevator doors.
Output parameters: First floor button inside elevator should light up and remain lit until first floor is reached.
Screen should display the direction the elevator is travelling (down), and each floor that the elevator reaches on its way to the first floor until it reads '1' once it reaches it.
Exceptions: None.
Remarks: Elevator will travel directly to the first floor once the first floor button is pressed inside the elevator.
It should only stop if picking up passengers on the elevator's path who are also travelling in this direction (down).
Cross-references: F7, F8, F9.

Index: F7
Name: Travel to the second floor
Purpose: To travel from the first, third, or fourth floor to the second
Input parameters: None
Action: Bring the elevator up or down to the second floor of the building, and open the elevator doors.
Output parameters: Second floor button inside elevator should light up and remain lit until second floor is reached.
Screen should display the direction the elevator is travelling (up or down), and each floor that the elevator reaches on its way to the second floor until it reads '2' once it reaches it.

Exceptions: None.
Remarks: Elevator will travel directly to the second floor once the second floor button is pressed inside the elevator.
It should only stop if picking up passengers on the elevator's path who are also travelling in this direction.
Cross-references: F6, F8, F9.

Index: F8
Name: Travel to the third floor
Purpose: To travel from the first, second, or fourth floor to the third.
Input parameters: None
Action: Bring the elevator up or down to the third floor of the building, and open the elevator doors.
Output parameters: Third floor button inside elevator should light up and remain lit until third floor is reached.
Screen should display the direction the elevator is travelling (up or down), and each floor that the elevator reaches on its way to the third floor until it reads '3' once it reaches it.
Exceptions: None.
Remarks: Elevator will travel directly to the third floor once the third floor button is pressed inside the elevator.
It should only stop if picking up passengers on the elevator's path who are also travelling in this direction.
Cross-references: F6, F7, F9.

Index: F9
Name: Travel to the fourth floor
Purpose: To travel from the first, second, or third floor to the fourth.
Input parameters: None
Action: Bring the elevator up to the fourth floor of the building, and open the elevator doors.
Output parameters: Fourth floor button inside elevator should light up and remain lit until fourth floor is reached.
Screen should display the direction the elevator is travelling (up), and each floor that the elevator reaches on its way to the fourth floor until it reads '4' once it reaches it.
Exceptions: None.
Remarks: Elevator will travel directly to the fourth floor once the fourth floor button is pressed inside the elevator.

It should only stop if picking up passengers on the elevator's path who are also travelling in this direction (up).

Cross-references: F6, F7, F8.

Index: F10

Name: Watch the elevator travel between floors

Purpose: To simulate the elevator's movements up and down the building

Input parameters: None

Action: Visual display of an elevator will move up and down a diagram of a building.

Output parameters: None.

Exceptions: None.

Remarks: Visual display of the elevator will include seeing whether the doors to the elevator are open or closed.

Cross-references: F3, F4.

Index: F11

Name: See the floor that the elevator is on and the direction it is travelling.

Purpose: To gather information about the elevator's location and direction when inside or waiting for the elevator

Input parameters: None

Action: Screen displays either '1', '2', '3', or '4'; and either an up or down arrow.

Output parameters: None.

Exceptions: When the floor is not travelling in any direction, no arrow should be present.

Remarks: Floor number on the display screen will remain until another floor is reached (even if elevator is currently between floors).

Cross-references: None.

---

4. Non-Functional Requirements

Accessibility

The entire application and all functionalities of it, including all the buttons and screens should be easy to see and use.

Aesthetics

The simulation should have a well-designed visual display that is easy to follow and aesthetically pleasing to the user. This includes the location and quality of the three views included in the system: the building model, the display outside the elevator on each floor, and inside the elevator.